



*A Jain Minority Institution*

## **Major Project Report**

# **Chronic Kidney Disease Prediction Using Machine Learning**

*Submitted By:*

Kunal (02955202719)

Gaurav Chamoli (04555202719)

Kartik Gupta (04955202719)

**BACHELOR OF TECHNOLOGY IN  
COMPUTER SCIENCE AND ENGINEERING**

**At**

**BHAGWAN MAHAVEER COLLEGE OF ENGINEERING AND MANAGEMENT**

**JAGDISHPUR, SONEPAT**

**AFFILIATED TO GGSIPU DWARKA, DELHI**

## Declaration

We Kunal(02955202719), Gaurav Chamoli(04555202719) and Kartik Gupta(04955202719), of Fourth Year B.Tech., in the Department of Computer Science and Engineering from BMCEM hereby declare that the work presented in this report entitled “Chronic Kidney Disease Prediction Using Machine Learning”, in fulfilment of the requirement for the award of the degree Bachelor of Technology in Computer Science Engineering submitted in CSE Department, Bhagwan Mahavir College of Engineering and Management affiliated to Guru Gobind Singh Indraprastha University, New Delhi, during the academic year 2019-2023 is an authentic record of our Work carried out during my degree under the guidance of Ms. Pratibha Gautam Ma’am. The work reported in this has not been submitted by me for the award of any other degree or diploma.

Kunal (02955202719)

Gaurav Chamoli (04555202719)

Kartik Gupta (04955202719)

# Acknowledgement

We would like to express our deep gratitude to our guide Ms. Pratibha Gautam Ma'am for her valuable guidance, faculty of computer science and engineering, BMCEM and timely suggestions during the entire duration of our dissertation work, without which this work would not have been possible. We would also like to convey our deep regards to all other faculty members of BMCEM, who have bestowed their great effort and guidance at appropriate times without which it would have been very difficult on our part to finish this work. Finally, we would also like to thank our friends for their advice and pointing out our mistakes, parents, and classmates for their encouragement throughout our project period. And last but not least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

# Abstract

Chronic Kidney disease (CKD) is a prevalent and potentially life-threatening condition affecting millions of individuals worldwide. 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. Early detection and accurate prediction of CKD can significantly improve patient outcomes and reduce the burden on healthcare systems. There has been a significant amount of research conducted on this topic. Nevertheless, we are bolstering our approach by making use of predictive modeling. In recent years, machine learning (ML) techniques have emerged as powerful tools for CKD prediction. This report provides a comprehensive overview of the current state of research on CKD prediction using machine learning, highlighting the methodologies, datasets, performance metrics, challenges, and future directions involved. This study starts with 25 variables in addition to the class property, but by the end, it has narrowed the list down to 30% of those parameters as the best subset to identify CKD. The purpose of this project is to help others with early detection of Chronic Kidney Disease and take preventive measures accordingly.

# TABLE OF CONTENTS

- 1. Declaration**
- 2. Acknowledgement**
- 3. Abstract**
- 4. Chapters:-**

## **Chapter 1: INTRODUCTION**

- 1.1. Purpose
- 1.2. Objective
- 1.3. Project Scope
- 1.4. Technologies to be Used

## **Chapter 2: LITERATURE SURVEY**

- 2.1 Existing System
- 2.2 Proposed System
- 2.3 Feasibility Study

## **Chapter 3: METHODOLOGY/PLANNING OF PROJECT**

- 3.1 Process Flow Chart
- 3.2 ER Diagram
- 3.3 Data Flow Diagram

## **Chapter 4: SYSTEM REQUIREMENTS**

- 4.1 Installation
- 4.2 User Interface
- 4.3 Minimum Requirements

## **Chapter 5: THEORY**

### **5.1 Basics of ML**

### **5.2 Algorithms Used**

## **Chapter 6: EXPERIMENTAL ANALYSIS**

### **6.1 Code**

#### **6.1.1 Backend Code**

##### **6.1.1.1 Training Code**

##### **6.1.1.2 Main Code**

#### **6.1.2 Frontend Code**

##### **6.1.2.1 Form Page**

##### **6.1.2.2 Result page**

### **6.2 Output**

## **Chapter 7: FUTURE ENHANCEMENTS**

## **Chapter 8: BIBLIOGRAPHY**

# INTRODUCTION

## ❖ Purpose

- The purpose of this project is to help others with early detection of Chronic Kidney Disease and take preventive measures accordingly. 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. Early detection, accurate prediction and treatment of CKD can significantly improve patient's quality of life and reduce the burden on healthcare systems. Machine learning techniques have shown great potential in predicting Chronic Kidney Diseases, allowing for early diagnosis and timely intervention.

## ❖ Objective

- The objective of this project is to improve my knowledge of different machine learning algorithms, learn to implement them in real-world problems and help in significantly improving a patient's quality of life and reduce the burden on healthcare systems.

## ❖ Project Scope

- This project is quite useful in predicting Chronic Kidney Disease as it helps in accurately predicting CKD. Hence this project is a good scope.

## ❖ Technologies Used

- **LANGUAGES**

- Python
- JavaScript
- HTML
- CSS

- **FRAMEWORK**

- Flask

- **OPERATING SYSTEM**

- Windows 10

- **IDE (Integrated Development Environment)**

- Jupyter
- VS code



# LITERATURE SURVEY

## ❖ Existing System

- There are existing systems which are similar to my model but some do not have many algorithms at one place, other do not have user friendly interface and most importantly most of them do not have our level of accuracy and precision.

## ❖ Propose System

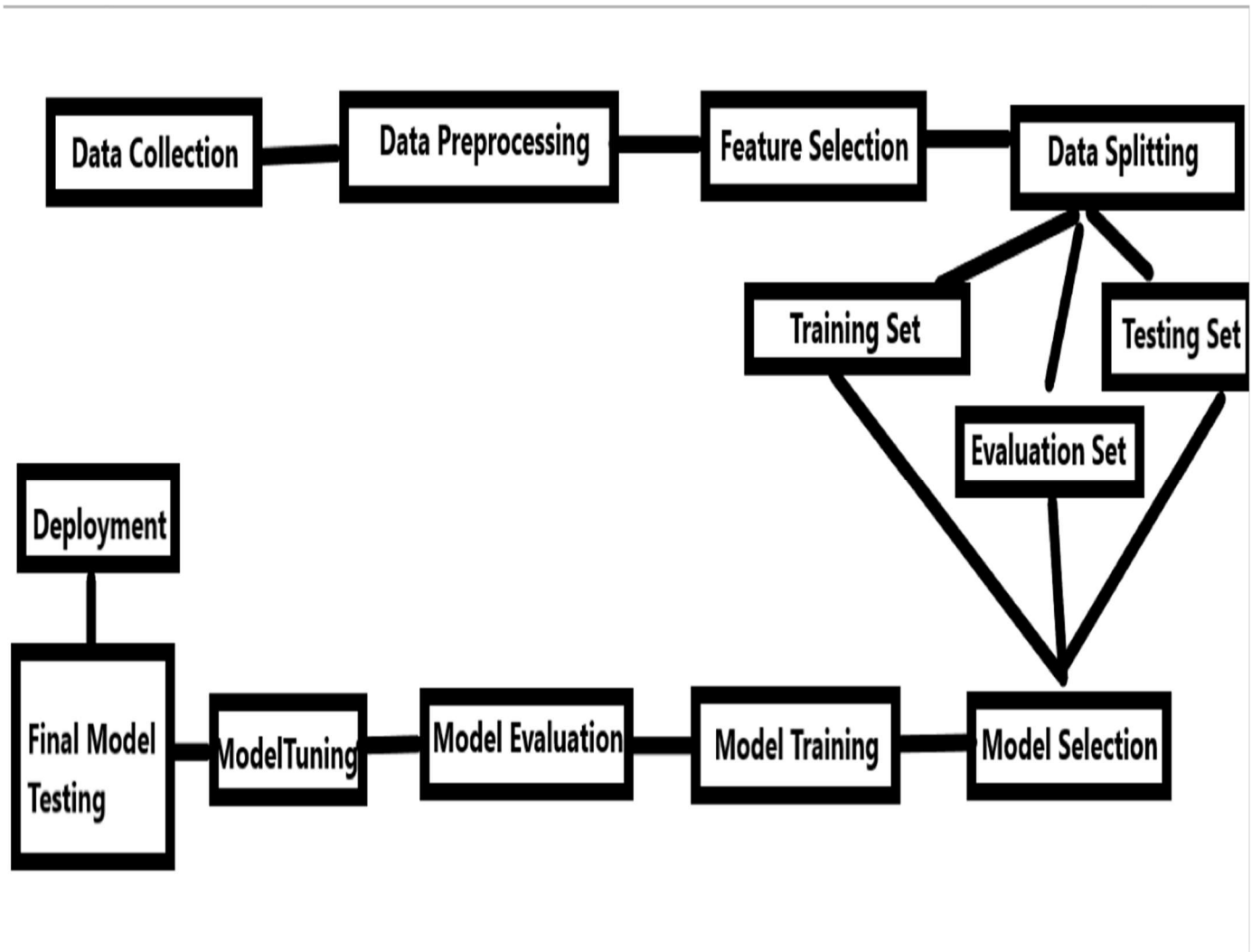
- Several machine learning algorithms can be used to develop a Chronic Kidney Diseases prediction model, including Logistic Regression, Random Forest, Decision Tree, Support Vector Machine, and Artificial Neural Network. The performance of the models is evaluated using different metrics, such as accuracy, precision, recall and F1 score.

## ❖ Feasibility Study

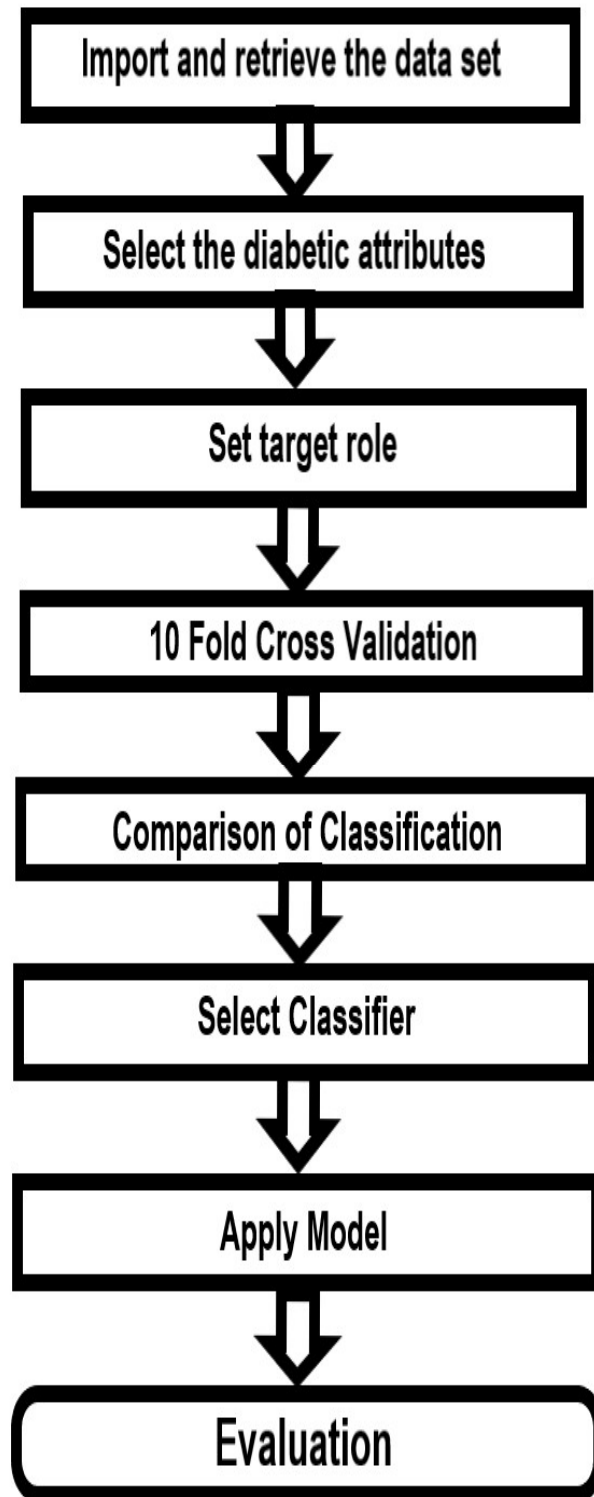
- In conclusion, developing a machine learning model for CKD prediction can be highly beneficial for patients and healthcare providers. It can aid in early detection and treatment, leading to improved patient outcomes and reduced healthcare costs. So, this is quite unique and useful model, hence this project is feasible.

# METHODOLOGY AND PLANNING

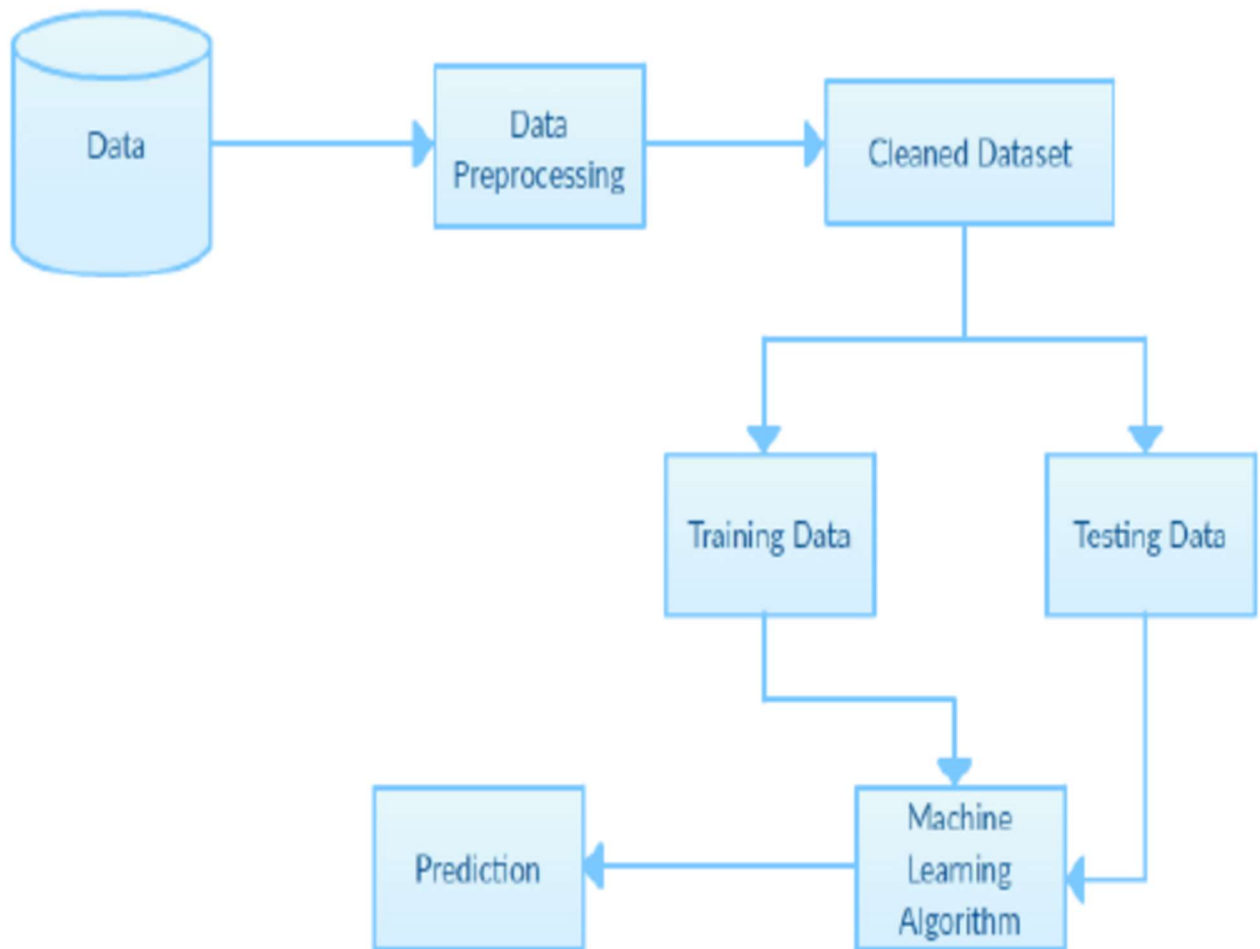
## ❖ Process Flow Chart



❖ ER Diagram



## ❖ Data Flow Diagram



# SYSTEM REQUIREMENTS

The application software is based on Chronic Kidney Disease prediction. The backend of the application is developed in Python and the frontend is developed in HTML5, CSS. The application can be accessed over any operating system.

## ❖ Installation

- Install Python: If python is not already installed on your system, you can download and install it from the official python website (<https://www.python.org/>). Make sure to choose the appropriate version of your operating system. Set and activate a virtual environment. Install required libraries and you are good to go.

## ❖ User Interface

- User interface design creates an effective communication medium between a human and a computer. Following asset of interface design principles, design identifies objects and actions and then creates a screen layout that forms the bases for a user interface prototype.
- Interface of Chronic Kidney Disease Prediction using Machine Learning is based on the following three principles:
  - Place the user in control.
  - Make the interface consistent.
  - Getting appropriate information from user for prediction.

## ❖ Minimum Requirements

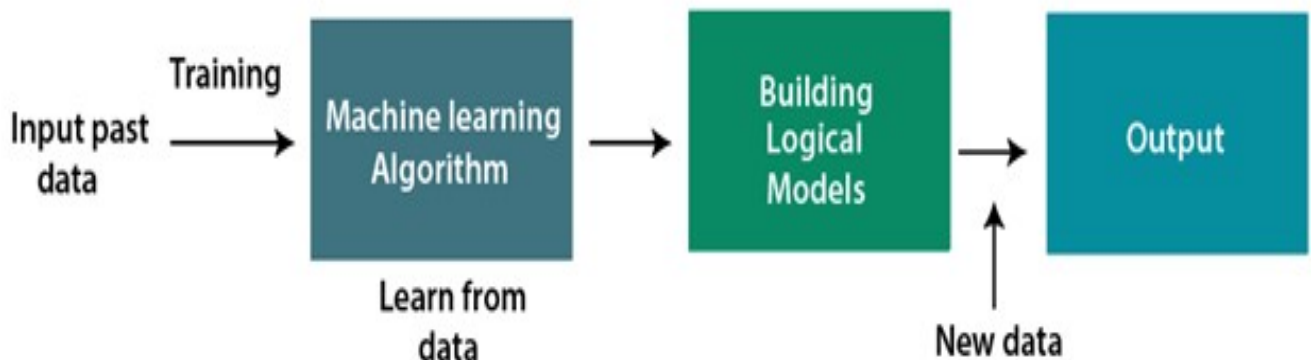
- Computer with a 2.27 GHz or faster processor (core i3)
- 3GB RAM or more
- 220 GB of available memory hard-disk space
- 5400 RPM hard drive
- 1024 x 768 or higher resolution display
- DVD-ROM Drive
- Windows 7 or above

# THEORY

## **Q. What is Machine Learning?**

**A.** Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information.

**“Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.”**

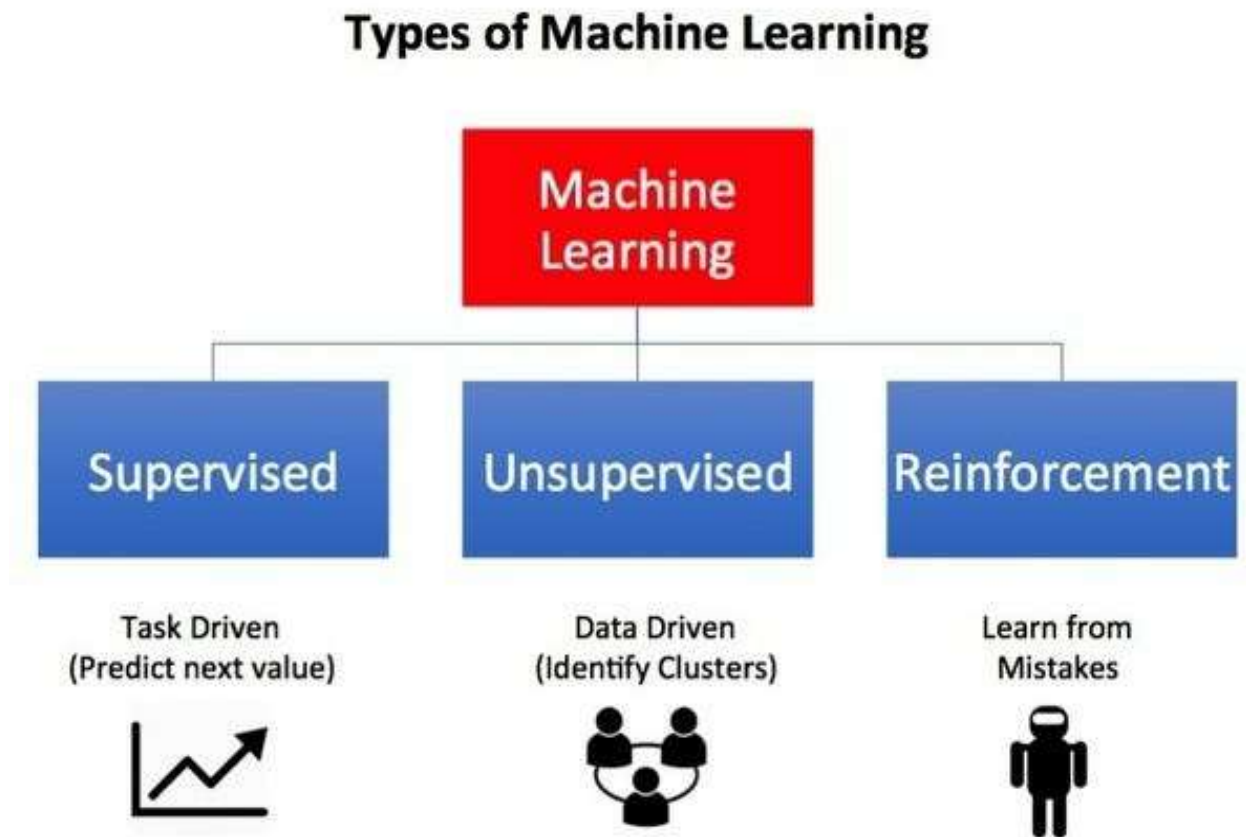


## **Q. What are the Types of Machine Learning?**

**A.** Machine learning is divided into mainly three types, which are:

1. **Supervised learning**
2. **Unsupervised learning**

### 3. Reinforcement learning



- 1. Supervised Learning:** Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output. The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The applications of Supervised learning are image segmentation, medical diagnosis, fraud detection, spam detection and speech recognition.
- 2. Unsupervised Learning:** Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has



not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. In unsupervised learning, we don't have a predetermined result. The examples of Unsupervised learning is clustering and association algorithms.

- 3. Reinforcement Learning:** Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance. The applications of Reinforcement learning is video games, resource management, robotics, text mining, etc.



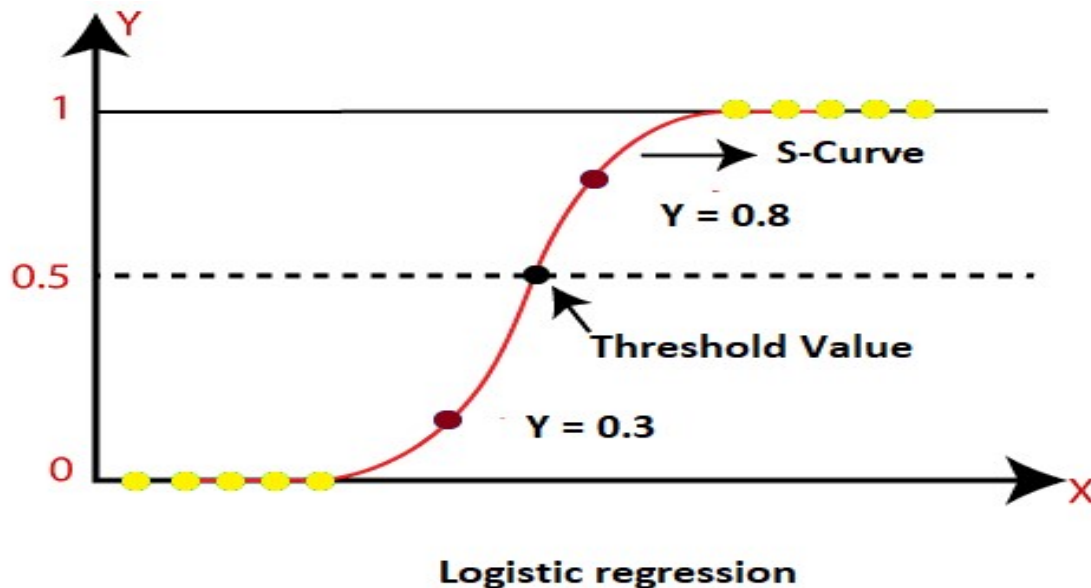
## **Q. How Machine learning works?**

**A.**

1. **A Decision Process:** In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labeled or unlabeled, your algorithm will produce an estimate about a pattern in the data.
2. **An Error Function:** An error function evaluates the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.
3. **A Model Optimization Process:** If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this “evaluate and optimize” process, updating weights autonomously until a threshold of accuracy has been met.

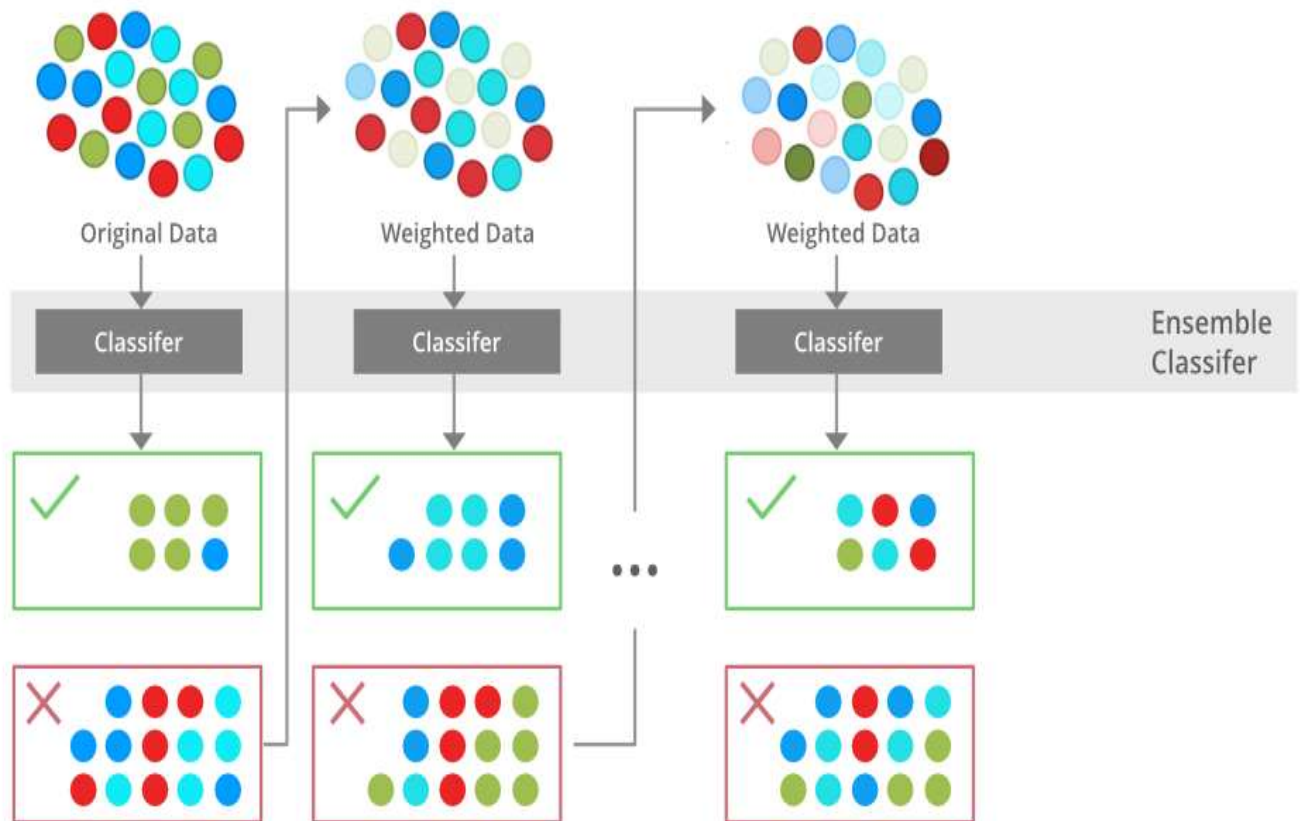
# Algorithms Used

1. **Logistic Regression:** Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.



2. **XGBoost:** XGBoost is an optimized distribution gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” and it has become one of the most popular and widely used machine learning algorithms due to its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression. One of the key features of XGBoost

is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. It can be used in applications like Kaggle competitions, click-through rate predictions, etc.



# EXPERIMENTAL ANALYSIS

- Code

- Training Code

```
• import pandas as pd
• from sklearn.preprocessing import LabelEncoder
• import numpy as np
• from sklearn.linear_model import LogisticRegression
• import pickle
• import random
```

## READING THE DATA

```
•
•
•
• df = pd.read_csv('kidney_disease.csv')
```

```
df = pd.read_csv(r'D:\GIT and Programming\CKD\kidney_disease.csv')
df.head()
```

✓ 7.6s

Python

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	class
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	
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	
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	
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	
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	

## CHANGING THE COLUMN NAME TO MAKE TO MORE USER-FRIENDLY

```
•
•
• columns = pd.read_csv(r"data_description.txt", sep='-')
• columns = columns.reset_index()
•
• columns.columns=['cols', 'abb_col_names']
•
• df.columns=columns['abb_col_names'].values
```

	id	age	blood pressure	specific gravity	albumin	sugar	red blood cells	pus cell	pus cell clumps	bacteria	...	packed cell volume	white blood cell count	red blood cell count	ypertension	di n
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	395	55.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	47	6700	4.9	no	
396	396	42.0	70.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	54	7800	6.2	no	
397	397	12.0	80.0	1.020	0.0	0.0	normal	normal	notpresent	notpresent	...	49	6600	5.4	no	
398	398	17.0	60.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	51	7200	5.9	no	
399	399	58.0	80.0	1.025	0.0	0.0	normal	normal	notpresent	notpresent	...	53	6800	6.1	no	

400 rows x 26 columns

```
•
•
• def convert_dtype(df, feature):
•     df[feature] = pd.to_numeric(df[feature], errors='coerce')
•
•
• features = ['packed cell volume', 'white blood cell count', 'red blood
• cell count']
• for feature in features:
•     convert_dtype(df, feature)
•
•
• df.drop('id', axis=1, inplace=True)
•
•
• def extract_cat_num(df):
•     cat_col = [col for col in df.columns if df[col].dtype == 'object']
•     num_col = [col for col in df.columns if df[col].dtype != 'object']
•     return cat_col, num_col
•
```

```

• cat_col, num_col = extract_cat_num(df)
•
• df['diabetes mellitus'].replace(to_replace={'\tno':'no', '\tyes':'yes', '
yes':'yes'}, inplace=True)
• df['coronary artery disease'].replace(to_replace={'\tno':'no'},
inplace=True)
• df['class'] = df['class'].replace(to_replace='ckd\t', value='ckd')
•
• data = df.copy()
•
• def assigning_missing_values(feature):
•     random_sample=data[feature].dropna().sample(data[feature].isnull().sum
• ())
•     random_sample.index = data[data[feature].isnull()].index
•     data.loc[data[feature].isnull(), feature] = random_sample
•
• for col in num_col:
•     assigning_missing_values(col)
•
• for col in cat_col:
•     assigning_missing_values(col)

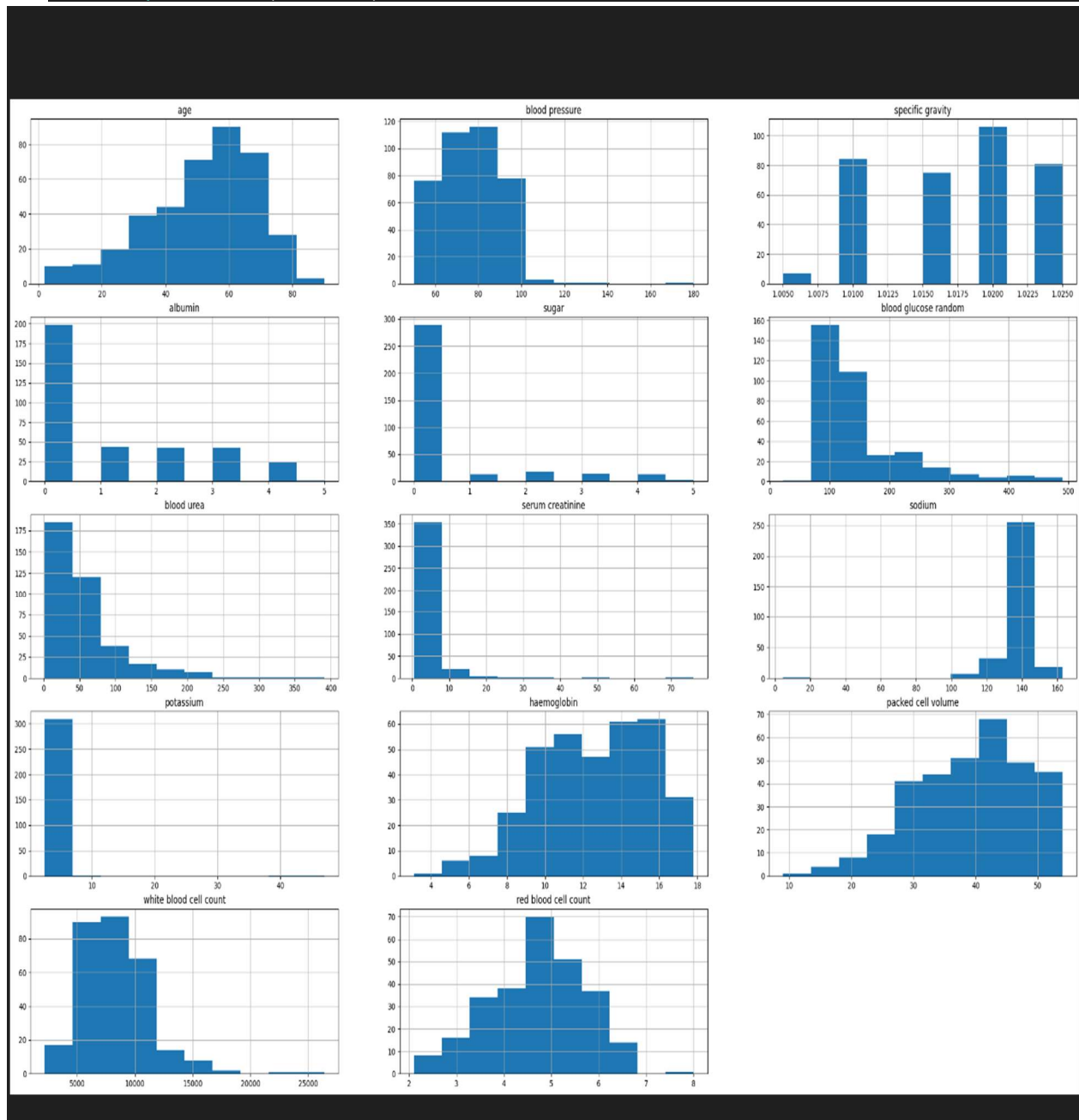
```

	age	blood pressure	specific gravity	albumin	sugar	red blood cells	pus cell	pus cell clumps	bacteria	blood glucose random	...	packed cell volume	white blood cell count	red blood cell count	ypertension	diabetes mellitus
0	48.0	80.0	1.020	1.0	0.0	1	1	0	0	121.0	...	44.0	7800.0	5.2	1	1
1	7.0	50.0	1.020	4.0	0.0	1	1	0	0	280.0	...	38.0	6000.0	4.7	0	0
2	62.0	80.0	1.010	2.0	3.0	1	1	0	0	423.0	...	31.0	7500.0	5.4	0	1
3	48.0	70.0	1.005	4.0	0.0	1	0	1	0	117.0	...	32.0	6700.0	3.9	1	0
4	51.0	80.0	1.010	2.0	0.0	1	1	0	0	106.0	...	35.0	7300.0	4.6	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	55.0	80.0	1.020	0.0	0.0	1	1	0	0	140.0	...	47.0	6700.0	4.9	0	0
396	42.0	70.0	1.025	0.0	0.0	1	1	0	0	75.0	...	54.0	7800.0	6.2	0	0
397	12.0	80.0	1.020	0.0	0.0	1	1	0	0	100.0	...	49.0	6600.0	5.4	0	0
398	17.0	60.0	1.025	0.0	0.0	1	1	0	0	114.0	...	51.0	7200.0	5.9	0	0
399	58.0	80.0	1.025	0.0	0.0	1	1	0	0	131.0	...	53.0	6800.0	6.1	0	0

400 rows × 25 columns

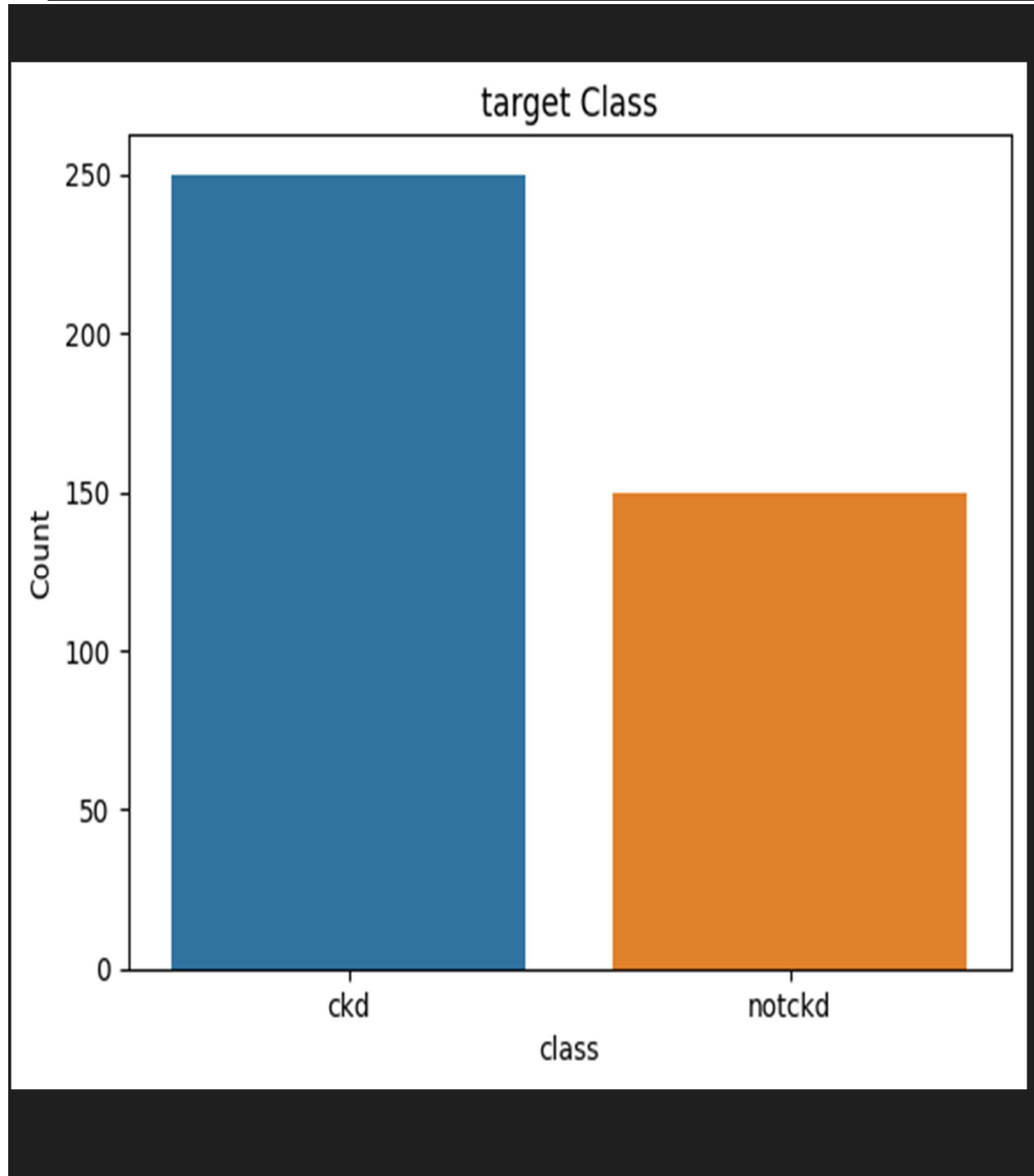
# STUDYING DATA AND PLOTTING GRAPH

- 
- 
- `plt.figure(figsize=(30,20))`
- `for i,feature in enumerate(num_col):`
- `plt.subplot(5,3,i+1)`
- `df[feature].hist()`
- `plt.title(feature)`



## PLOTTING GRAPH FOR CLASSIFICATION

- `sns.countplot(x='class',data=df)`
- `plt.xlabel("class")`
- `plt.ylabel("Count")`
- `plt.title("target Class")`





- 

## CORRELATION BETWEEN RED BLOOD CELL AND CLASSIFICATION

- ```
df.groupby(['red blood cells','class'])['red blood cell count'].agg(['count','mean','median','min','max'])
```

|                 |        | count | mean     | median | min | max |
|-----------------|--------|-------|----------|--------|-----|-----|
| red blood cells | class  |       |          |        |     |     |
| abnormal        | ckd    | 25    | 3.832000 | 3.7    | 2.5 | 5.6 |
| normal          | ckd    | 40    | 3.782500 | 3.8    | 2.1 | 8.0 |
|                 | notckd | 134   | 5.368657 | 5.3    | 4.4 | 6.5 |

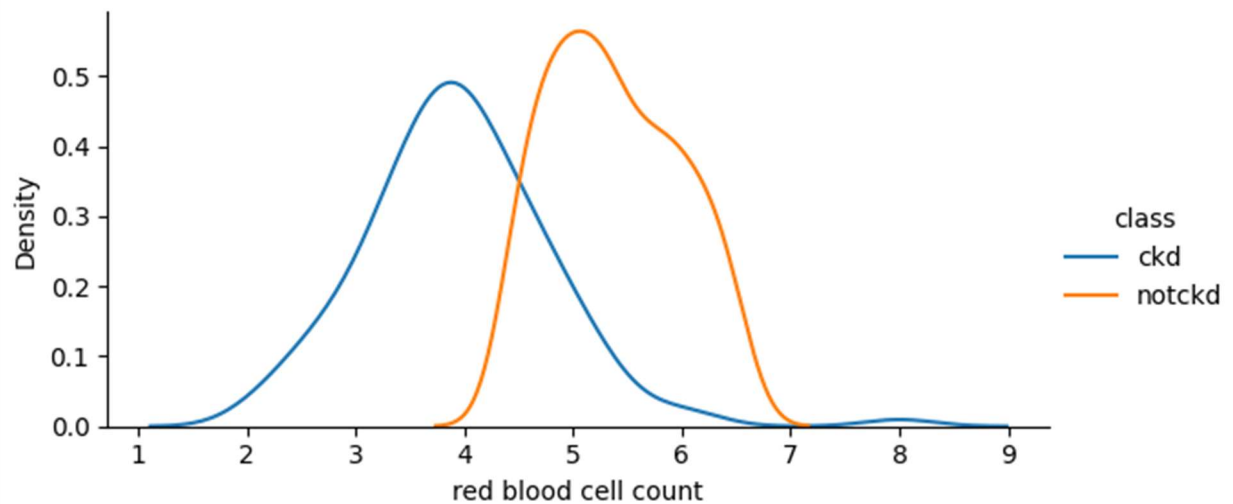
## CORRELATION BETWEEN RED BLOOD CELL COUNT AND CLASS

- ```
import plotly.express as px
```
- ```
px.violin(df,y='red blood cell count',x="class", color="class")
```



```
### analysing distribution of 'red blood cell count' in both Labels
```

```
grid=sns.FacetGrid(df, hue="class",aspect=2)  
grid.map(sns.kdeplot, 'red blood cell count')  
grid.add_legend()
```



- `scatters('red blood cell count', 'haemoglobin')`



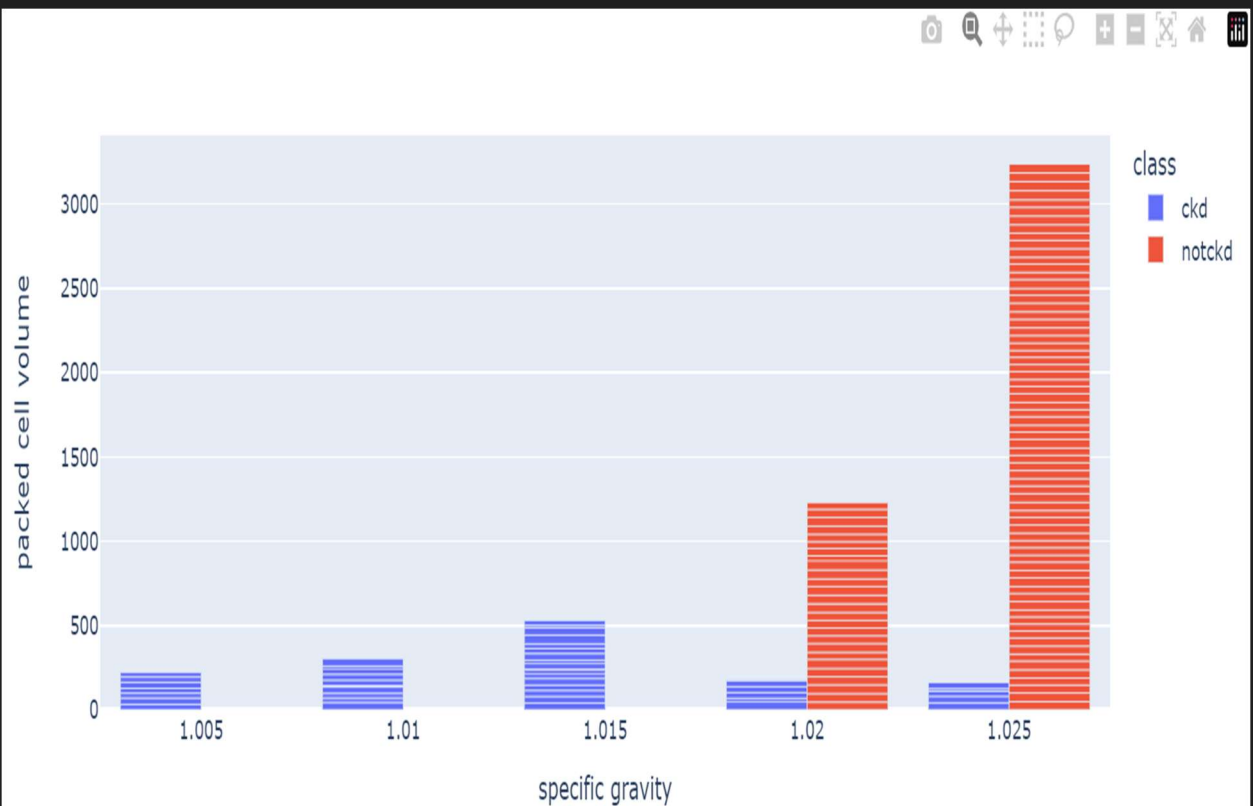
1.RBC count range ~2 to <4.5 and Hemoglobin between 3 to <13 are mostly classified as positive for chronic kidney disease(i.e ckd).

2.RBC count range >4.5 to ~6.1 and Hemoglobin between >13 to 17.8 are classified as negative for chronic kidney disease(i.e notckd).

```

•
•
• fig = px.bar(df, x="specific gravity", y="packed cell volume",
•           color='class', barmode='group',
•           height=400)
• fig.show()

```



```

•
•
• le = LabelEncoder()
•
•
• for col in cat_col:
•     data[col] = le.fit_transform(data[col])
•
•
• def data_split(entry, ratio):
•     # np.random.seed(42)
•     shuffled = np.random.permutation(len(entry))
•     test_set_size = int(len(entry) * ratio)
•     test_indices = shuffled[:test_set_size]
•     train_indices = shuffled[test_set_size:]
•     return data.iloc[train_indices], data.iloc[test_indices]

```

```

•
• if __name__ == "__main__":
•
•     # Read the data
•     # df = pd.read_csv('kidney_disease.csv')
•
•     train, test = data_split(data, 0.25)
•
•     X_train = train[['age', 'blood pressure', 'specific gravity',
• 'albumin', 'sugar',
•     'red blood cells', 'pus cell', 'pus cell clumps', 'bacteria',
•     'blood glucose random', 'blood urea', 'serum creatinine',
• 'sodium',
•     'potassium', 'haemoglobin', 'packed cell volume',
•     'white blood cell count', 'red blood cell count', 'ypertension',
•     'diabetes mellitus', 'coronary artery disease', 'appetite',
•     'pedal edema', 'anemia']].to_numpy()
•
•     X_test = test[['age', 'blood pressure', 'specific gravity', 'albumin',
• 'sugar',
•     'red blood cells', 'pus cell', 'pus cell clumps', 'bacteria',
•     'blood glucose random', 'blood urea', 'serum creatinine',
• 'sodium',
•     'potassium', 'haemoglobin', 'packed cell volume',
•     'white blood cell count', 'red blood cell count', 'ypertension',
•     'diabetes mellitus', 'coronary artery disease', 'appetite',
•     'pedal edema', 'anemia']].to_numpy()
•
•     Y_train = train[['class']].to_numpy().reshape(300,)
•     Y_test = test[['class']].to_numpy().reshape(100,)
•
•     clf = LogisticRegression(solver='lbfgs', max_iter=10000000)
•     clf.fit(X_train, Y_train)
•
•     file = open('model.pkl', 'wb')
•
•     pickle.dump(clf, file)
•
•     input_features =
[5.0, 70.0, 1.025, 1.0, 0.0, 0, 1, 0, 0, 97.0,
•     56, 3.8, 111, 2.5, 11.2,
•     34.0, 7200.0, 4.1, 0, 1, 0, 1, 0, 0]
•     ckd_prob = clf.predict_proba([input_features])[0][1]
•
•     print(ckd_prob)

```

## ○ Main Code

### USING FLASK FRAMEWORK TO CONNET THE BACKEND WITH FRONTEND

```
•
• from flask import Flask, render_template, request
• app = Flask(__name__)
• import pickle
• import random
•
• file = open('model.pkl', 'rb')
•
• clf = pickle.load(file)
• file.close()
•
• @app.route('/', methods=["GET", "POST"])
• def hello_world():
•     if request.method == "POST":
•         MyDic = request.form
•
•         Age = int(request.form['Age'])
•         BloodPressure = float(request.form['BloodPressure'])
•         SpecificGravity = float(MyDic['SpecificGravity'])
•         Albumin = float(MyDic['Albumin'])
•         Diabetes = float(MyDic['Diabetes'])
•         RedBloodCells = int(MyDic['RedBloodCells'])
•         PusCells = int(MyDic['PusCells'])
•         PusCellClumps = int(MyDic['PusCellClumps'])
•         Bacteria = int(MyDic['Bacteria'])
•         k = round(random.uniform(15,95),2)
•         BloodGlucoseRandom = float(MyDic['BloodGlucoseRandom'])
•         BloodUrea = float(MyDic['BloodUrea'])
•         SerumCreatinine = float(MyDic['SerumCreatinine'])
•         Sodium = float(MyDic['Sodium'])
•         Potasium = float(MyDic['Potasium'])
•         Haemoglobin = float(MyDic['Haemoglobin'])
•         PackedCellVolume = float(MyDic['PackedCellVolume'])
•         whiteBloodCellCount = float(MyDic['whiteBloodCellCount'])
•         RedBloodCellCount = float(MyDic['RedBloodCellCount'])
•         Y_pertension = int(MyDic['Y_pertension'])
•         DiabetesMellitus = int(MyDic['DiabetesMellitus'])
•         CoronaryArteryDisease = int(MyDic['CoronaryArteryDisease'])
```

```

•     Appetite = int(MyDic['Appetite'])
•     PedalEdema = int(MyDic['PedalEdema'])
•     Anemia = int(MyDic['Anemia'])
•
•     input_features = [Age, BloodPressure, SpecificGravity,
Albumin, Diabetes, RedBloodCells, PusCells, PusCellClumps, Bacteri
a, BloodGlucoseRandom,
•         BloodUrea, SerumCreatinine, Sodium, Potasium, Haemoglobin,
•         PackedCellVolume, whiteBloodCellCount, RedBloodCellCo
unt, Y_pertension, DiabetesMellitus, CoronaryArteryDisease, Appetite
, PedalEdema, Anemia]
•     ckd_prob = clf.predict_proba([input_features])[0][1]
•     print(input_features)
•     print(ckd_prob)
•
•     return render_template('result.html', ckd = k, Age =Age,
BloodPressure=BloodPressure, SpecificGravity=SpecificGravity, Albumin =
Albumin, Diabetes=
Diabetes, RedBloodCells=RedBloodCells, PusCells=PusCells, PusCellClu
mps=PusCellClumps, Bacteria=Bacteria, BloodGlucoseRandom=BloodGlucoseR
andom,
•         BloodUrea=BloodUrea, SerumCreatinine=SerumCreatinine,
Sodium=Sodium, Potasium=Potasium , Haemoglobin=Haemoglobin,
•         PackedCellVolume=PackedCellVolume, whiteBloodCellCount=wh
iteBloodCellCount, RedBloodCellCount=RedBloodCellCount, Y_pertension
=Y_pertension, DiabetesMellitus=DiabetesMellitus, CoronaryArteryDisease=
CoronaryArteryDisease, Appetite=Appetite, PedalEdema=PedalEdema,
Anemia=Anemia )
•     return render_template('index.html')
•
• if __name__ == "__main__":
•     app.run(debug=True)
•
•

```

- Frontend Code

- Form Page

```
• <body>
•   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
•     <div class="container-fluid">
•       <a class="navbar-brand" href="#">
•         <h2 class="title">Chronic Kidney Disease Probability
Detector</h2>
•       </a>
•       <!-- <button
•         class="navbar-toggler"
•         type="button"
•         data-bs-toggle="collapse"
•         data-bs-target="#navbarNavAltMarkup"
•         aria-controls="navbarNavAltMarkup"
•         aria-expanded="false"
•         aria-label="Toggle navigation"
•       >
•         <span class="navbar-toggler-icon"></span>
•       </button> -->
•       <!-- <div class="collapse navbar-collapse"
id="navbarNavAltMarkup">
•         <div class="navbar-nav"> -->
•           <!-- <a class="nav-link active" aria-current="page"
href="#">Home</a> -->
•           <!-- <a class="nav-link" href="#">Features</a> -->
•           <!-- <a class="nav-link" href="#">Pricing</a> -->
•           <!-- <a class="nav-link disabled">Disabled</a> -->
•           <!-- </div>
•         </div> -->
•       </div>
•     </nav>
•     <div class="headingDiv">
•       <h2 class="heading">Fill this form</h2>
•     </div>
•     <form action="/" method="POST">
•       <div class="left">
•         <div class="input-group mb-1 form-group Age common">
•           <label for="Age" class="input-group-text" id="Age">Age</label>
•           <input
•             type="number"
•             name="Age"
```

```

•         class="form-control"
•         aria-label="Sizing example input"
•         aria-describedby="inputGroup-sizing-default"
•     />
• </div>
•
• <div class="input-group mb-1 BP common">
•     <span class="input-group-text" id="BloodPressure">
•         >Blood Pressure</span>
•     >
•     <input
•         type="text"
•         name="BloodPressure"
•         class="form-control"
•         aria-label="Sizing example input"
•         aria-describedby="inputGroup-sizing-default"
•     />
• </div>
• <div class="input-group mb-3 BP common">
•     <label class="input-group-text" for="inputGroupSelect02">
•         >Specific Gravity</label>
•     >
•     <select
•         class="form-select"
•         id="SpecificGravity"
•         name="SpecificGravity"
•     >
•         <option selected>Choose...</option>
•         <option value="0">1.005</option>
•         <option value="1">1.010</option>
•         <option value="2">1.015</option>
•         <option value="3">1.020</option>
•         <option value="4">1.025</option>
•         <!-- <option value="5">STAGE 5: SEVERE
DECOMPENSATION.</option> -->
•     </select>
•     <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
• </div>
• <div class="input-group mb-3 BP common">
•     <label class="input-group-text" for="inputGroupSelect02">
•         >Albumin</label>
•     >
•     <select class="form-select" id="Albumin" name="Albumin">
•         <option selected>Choose...</option>

```



```

•         <option value="0">0</option>
•         <option value="1">1</option>
•         <option value="2">2</option>
•         <option value="3">3</option>
•         <option value="4">4</option>
•         <option value="5">5</option>
•     </select>
•     <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
• </div>
•
• <div class="input-group mb-3 BP common">
•     <label class="input-group-text" for="inputGroupSelect02"
•         >Diabetes</label>
•     >
•     <select class="form-select" id="Diabetes" name="Diabetes">
•         <option selected>Choose...</option>
•         <option value="0">Non Diabetic</option>
•         <option value="1">STAGE 1: COMPENSATION</option>
•         <option value="2">STAGE 2: STABLE ADAPTATION</option>
•         <option value="3">STAGE 3: UNSTABLE EARLY
DECOMPENSATION</option>
•         <option value="4">STAGE 4: STABLE DECOMPENSATION.</option>
•         <option value="5">STAGE 5: SEVERE DECOMPENSATION.</option>
•     </select>
•     <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
• </div>
• <div class="input-group mb-3 BP common">
•     <label class="input-group-text" for="inputGroupSelect02"
•         >Red Blood Cells</label>
•     >
•     <select class="form-select" id="RedBloodCells"
name="RedBloodCells">
•         <option selected>Choose...</option>
•         <option value="0">Abnormal</option>
•         <option value="1">Normal</option>
•     </select>
•     <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
• </div>
• <div class="input-group mb-3 BP common">
•     <label class="input-group-text" for="inputGroupSelect02"
•         >Pus Cells</label>
•     >

```

```

•         <select class="form-select" id="PusCells" name="PusCells">
•             <option selected>Choose...</option>
•             <option value="0">Abnormal</option>
•             <option value="1">Normal</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Pus Cell Clumps</label>
•         >
•         <select class="form-select" id="PusCellClumps"
name="PusCellClumps">
•             <option selected>Choose...</option>
•             <option value="0">Not Present</option>
•             <option value="1">Present</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Bacteria</label>
•         >
•         <select class="form-select" id="Bacteria" name="Bacteria">
•             <option selected>Choose...</option>
•             <option value="0">Not Present</option>
•             <option value="1">Present</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
•     <div class="input-group mb-1 BP common">
•         <span class="input-group-text" id="BloodGlucoseRandom"
•             >Blood Glucose Random</span>
•         >
•         <input
•             type="text"
•             name="BloodGlucoseRandom"
•             class="form-control"
•             aria-label="Sizing example input"
•             aria-describedby="inputGroup-sizing-default"
•         />
•     </div>

```

```
• <div class="input-group mb-1 BP common">
•   <span class="input-group-text" id="BloodUrea">Blood Urea</span>
•   <input
•     type="text"
•     name="BloodUrea"
•     class="form-control"
•     aria-label="Sizing example input"
•     aria-describedby="inputGroup-sizing-default"
•   />
• </div>
• <div class="input-group mb-1 BP common">
•   <span class="input-group-text" id="SerumCreatinine"
•     >Serum Creatinine</span>
•   >
•   <input
•     type="text"
•     name="SerumCreatinine"
•     class="form-control"
•     aria-label="Sizing example input"
•     aria-describedby="inputGroup-sizing-default"
•   />
• </div>
• </div>
• <div class="right">
•   <div class="input-group mb-1 BP common">
•     <span class="input-group-text" id="Sodium">Sodium</span>
•     <input
•       type="text"
•       name="Sodium"
•       class="form-control"
•       aria-label="Sizing example input"
•       aria-describedby="inputGroup-sizing-default"
•     />
•   </div>
•   <div class="input-group mb-1 BP common">
•     <span class="input-group-text" id="Potasium">Potasium</span>
•     <input
•       type="text"
•       name="Potasium"
•       class="form-control"
•       aria-label="Sizing example input"
•       aria-describedby="inputGroup-sizing-default"
•     />
•   </div>
•   <div class="input-group mb-1 BP common">
```

```
•      <span class="input-group-text"
id="Haemoglobin">Haemoglobin</span>
•      <input
•        type="text"
•        name="Haemoglobin"
•        class="form-control"
•        aria-label="Sizing example input"
•        aria-describedby="inputGroup-sizing-default"
•      />
•    </div>
•    <div class="input-group mb-1 BP common">
•      <span class="input-group-text" id="PackedCellVolume"
•        >Packed Cell Volume</span>
•      >
•      <input
•        type="text"
•        name="PackedCellVolume"
•        class="form-control"
•        aria-label="Sizing example input"
•        aria-describedby="inputGroup-sizing-default"
•      />
•    </div>
•    <div class="input-group mb-1 BP common">
•      <span class="input-group-text" id="whiteBloodCellCount"
•        >White Blood Cell Count</span>
•      >
•      <input
•        type="text"
•        name="whiteBloodCellCount"
•        class="form-control"
•        aria-label="Sizing example input"
•        aria-describedby="inputGroup-sizing-default"
•      />
•    </div>
•    <div class="input-group mb-1 BP common">
•      <span class="input-group-text" id="RedBloodCellCount"
•        >Red Blood Cell Count</span>
•      >
•      <input
•        type="text"
•        name="RedBloodCellCount"
•        class="form-control"
•        aria-label="Sizing example input"
•        aria-describedby="inputGroup-sizing-default"
•      />
•    </div>
```

```

•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Y-pertension</label>
•         >
•         <select class="form-select" id="Y_pertension"
name="Y_pertension">
•             <option selected>Choose...</option>
•             <option value="0">No</option>
•             <option value="1">Yes</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Diabetes Mellitus</label>
•         >
•         <select
•             class="form-select"
•             id="DiabetesMellitus"
•             name="DiabetesMellitus"
•         >
•             <option selected>Choose...</option>
•             <option value="0">No</option>
•             <option value="1">Yes</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Coronary Artery Disease</label>
•         >
•         <select
•             class="form-select"
•             id="CoronaryArteryDisease"
•             name="CoronaryArteryDisease"
•         >
•             <option selected>Choose...</option>
•             <option value="0">No</option>
•             <option value="1">Yes</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->

```

```

•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Appetite</label>
•         >
•         <select class="form-select" id="Appetite" name="Appetite">
•             <option selected>Choose...</option>
•             <option value="0">Good</option>
•             <option value="1">Poor</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Pedal Edema</label>
•         >
•         <select class="form-select" id="PedalEdema" name="PedalEdema">
•             <option selected>Choose...</option>
•             <option value="0">No</option>
•             <option value="1">Yes</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
•     <div class="input-group mb-3 BP common">
•         <label class="input-group-text" for="inputGroupSelect02"
•             >Anemia</label>
•         >
•         <select class="form-select" id="Anemia" name="Anemia">
•             <option selected>Choose...</option>
•             <option value="0">No</option>
•             <option value="1">Yes</option>
•         </select>
•         <!-- <label class="input-group-text"
for="inputGroupSelect02">Options</label> -->
•     </div>
• </div>
• <button class="btn btn-primary">Submit</button>
• </form>
• <script
•     src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"
•     integrity="sha384-
ENjd04Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdvQSZ"

```

- `crossorigin="anonymous"`
- `></script>`
- `</body>`

## Chronic Kidney Disease Probability Detector

Fill this form

Age

Blood Pressure

Specific Gravity

Albumin

Diabetes

Red Blood Cells

Pus Cells

Pus Cell Clumps

Bacteria

Blood Glucose Random

Blood Urea

Serum Creatinine

Sodium

Potassium

Haemoglobin

Packed Cell Volume

White Blood Cell Count

Red Blood Cell Count

Y-pertension

Diabetes Mellitus

Coronary Artery Disease

Appetite

Pedal Edema

Anemia

Submit

- Result Page

```
<body onload="changeColor()">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">
                <h2>Chronic Kidney Disease Probability Detector</h2>
            </a>
            <button
                class="navbar-toggler"
                type="button"
                data-bs-toggle="collapse"
                data-bs-target="#navbarNavAltMarkup"
                aria-controls="navbarNavAltMarkup"
                aria-expanded="false"
                aria-label="Toggle navigation"
            >
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
                <div class="navbar-nav">
                    <!-- <a class="nav-link active" aria-current="page"
href="#">Home</a> -->
                    <!-- <a class="nav-link" href="#">Features</a> -->
                    <!-- <a class="nav-link" href="#">Pricing</a> -->
                    <!-- <a class="nav-link disabled">Disabled</a> -->
                </div>
            </div>
        </div>
    </nav>
    <div class="whole"></div>
    <div class="details2"></div>
    <div class="details">
        <h4 class="entry">
            Age = {{Age}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Blood
            Pressure={{BloodPressure}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Specific
            Gravity={{SpecificGravity}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& <br />
            Albumin = {{Albumin}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Diabetes=
            {{Diabetes}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Red Blood
            Cells={{RedBloodCells}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& <br />
            Pus Cells={{PusCells}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& Pus Cell
            Clumps={{PusCellClumps}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
            Bacteria={{Bacteria}}&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& <br />
            Blood Glucose Random={{BloodGlucoseRandom}}&nbsp;&nbsp;&nbsp;&nbsp;&~Blood
            Urea={{BloodUrea}}&nbsp;&nbsp;&nbsp;&nbsp;&~<br />
```



[illegible]

# Outputs

## Chronic Kidney Disease Probability Detector

**Fill this form**

Age	48	Sodium	111
Blood Pressure	70	Potassium	2.5
Specific Gravity	1.005	Haemoglobin	11.2
Albumin	4	Packed Cell Volume	32
Diabetes	Non Diabetic	White Blood Cell Count	6700
Red Blood Cells	Normal	Red Blood Cell Count	3.9
Pus Cells	Abnormal	Y-pertension	Yes
Pus Cell Clumps	Present	Diabetes Mellitus	No
Bacteria	Not Present	Coronary Artery Disease	No
Blood Glucose Random	117	Appetite	Poor
Blood Urea	56	Pedal Edema	Yes
Serum Creatinine	3.8	Anemia	Yes

## Chronic Kidney Disease Probability Detector

Age = 48   Blood Pressure=70.0   Specific Gravity=0.0  
Albumin = 4.0   Diabetes= 0.0   Red Blood Cells=1  
Pus Cells=0   Pus Cell Clumps=1   Bacteria=0  
Blood Glucose Random=117.0   Blood Urea=56.0  
Serum Creatinine=3.8   Sodium=111.0   Potassium=2.5  
Haemoglobin=11.2   Packed Cell Volume=32.0  
white Blood Cell Count=6700.0   Red Blood Cell Count=3.9  
Y\_pertension=1   Diabetes Mellitus=0   Anemia=1  
Appetite=1   Pedal Edema=1   Coronary Artery Disease=0

**Infection Probability = 85.15**

[Home](#)

## Chronic Kidney Disease Probability Detector

Fill this form

Age	71	Sodium	145
Blood Pressure	70	Potassium	5
Specific Gravity	1.020	Haemoglobin	14.7
Albumin	0	Packed Cell Volume	44
Diabetes	Non Diabetic	White Blood Cell Count	9800
Red Blood Cells	Normal	Red Blood Cell Count	6
Pus Cells	Normal	Y-pertension	No
Pus Cell Clumps	Not Present	Diabetes Mellitus	No
Bacteria	Not Present	Coronary Artery Disease	No
Blood Glucose Random	81	Appetite	Good
Blood Urea	18	Pedal Edema	No
Serum Creatinine	0.8	Anemia	No

## Chronic Kidney Disease Probability Detector

Age = 71   Blood Pressure=70.0   Specific Gravity=3.0  
Albumin = 0.0   Diabetes= 0.0   Red Blood Cells=1  
Pus Cells=1   Pus Cell Clumps=0   Bacteria=0  
Blood Glucose Random=81.0   Blood Urea=18.0  
Serum Creatinine=0.8   Sodium=145.0   Potassium=5.0  
Haemoglobin=14.7   Packed Cell Volume=44.0  
white Blood Cell Count=9800.0   Red Blood Cell Count=6.0  
Y\_pertension=0   Diabetes Mellitus=0   Anemia=0  
Appetite=0   Pedal Edema=0   Coronary Artery Disease=0

Infection Probability = 23.7

[Home](#)

# FUTURE ENHANCEMENTS

**Data Integration:** Incorporating a broader range of data sources, such as electronic health records, genetic information, lifestyle data, and environmental factors, can enhance the accuracy of the predictor. This would require collaboration with healthcare providers, research institutions, and other relevant sources to gather comprehensive and diverse datasets.

**Machine Learning Algorithms:** Exploring more advanced machine learning techniques, such as deep learning, ensemble methods, or reinforcement learning, can potentially improve the performance of the predictor. These algorithms may be better equipped to extract complex patterns and relationships from large-scale datasets, leading to more accurate predictions.

**Real-time Monitoring:** Integrating real-time monitoring devices and wearable technologies can provide continuous data streams, enabling dynamic assessment of kidney health. This could involve tracking blood pressure, heart rate, glucose levels, and other relevant parameters. Such continuous monitoring can enhance the predictive capabilities of the system by capturing fluctuations and trends over time.

**Personalized Risk Assessment:** Developing personalized risk assessment models by considering individual patient characteristics, including age, gender, medical history, and comorbidities, can enhance the accuracy of predictions. This would involve tailoring the predictor to specific patient profiles, allowing for more precise risk estimation and personalized interventions.

**Explainability and Interpretability:** Addressing the black-box nature of predictive models is crucial to gain trust and acceptance from healthcare professionals and patients. Developing methods to explain and interpret the predictions can provide insights into the underlying factors

contributing to the risk, aiding in decision-making and treatment planning.

**Longitudinal Analysis:** Incorporating longitudinal data analysis can enable the prediction of disease progression and help identify critical time points for intervention. By analyzing changes in biomarkers, clinical parameters, and lifestyle factors over time, the predictor can provide early warnings and facilitate timely interventions to slow down or prevent kidney disease progression.

**External Validation:** Conducting extensive external validation studies with diverse populations and healthcare settings can ensure the generalizability and reliability of the predictor. Collaborating with multiple institutions and involving large cohorts of patients can help validate and refine the predictive model's performance.

**User-Friendly Interface:** Designing a user-friendly interface that presents the predictions and risk factors in a clear and understandable manner is essential. This would enable healthcare professionals and patients to easily interpret and act upon the predictor's results, fostering effective communication and shared decision-making.

**Integration with Clinical Guidelines:** Integrating the predictor with existing clinical guidelines and decision support systems can help clinicians make informed decisions about patient management and treatment plans. Aligning the predictor's recommendations with evidence-based practices can enhance its utility in real-world healthcare settings.

# BIBLIOGRAPHY

- <https://www.analyticsvidhya.com/blog/2022/01/predicting-chronic-kidney-disease-using-machine-learning/>
- <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00657-5>
- [https://en.wikipedia.org/wiki/Chronic\\_kidney\\_disease](https://en.wikipedia.org/wiki/Chronic_kidney_disease)
- <https://journals.plos.org/digitalhealth/article?id=10.1371/journal.pdig.0000188>
- Kaur, R., Singh, H., & Kaur, M. (2020). A comprehensive review of machine learning techniques for chronic kidney disease prediction. Computer methods and program in Biomedicine, 187.
- <https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2016/Issue-29/Article29.pdf>
- <https://bmcnephrol.biomedcentral.com/articles/10.1186/s12882-021-02474-z>
- <https://www.mdpi.com/2504-2289/6/3/98>
- [researchgate.net/publication/329395701\\_Prediction\\_of\\_Chronic\\_Kidney\\_Disease\\_Using\\_Machine\\_Learning\\_Algorithm](https://researchgate.net/publication/329395701_Prediction_of_Chronic_Kidney_Disease_Using_Machine_Learning_Algorithm)
- <https://www.javatpoint.com/machine-learning>
- <https://www.ibm.com/topics/machine-learning>

