MACHINE LEARNING PROJECT-2024

"KIDNEY DISEASE PREDICTION"

Submitted to:

Dr. Sumit Kumar

(Assistant Professor, CSED)

Submitted by:

Abhaijeet Singh – (102217186)

Uday Pratap Singh Atlwal – (102217224)

INDEX

INTRODUCTION

Objective:

The Kidney Disease Prediction project uses machine learning to accurately predict whether a person suffers from kidney disease based on their medical data. The model analyses various patient attributes such as age, blood pressure, specific gravity, albumin levels, and numerous other clinical factors. These include blood urea, serum creatinine, sodium, potassium levels, and more.

Dataset Citation:

The dataset we use is provided by the **University of California, Irvine'**s Machine Learning Laboratory, under a CC-BY license released in 2015.

Dataset Link: https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease

Significance:

The model can detect patterns and correlations indicative of kidney disease by processing these features.

We train the model on historical patient data, and it learns to differentiate between healthy individuals and those likely to have kidney disease. Our ultimate goal is to assist healthcare professionals in early diagnosis, enabling timely intervention and treatment.

This model could be a valuable tool in clinic settings helping us identify kidney-related diseases early.

Scope:

We employ multiple machine learning algorithms, and we find that the Random Forest Classifier (RFC) gives us the best results. Random Forest creates multiple decision trees and combines their outputs to make predictions, ensuring better generalisation and reducing the risk of overfitting. This technique is particularly suitable for medical data, where relationships between variables may not always be linear or straightforward.

DATASET ATTRIBUTES

The dataset for training the kidney disease prediction model contains 400 patient records with 24 medical attributes and a target class (class). The dataset includes the following attributes:

Attributes:

1. **Numerical Attributes:**
    - **Age**: Age of the patient (in years), with 391 valid entries.
    - **Blood Pressure**: Measured in mmHg, indicating the patient's blood pressure, with 388 non-null values.
    - **Specific Gravity**: Represents the urine concentration, having 353 valid records.
    - **Albumin**: Protein level in urine, with 354 entries.
    - **Sugar**: Sugar level in urine, with 351 valid data points.
    - **Blood Glucose Random**: Random blood glucose levels, with 356 valid entries.
    - **Blood Urea**: Amount of urea in the blood, available for 381 samples.
    - **Serum Creatinine**: Creatinine concentration in blood, with 383 valid records.
    - **Sodium & Potassium**: Sodium (313 non-null) and potassium (312 non-null) are important for electrolyte balance.
    - **Haemoglobin**: Haemoglobin levels in blood, with 348 entries.
    - **Packed Cell Volume**: Indicates the proportion of red blood cells in the blood (330 non-null, treated as numeric after preprocessing).
    - **White Blood Cell Count**: Count of white blood cells, with 295 non-null entries.
    - **Red Blood Cell Count**: Number of red blood cells with 270 valid entries.
2. **Categorical Attributes:**
    - **Red Blood Cells**: Qualitative measure of RBCs, with 248 non-null values.
    - **Pus Cell & Pus Cell Clumps**: Qualitative urine analysis attributes, with 335 and 396 valid records, respectively.
    - **Bacteria**: Presence of bacteria in urine, with 396 non-null values.
    - **Hypertension**: Indicates if the patient has high blood pressure, with 398 non-null entries.
    - **Diabetes Mellitus**: Indicates diabetes status, with 398 non-null values.

- **Coronary Artery Disease**: Shows the presence of CAD, with 398 valid entries.
- **Appetite**: Reflects the patient's appetite, with 399 non-null values.
- **Pedal Edema**: Indicates swelling in legs, with 399 valid records.
- **Anemia**: Reflects the presence of anemia, with 399 entries.

3. **Target Class:**
   - **Class**: Binary classification indicating whether the patient has kidney disease (1) or not (0). This attribute is fully populated (400 non-null).

Image of dataset's first 30 rows:

id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	48	80	1.02	1	0		normal	notprese	notprese	121	36	1.2			15.4	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	7	50	1.02	4	0		normal	notprese	notpresent		18	0.8			11.3	38	6000		no	no	no	good	no	no	ckd
2	62	80	1.01	2	3	normal	normal	notprese	notprese	423	53	1.8			9.6	31	7500		no	yes	no	poor	no	yes	ckd
3	48	70	1.005	4	0	normal	abnorma	present	notprese	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	51	80	1.01	2	0	normal	normal	notprese	notprese	106	26	1.4			11.6	35	7300	4.6	no	no	no	good	no	no	ckd
5	60	90	1.015	3	0			notprese	notprese	74	25	1.1	142	3.2	12.2	39	7800	4.4	yes	yes	no	good	yes	no	ckd
6	68	70	1.01	0	0		normal	notprese	notprese	100	54	24	104	4	12.4	36			no	no	no	good	no	no	ckd
7	24		1.015	2	4	normal	abnorma	notprese	notprese	410	31	1.1			12.4	44	6900	5	no	yes	no	good	yes	no	ckd
8	52	100	1.015	3	0	normal	abnorma	present	notprese	138	60	1.9			10.8	33	9600	4	yes	yes	no	good	no	yes	ckd
9	53	90	1.02	2	0	abnorma	abnorma	present	notprese	70	107	7.2	114	3.7	9.5	29	12100	3.7	yes	yes	no	poor	no	yes	ckd
10	50	60	1.01	2	4		abnorma	present	notprese	490	55	4			9.4	28			yes	yes	no	good	no	yes	ckd
11	63	70	1.01	3	0	abnorma	abnorma	present	notprese	380	60	2.7	131	4.2	10.8	32	4500	3.8	yes	yes	no	poor	yes	no	ckd
12	68	70	1.015	3	1		normal	present	notprese	208	72	2.1	138	5.8	9.7	28	12200	3.4	yes	yes	yes	poor	yes	no	ckd
13	68	70						notprese	notprese	98	86	4.6	135	3.4	9.8				yes	yes	yes	poor	yes	no	ckd
14	68	80	1.01	3	2	normal	abnorma	present	present	157	90	4.1	130	6.4	5.6	16	11000	2.6	yes	yes	yes	poor	yes	no	ckd
15	40	80	1.015	3	0		normal	notprese	notprese	76	162	9.6	141	4.9	7.6	24	3800	2.8	yes	no	no	good	no	yes	ckd
16	47	70	1.015	2	0		normal	notprese	notprese	99	46	2.2	138	4.1	12.6				no	no	no	good	no	no	ckd
17	47	80						notprese	notprese	114	87	5.2	139	3.7	12.1				yes	no	no	poor	no	no	ckd
18	60	100	1.025	0	3		normal	notprese	notprese	263	27	1.3	135	4.3	12.7	37	11400	4.3	yes	yes	yes	good	no	no	ckd
19	62	60	1.015	1	0		abnorma	present	notprese	100	31	1.6			10.3	30	5300	3.7	yes	no	yes	good	no	no	ckd
20	61	80	1.015	2	0	abnorma	abnorma	notprese	notprese	173	148	3.9	135	5.2	7.7	24	9200	3.2	yes	yes	yes	poor	yes	yes	ckd
21	60	90						notprese	notpresent		180	76	4.5		10.9	32	6200	3.6	yes	yes	yes	good	no	no	ckd
22	48	80	1.025	4	0	normal	abnorma	notprese	notprese	95	163	7.7	136	3.8	9.8	32	6900	3.4	yes	no	no	good	no	yes	ckd
23	21	70	1.01	0	0		normal	notprese	notpresent										no	no	no	poor	no	yes	ckd
24	42	100	1.015	4	0	normal	abnorma	notprese	present		50	1.4	129	4	11.1	39	8300	4.6	yes	no	no	poor	no	no	ckd
25	61	60	1.025	0	0		normal	notprese	notprese	108	75	1.9	141	5.2	9.9	29	8400	3.7	yes	yes	no	good	no	yes	ckd
26	75	80	1.015	0	0		normal	notprese	notprese	156	45	2.4	140	3.4	11.6	35	10300	4	yes	yes	no	poor	no	no	ckd
27	69	70	1.01	3	4	normal	abnorma	notprese	notprese	264	87	2.7	130	4	12.5	37	9600	4.1	yes	yes	yes	good	yes	no	ckd
28	75	70		1	3			notprese	notprese	123	31	1.4							no	yes	no	good	no	no	ckd
29	68	70	1.005	1	0	abnorma	abnorma	present	notpresent		28	1.4			12.9	38			no	no	yes	good	no	no	ckd
30		70						notprese	notprese	93	155	7.3	132	4.9					yes	yes	no	good	no	no	ckd

Figure 1.1 *- Screenshot of CSV file of the dataset.*

DATASET ANALYSIS

Numerical Features

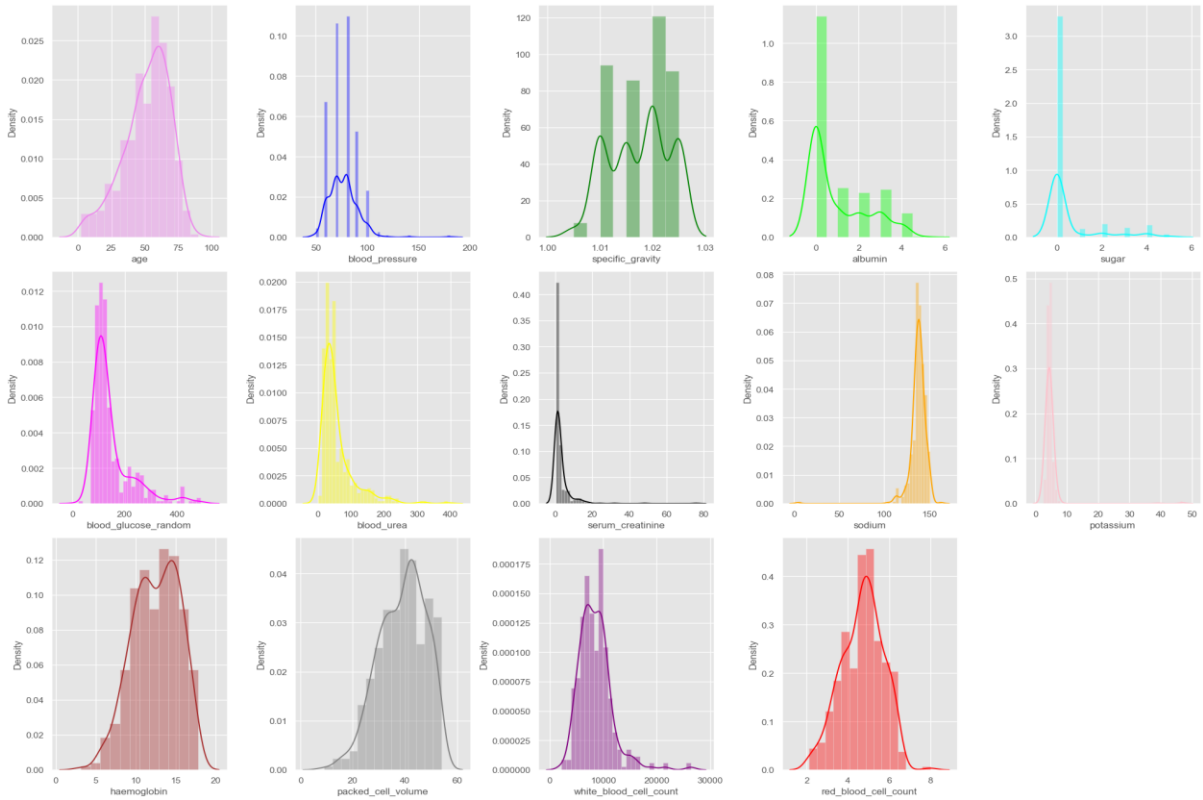


Fig. 2.1 *- This figure shows us the distribution* ***numerical*** *features of the dataset*

This visualisation shows us the distribution of numerical data within the dataset. Creating distribution plots (distplots) for each numerical column helps us analyse the data's spread, central tendency, and skewness.

Proceeding with **EDA** (Exploratory Data Analysis), we see the following insights from the numerical features of the data:

Age Distribution: The age of patients is widely spread, with a concentration of adults to elderly individuals, indicating the age range most susceptible to kidney issues.

Blood Pressure: Blood pressure values are primarily within the normal range, although variations suggest cases of hypertension, which is a common risk factor for kidney disease.

Specific Gravity: Specific gravity readings cluster around typical human levels, suggesting varying urine concentration, which can be crucial in diagnosing kidney function abnormalities.

Hemoglobin: Hemoglobin levels display significant variations, with signs of anemia present in a notable number of cases, aligning with common kidney disease symptoms.

Blood Glucose Random: The distribution reveals substantial variation in blood glucose levels, indicating the presence of diabetes or irregular blood sugar conditions among patients.

Serum Creatinine: Serum creatinine levels are skewed, with higher concentrations pointing to potential kidney impairment in affected individuals.

Albumin: The values are heavily skewed towards the lower end, suggesting that many patients have low albumin levels in urine, which is expected in healthy individuals but may indicate kidney issues when elevated.

Sugar: The distribution is also skewed, with most cases showing little to no sugar present in the urine. High values may suggest diabetes, which is a significant risk factor for kidney disease.

Blood Urea: The distribution shows a right skew, with higher values indicating impaired kidney function in some patients.

Sodium and Potassium: Sodium levels have a narrower distribution, concentrated around a normal range, while potassium shows a wide and skewed distribution. Both attributes are crucial for understanding electrolyte balance and kidney function.

Packed Cell Volume: The distribution of packed cell volume is relatively bell-shaped, but lower values may indicate anemia, a common symptom in chronic kidney disease patients.

White Blood Cell Count: The white blood cell count distribution is right-skewed, with outliers possibly indicating infections or inflammation.

Red Blood Cell Count: This distribution is slightly right-skewed, with lower values indicating potential anaemia

Categorical Features

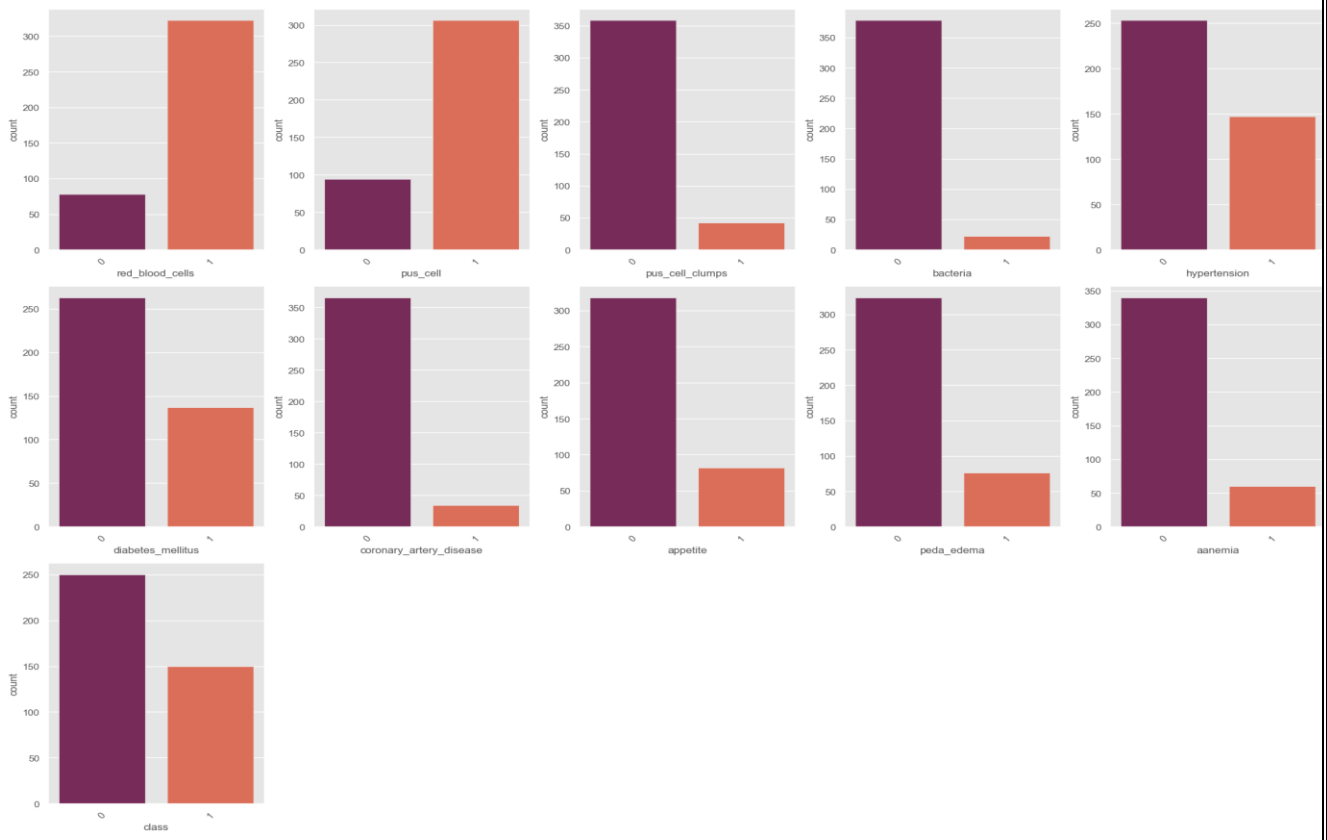


***Fig 2.2** - This figure shows us the distribution of **categorical** features of the dataset*

This visualisation shows us the distribution of categorical data within the dataset. It shows count plots for each categorical column showing the frequency of unique values in the columns. Which is useful for understanding the composition of the data and detecting potential issues like class imbalances, missing values, or irregularities in the categorical variables.

We get the following insights from the **categorical** data distribution of the dataset:

1. **Red Blood Cells**: There is a significant imbalance, with most cases showing normal red blood cell counts. The fewer abnormal cases might indicate kidney function deterioration.
2. **Pus Cell**: Most samples have average pus cell counts, but a noticeable number have abnormal counts, potentially indicating urinary tract infections or kidney issues.
3. **Pus Cell Clumps**: A vast majority show no pus cell clumps, with only a tiny proportion indicating the presence, which may point to infections.
4. **Bacteria**: Most cases do not have bacterial infections, with only a minority showing bacterial presence, suggesting that infection is not a common symptom in the dataset.
5. **Hypertension**: There is a relatively even distribution, with a significant number of patients having hypertension, a known risk factor for kidney disease.

6. **Diabetes Mellitus**: A considerable number of cases have diabetes, which is another significant risk factor for chronic kidney disease.
7. **Coronary Artery Disease**: Only a few patients have coronary artery disease, indicating it is less common in this dataset.
8. **Appetite**: Most patients report a good appetite, but there are notable cases of poor appetite, which can be a symptom of advanced kidney disease.
9. **Pedal Edema**: The majority of patients do not experience pedal oedema, but a significant number do, suggesting fluid retention issues in some cases.
10. **Anaemia**: Many patients are anaemic, consistent with kidney disease's common symptom of reduced red blood cell production.
11. **Class (Target Variable)**: There is a visible imbalance in the classes, with more cases indicating the presence of kidney disease (or vice versa).

Correlation Heatmap

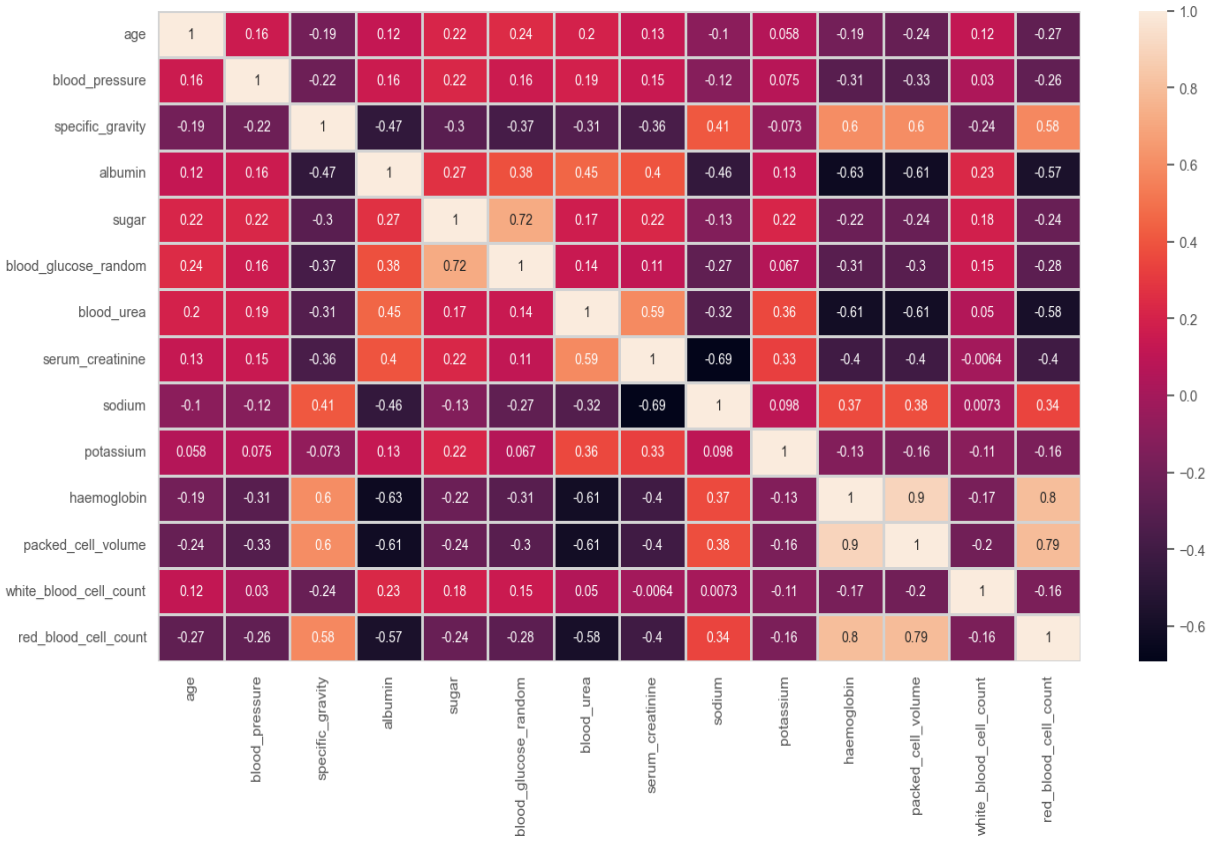


***Fig 2.3** - This figure shows us the distribution of **categorical** features of the dataset*

This is a heatmap to visualise the correlation between numerical columns in a dataset.

Why? - Correlation measures the strength and direction of a linear relationship between two variables, ranging from -1 (strong negative correlation) to 1 (strong positive correlation). This visualisation helps identify relationships between features, which is critical for feature selection and model building.

The following is an analysis of the key insights:

1. **Positive Correlations**:
    - **Packed Cell Volume and Hemoglobin (0.90)**: A very strong positive correlation between packed cell volume and hemoglobin indicates that as hemoglobin levels increase, the packed cell volume also tends to increase. This is biologically plausible, as both attributes are related to the oxygen-carrying capacity of the blood.
    - **Red Blood Cell Count and Hemoglobin (0.80)**: A strong positive correlation exists between red blood cell count and hemoglobin levels, as more red blood cells typically mean higher hemoglobin levels.

- **Red Blood Cell Count and Packed Cell Volume (0.79)**: This strong correlation suggests that a higher red blood cell count is associated with a higher packed cell volume.

2. **Negative Correlations**:
   - **Albumin and Hemoglobin (-0.63)**: There is a notable negative correlation between albumin in urine and hemoglobin levels, suggesting that higher protein levels in urine are associated with lower hemoglobin levels, which is often seen in kidney disease cases.
   - **Blood Pressure and Hemoglobin (-0.31)**: Blood pressure has a moderate negative correlation with hemoglobin levels, indicating that higher blood pressure may be linked to lower hemoglobin, which is a potential indicator of chronic kidney disease.
3. **Other Notable Relationships**:
   - **Serum Creatinine and Blood Urea (0.59)**: There is a moderate positive correlation between serum creatinine and blood urea, both of which are indicators of kidney function. Higher levels of these markers often point to impaired kidney function.
   - **Specific Gravity and Packed Cell Volume (0.60)**: A positive correlation between urine specific gravity and packed cell volume suggests that urine concentration may be related to the blood's capacity to carry oxygen.
4. **Weak Correlations**:
   - Attributes like **sodium and potassium** show weak or negligible correlations with most other features, which means they are less related to the overall trends captured by the heatmap.

DATA PREPROCESSING

We preprocess our dataset by handling **missing** values, **encoding** categorical features, and then finally **preparing** the dataset for machine learning. Our approach ensures the data is clean, consistent, and ready for use in predictive modeling.

Our data preprocessing pipeline includes the following (we have described our Python code here):

1. Checking for Missing Values

The process begins by identifying missing values in the dataset:

- df.isnull().sum().sort_values(ascending=False): Lists the features in descending order of missing values to prioritise imputation.

- df[num_cols].isnull().sum(): Counts missing values for numerical columns.

- **df[cat_cols].isnull().sum()**: Counts missing values for categorical columns.

2. Imputation of Missing Values

Two different approaches are used to fill missing values, depending on the nature of the data:

a. Random Sampling for Numerical Features

- Random sampling is suitable for columns with many missing values, as it maintains the distribution of the feature.

- **random_sampling()** Function:

1. Extracts non-null values from the feature using df[feature].dropna().

2. Randomly samples the required number of values to match the count of missing values (df[feature].isna().sum()).

3. Replaces missing values with the sampled data using .loc[].

- Applied iteratively to numerical features and specified columns (red_blood_cells and pus_cell) for better accuracy.

b. Mode Imputation for Categorical Features

- Missing categorical values are replaced with the **mode** (most frequent value).

- **impute_mode()** Function:

1. Computes the mode of the feature with df[feature].mode()[0].

2. Fills missing values with this mode using .fillna().

- Applied to all categorical columns using a loop.

We are using a two-fold approach to minimise bias and ensure data consistency.

3. Categorical Data Transformation

Categorical data is encoded into numerical values using **Label Encoding**:

- LabelEncoder:

- Converts unique categories in each column into integer values (e.g., "yes" becomes 1, "no" becomes 0).

- Applied to all categorical columns (cat_cols) in a loop, making the data model-ready.

4. Feature Preparation

Once missing values are imputed and categorical features are encoded, the dataset is split into:

- X (Independent Variables): All features except the target column (class).

- y (Target Variable): The class column, which indicates whether a patient has kidney disease.

This separation ensures that the model is trained only on the predictors (X) while learning to predict the target (y).

MODEL EVALUATION

1. Logistic Regression

The Logistic Regression model is a widely used supervised machine learning algorithm for binary classification tasks. Unlike Linear Regression, which predicts continuous values, Logistic Regression predicts the probability of an instance belonging to a particular class, typically outputting values between 0 and 1. A threshold (e.g., 0.5) is applied to classify instances into two categories

```
Training Accuracy of Logistic Regression is 0.909375
Testing Accuracy of Logistic Regression is 0.9
Confusion Matrix of Logistic Regression is
[[42  5]
 [ 3 30]]

Classification Report of Logistic Regression is
              precision    recall  f1-score   support

           0       0.93      0.89      0.91        47
           1       0.86      0.91      0.88        33

    accuracy                           0.90        80
   macro avg       0.90      0.90      0.90        80
weighted avg       0.90      0.90      0.90        80
```

Logistic Regression Model Analysis -

Training Accuracy: 0.909375 (90.93%)

Testing Accuracy: 0.9 (90%)

The model demonstrates high training and testing accuracy, indicating that it is able to generalise well on unseen data. The minimal difference between training and testing accuracy suggests that our model is not overfitting.

Confusion Matrix Analysis:

True Positives (TP): 30 (Correctly predicted cases of kidney disease)

True Negatives (TN): 42 (Correctly predicted cases of no kidney disease)

False Positives (FP): 5 (Cases predicted as kidney disease but are actually healthy)

False Negatives (FN): 3 (No cases of kidney disease were missed)

The model has a perfect recall for class '1' (kidney disease), meaning it correctly identifies all patients with kidney disease, but it has a small number of false positives for class '0'.

2. K-Nearest Neighbors (KNN) algorithm

The K-Nearest Neighbors (KNN) algorithm is a simple, non-parametric supervised machine learning algorithm used for classification (and regression) tasks. The core idea behind KNN is to classify a data point based on how its neighbors are classified. It does this by identifying the "K" nearest points in the feature space and assigning the most common class among them to the data point.

```
Training Accuracy of KNN is 0.80625
Testing Accuracy of KNN is 0.625
Confusion Matrix of KNN is
 [[31 16]
 [14 19]]

Classification Report of KNN is
              precision    recall  f1-score   support

           0       0.69      0.66      0.67        47
           1       0.54      0.58      0.56        33

    accuracy                           0.62        80
   macro avg       0.62      0.62      0.62        80
weighted avg       0.63      0.62      0.63        80
```

KNN Model Analysis -

Training Accuracy: 0.80625 (80.63%)

Testing Accuracy: 0.625 (62.5%)

The KNN model shows a significant drop in performance when comparing the training set to the testing set which indicates overfitting. The testing accuracy of 62.5% is relatively low, which suggests that the model struggles to generalise to unseen data.

Confusion Matrix Analysis -

True Positives (TP): 19 (Correctly predicted cases of kidney disease)

True Negatives (TN): 31 (Correctly predicted cases of no kidney disease)

False Positives (FP): 16 (Healthy patients incorrectly predicted as having kidney disease)

False Negatives (FN): 14 (Kidney disease cases that were missed)

The confusion matrix indicates that the model has a considerable number of both false positives and false negatives, making it unreliable.

3. Naive Bayes Model

The Naive Bayes model is a probabilistic classifier based on applying Bayes' Theorem with the "naive" assumption of feature independence. This means that Naive Bayes assumes that the presence (or absence) of a feature in a class is independent of the presence (or absence) of any other feature. This is a simplifying assumption and may not always hold true.

```
Training Accuracy of Naive Bayes is 0.940625
Testing Accuracy of Naive Bayes is 0.925
Confusion Matrix of Naive Bayes is
[[41  6]
 [ 0 33]]

Classification Report of Naive Bayes is
              precision    recall  f1-score   support

           0       1.00      0.87      0.93        47
           1       0.85      1.00      0.92        33

    accuracy                           0.93        80
   macro avg       0.92      0.94      0.92        80
weighted avg       0.94      0.93      0.93        80
```

Naive Bayes Model Analysis -

- **Training Accuracy:** 0.940625 (94.06%)
- **Testing Accuracy:** 0.925 (92.5%)
- The Naive Bayes model demonstrates strong performance with both high training and testing accuracy, indicating that the model generalises well and does not appear to overfit. The small gap between the training and testing accuracy reflects good model robustness.

Confusion Matrix Analysis -

True Positives (TP): 33 (Correctly predicted cases of kidney disease)

True Negatives (TN): 41 (Correctly predicted cases of no kidney disease)

False Positives (FP): 6 (Healthy patients incorrectly predicted as having kidney disease)

False Negatives (FN): 0 (No cases of kidney disease were missed)

The model has a perfect recall for class 1 (kidney disease), meaning it correctly identifies all kidney disease cases. However, it has a few false positives (6 cases)

4. The Decision Tree Classifier (DTC)

The Decision Tree Classifier (DTC) is a supervised machine learning algorithm that can be used for both classification and regression tasks.

It works by recursively partitioning the data into subsets based on the value of the features, and this results in a tree-like structure of decisions. In classification tasks, the decision tree splits the data into different classes by making decisions based on the input features, ultimately classifying the data points into one of the predefined categories.

```
Training Accuracy of DTC is 1.0
Testing Accuracy of DTC is 0.95
Confusion Matrix of DTC is
 [[44  3]
 [ 1 32]]

Classification Report of DTC is
              precision    recall  f1-score   support

           0       0.98      0.94      0.96        47
           1       0.91      0.97      0.94        33

    accuracy                           0.95        80
   macro avg       0.95      0.95      0.95        80
weighted avg       0.95      0.95      0.95        80
```

DTC Model Analysis -

Overall Performance:

- **Training Accuracy:** 1.0 (100%)
- **Testing Accuracy:** 0.95 (95%)
- The Decision Tree Classifier (DTC) achieves perfect accuracy on the training set which indicates that the model has learned the training data perfectly. However, this may point to **overfitting**, as the model is likely memorising rather than generalising. Despite this, the testing accuracy of 95% suggests that the model generalises quite well on unseen data in this case

Confusion Matrix Analysis

True Positives (TP): 32 (Correctly predicted cases of kidney disease)

True Negatives (TN): 44 (Correctly predicted cases of no kidney disease)

False Positives (FP): 3 (Healthy patients incorrectly predicted as having kidney disease)

False Negatives (FN): 1 (Kidney disease case that was missed)

The model has very few false positives and false negatives, which is a desirable outcome.

5. The Random Forest Classifier (RFC)

The Random Forest Classifier (RFC) is an ensemble machine learning algorithm that combines multiple decision trees to improve classification accuracy and reduce the risk of overfitting. Unlike a single decision tree, which may be overly sensitive to noise in the data, a random forest constructs several trees during training and outputs the mode (most frequent) class of the individual trees for classification tasks.

```
Training Accuracy of Random Forest is 0.996875
Testing Accuracy of Random Forest is 0.9875
Confusion Matrix of Random Forest is
 [[47  0]
 [ 1 32]]

Classification Report of Random Forest is
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        47
           1       1.00      0.97      0.98        33

    accuracy                           0.99        80
   macro avg       0.99      0.98      0.99        80
weighted avg       0.99      0.99      0.99        80
```

RFC Model Analysis -

Training Accuracy: 0.996875 (99.69%)

Testing Accuracy: 0.9875 (98.75%)

The Random Forest Classifier (RFC) demonstrates exceptional performance with near-perfect training and testing accuracy. The minimal difference between the two accuracies indicates that the model generalises very well, with minimal overfitting.

Confusion Matrix Analysis -

True Positives (TP): 32 (Correctly predicted cases of kidney disease)

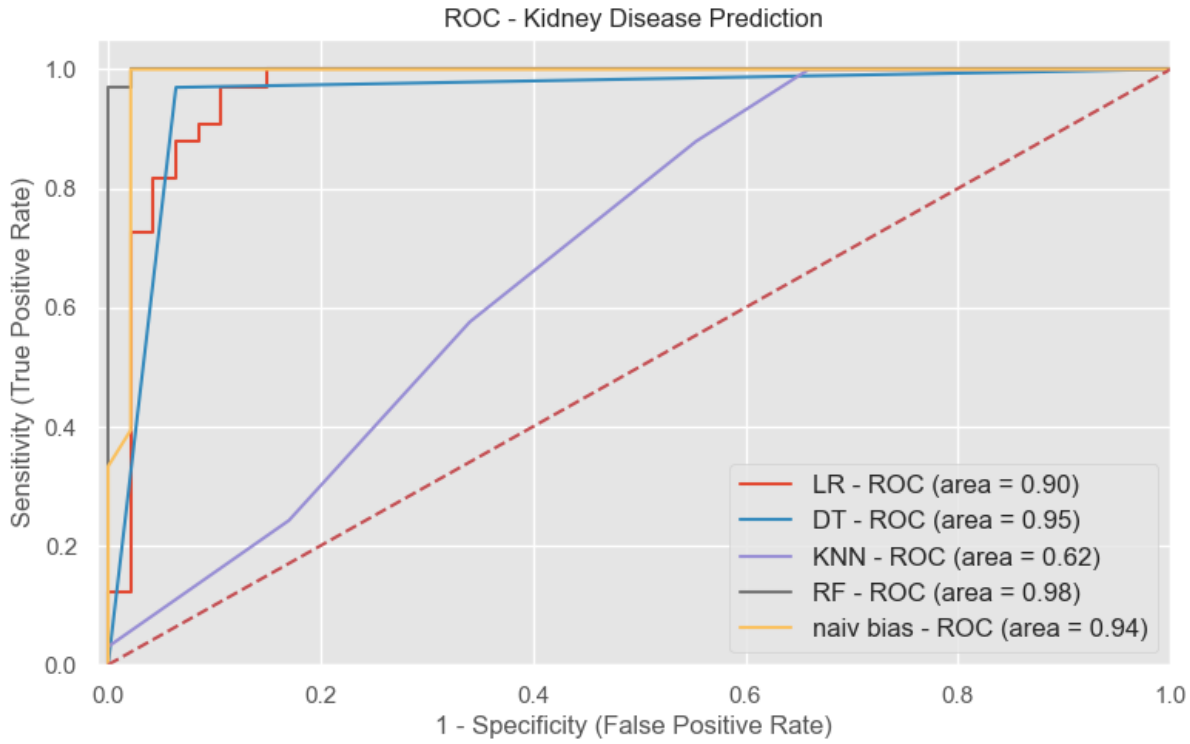
True Negatives (TN): 47 (Correctly predicted cases of no kidney disease)

False Positives (FP): 0 (No healthy patients incorrectly predicted as having kidney disease)

False Negatives (FN): 1 (One case of kidney disease missed)

The model has perfect performance for classifying healthy individuals (no false positives) and only one missed case of kidney disease, showcasing its high reliability.

MODEL COMPARISON & RESULTS



ROC Curve Analysis:

The Receiver Operating Characteristic (ROC) curve visually represents the performance of different classification models by plotting the **True Positive Rate (Sensitivity)** against the **False Positive Rate (1 - Specificity)**. The Area Under the Curve (AUC) provides a single metric to summarise each of the model's ability to distinguish between classes.

A higher AUC indicates better model performance.

Model Comparison:

- **Random Forest (RF)**
  - **AUC:** 0.98
  - The Random Forest model achieves the highest AUC, indicating it can best differentiate between patients with and without kidney disease. The curve is close to the top left corner, showing high sensitivity and low false positive rate.
- **Decision Tree (DT)**
  - **AUC:** 0.95
  - The Decision Tree model also performs well, with a high AUC of 0.95. Its ROC curve is quite close to that of the Random Forest model, demonstrating excellent performance but slightly lower compared to RF. However, as mentioned before, the model may show overfitting.
- **Naive Bayes**
  - **AUC:** 0.94

    - The Naive Bayes model has a strong AUC, suggesting good classification performance. However, it is slightly less effective compared to the Decision Tree and Random Forest models.
- **Logistic Regression (LR)**
    - **AUC:** 0.90
    - The Logistic Regression model performs well, with an AUC of 0.90. Although it is a solid performer, it lags behind the Decision Tree and Random Forest models.
- **K-Nearest Neighbors (KNN)**
    - **AUC:** 0.62
    - The KNN model has the lowest AUC of 0.62, and shows poor performance in distinguishing between classes. The curve is much closer to the diagonal line, representing a model that is only slightly better than random guessing.

Key Points:

- The **Random Forest** model stands out as the top performer, with an AUC of 0.98, making it highly reliable for kidney disease prediction.
- The **Decision Tree** and **Naive Bayes** models also exhibit strong performance, with AUCs of 0.95 and 0.94, respectively.
- **Logistic Regression** provides a decent performance with an AUC of 0.90, making it a viable option but not as strong as the ensemble methods.
- The **K-Nearest Neighbors** model performs poorly with a low AUC, confirming that it is unsuitable for this problem without significant tuning or modification.

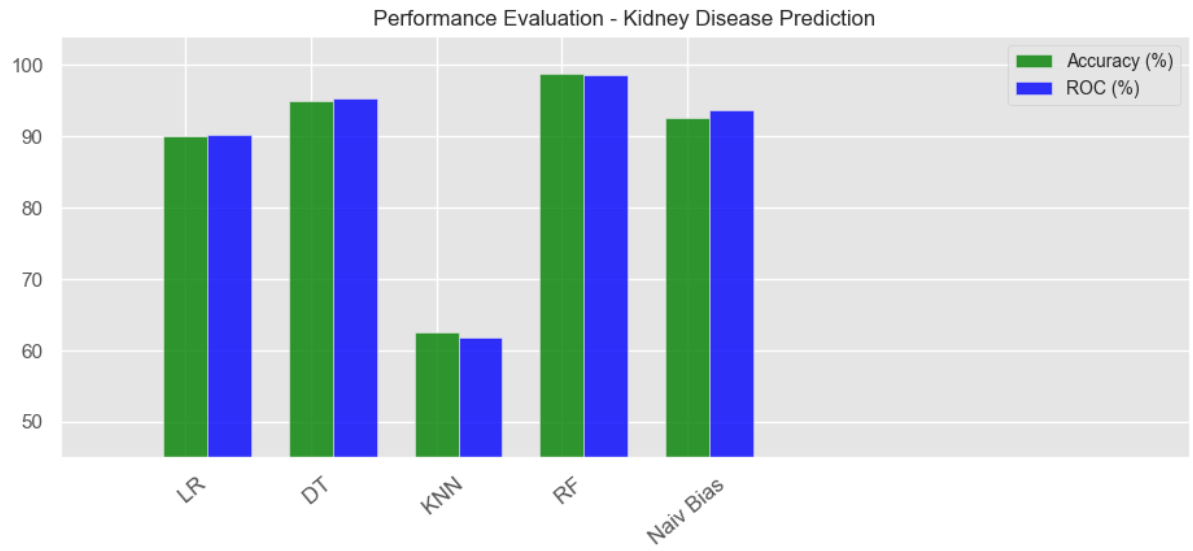


Fig - *The bar chart compares the* ***Accuracy*** *and* ***ROC AUC (%)*** *of different models used for kidney disease prediction.*

Results -

On comparing our ML models for kidney disease prediction reveals distinct strengths and weaknesses. **Logistic Regression** and the **Random Forest Classifier** emerged as top performers, with 90% and 98.75% testing accuracies, respectively. Logistic Regression achieved a really good precision for healthy cases and almost perfect recall for kidney disease cases, making it highly effective for early diagnosis. The **Random Forest Classifier** excelled with near-perfect scores across all metrics. In contrast, the **K-Nearest Neighbors** model underperformed with only 62.5% accuracy, suffering from significant overfitting and poor generalisation. The **Naive Bayes** model performed well with high recall, which is important for minimising missed cases, but it showed a moderate precision for kidney disease cases. Finally, the **Decision Tree Classifier** algorithm demonstrated strong performance but indicated some risk of overfitting due to perfect training accuracy which we have described earlier as well. Overall, the Random Forest Classifier stands out as the most accurate and reliable model, suitable for a 'real-world' application and we can even apply it in medical scenarios owing to its performance, we can further improve our models by optimising them and addressing their limitations as we have described in Model Evaluation.

We have also created a **web frontend** so that one can easily predict Chronic Kidney Disease using our best-performing model, **RFC (Random Forest Classifier).**