Predictive Modeling for Chronic Kidney Disease (CKD): Development and Performance Analysis

DA5030

Yogesh Purushotham

Fall 2023

Contents

Objective:

The primary goal of my analysis is to predict whether an individual has chronic kidney disease, making the target variable 'classification', which categorizes entries into 'ckd' or not 'ckd'. This makes it a classification task, as the target variable is categorical.

Significance:

Early detection and intervention of chronic kidney disease (CKD) can significantly improve patient outcomes and give more time for early diagnosis.

Algorithms and approach:

In this project, I plan to employ logistic regression, decision trees, and SVM models followed by Random forest model as an ensemble for decision tree and also an ensemble model using the predictions of logistic regression, decision trees, and SVM models. For feature engineering, given the mix of numeric and categorical data, I anticipate encoding categorical variables and normalizing numerical features.

Model evaluation:

To evaluate the fit of these algorithms, I will use metrics such as accuracy, precision, recall, and F1 score. These metrics are particularly relevant for classification tasks and will help in assessing the performance of each model.

While similar analyses have been conducted on datasets related to kidney disease, my approach will focus on a comprehensive exploration of this specific dataset, potentially uncovering new insights. I aim to integrate advanced machine learning techniques and a thorough exploratory analysis to understand the complexities of kidney disease prediction. Here I employ Logistic regression model which is a standard choice for binary classification tasks and can provide a baseline for performance and Decision tree mdoel which is more complex and can handle non-linear relationships better, followed by SVM or a support vector machine. They are particularly effective for classification tasks and can also provide insights into feature importance. My project will differ in its detailed focus on the in-depth evaluation of multiple machine learning models, aimed at deriving the most accurate predictions possible.

Data Acquisition

Overview of data:

For my signature term project in machine learning and data mining, I have selected a dataset focused on kidney disease, specifically chronic kidney disease (CKD), collected from UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease. (UCI Machine Learning Repository, n.d.) This dataset is used to predict the chronic kidney disease which is collected from hospitals for 2 months of period and Comprises 400 rows and 26 columns (variables/features). These attributes include age, blood pressure, specific gravity, albumin level, sugar level, red blood cells count, and more, offering a comprehensive view of factors potentially influencing kidney health.

```
## 'data.frame':    400 obs. of  26 variables:
##  $ id            : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ age           : num  48 7 62 48 51 60 68 24 52 53 ...
##  $ bp            : num  80 50 80 70 80 90 70 NA 100 90 ...
##  $ sg            : num  1.02 1.02 1.01 1 1.01 ...
##  $ al            : num  1 4 2 4 2 3 0 2 3 2 ...
##  $ su            : num  0 0 3 0 0 0 0 4 0 0 ...
##  $ rbc           : chr  NA NA "normal" "normal" ...
##  $ pc            : chr  "normal" "normal" "normal" "abnormal" ...
##  $ pcc           : chr  "notpresent" "notpresent" "notpresent" "present" ...
##  $ ba            : chr  "notpresent" "notpresent" "notpresent" "notpresent" ...
##  $ bgr           : num  121 NA 423 117 106 74 100 410 138 70 ...
##  $ bu            : num  36 18 53 56 26 25 54 31 60 107 ...
##  $ sc            : num  1.2 0.8 1.8 3.8 1.4 1.1 24 1.1 1.9 7.2 ...
##  $ sod           : num  NA NA NA 111 NA 142 104 NA NA 114 ...
##  $ pot           : num  NA NA NA 2.5 NA 3.2 4 NA NA 3.7 ...
##  $ hemo          : num  15.4 11.3 9.6 11.2 11.6 12.2 12.4 12.4 10.8 9.5 ...
##  $ pcv           : chr  "44" "38" "31" "32" ...
##  $ wc            : chr  "7800" "6000" "7500" "6700" ...
##  $ rc            : chr  "5.2" NA NA "3.9" ...
##  $ htn           : chr  "yes" "no" "no" "yes" ...
```

```
##  $ dm            : chr  "yes" "no" "yes" "no" ...
##  $ cad           : chr  "no" "no" "no" "no" ...
##  $ appet         : chr  "good" "good" "poor" "poor" ...
##  $ pe            : chr  "no" "no" "no" "yes" ...
##  $ ane           : chr  "no" "no" "yes" "yes" ...
##  $ classification: chr  "ckd" "ckd" "ckd" "ckd" ...
```

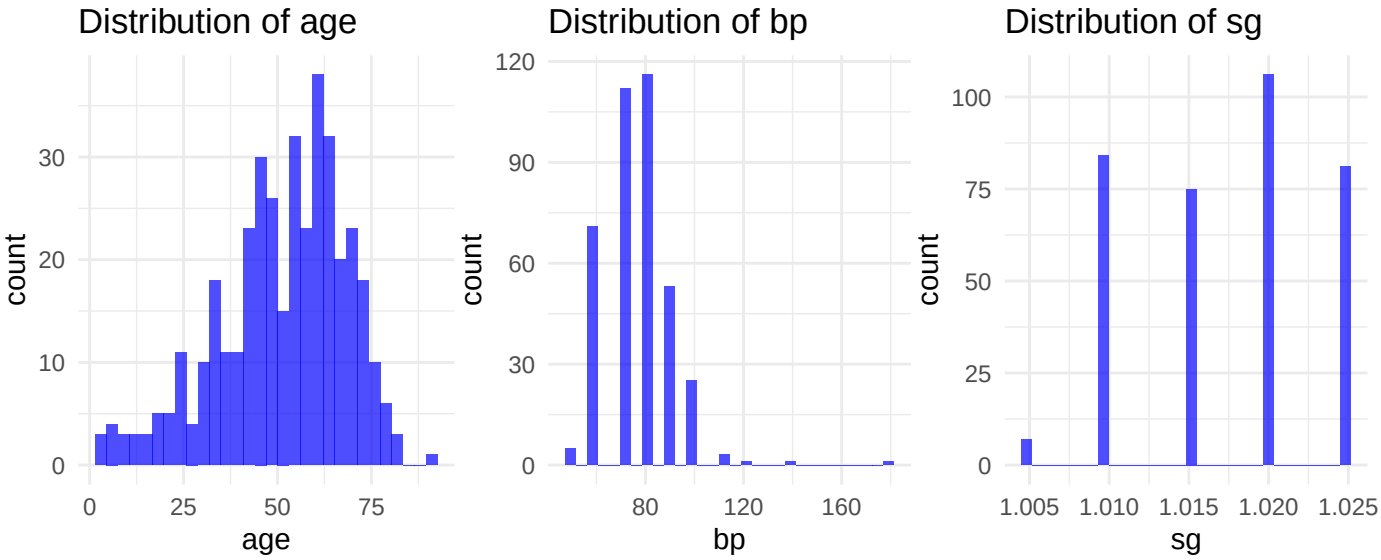
The initial exploration of the dataset reveals the following:

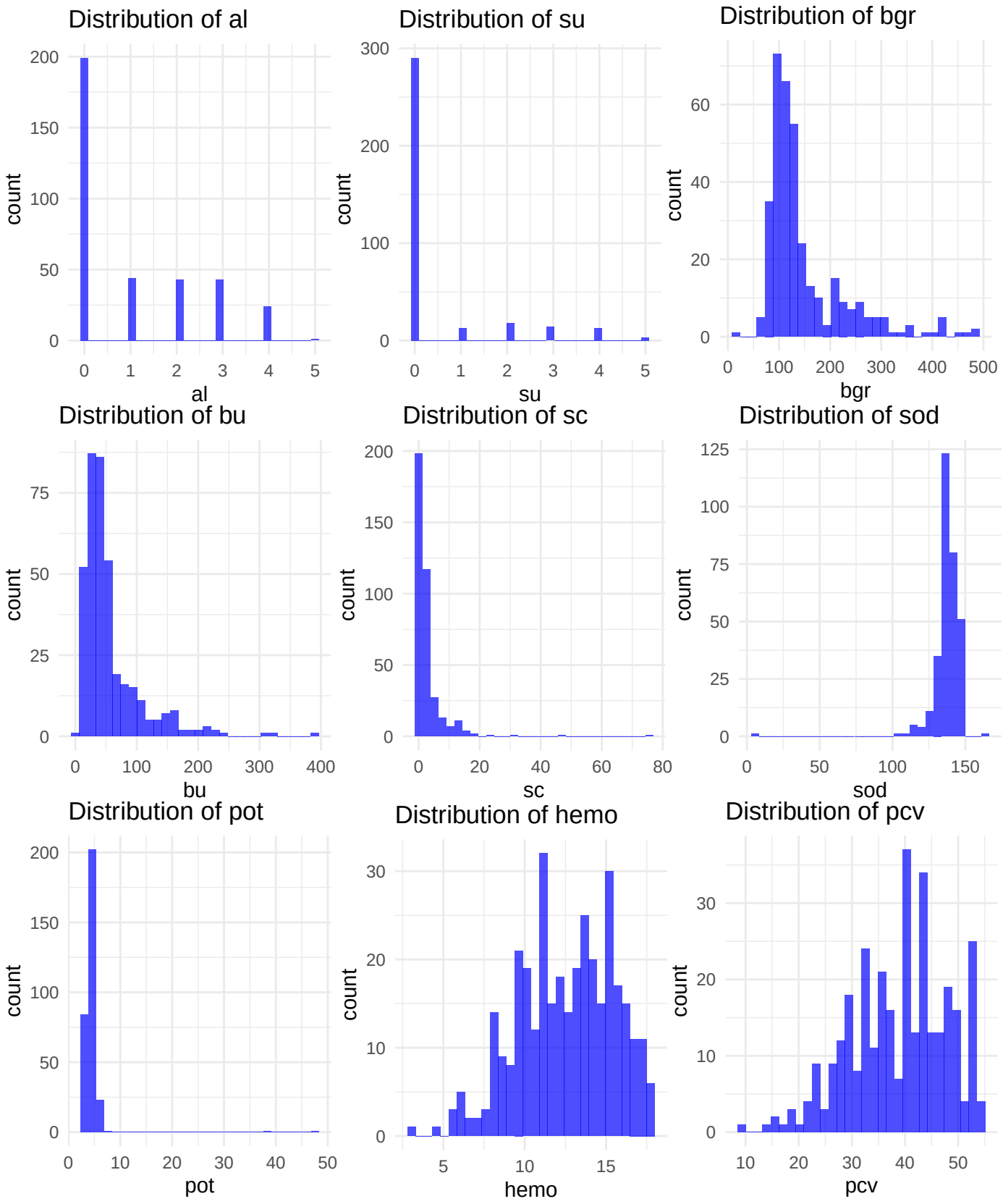
- The dataset contains 400 entries and 26 columns.
- There are both numerical (e.g., age, blood pressure) and categorical variables (e.g., red blood cell count, pus cell clumps).
- The id column is present, which I will drop for modeling purposes.
- The classification column is the target variable, with classes like 'ckd' (chronic kidney disease) and not 'ckd'.
- Several columns have missing values, evident from the non-null count being less than 400 in many columns.
- Change 'ckd' to 'ckd' in the classification column to ensure consistency in the target variable.

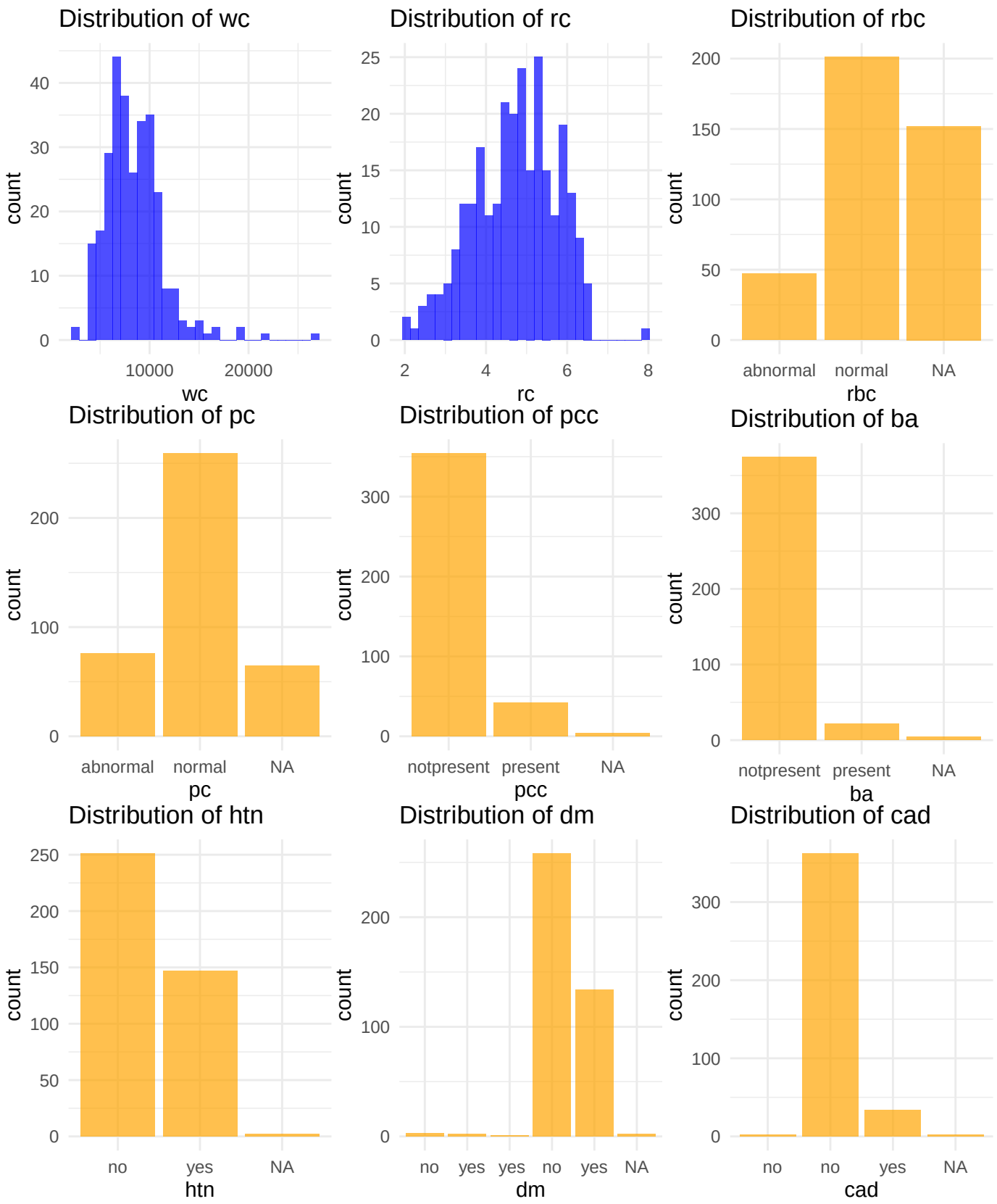
Data Exploration and data distribution:

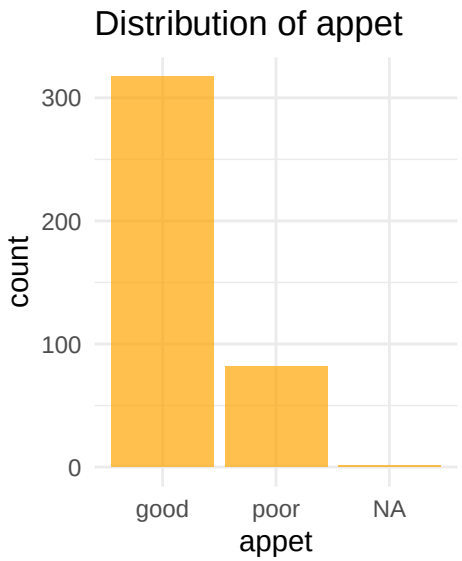
Exploratory Data Plots: The histograms provide insights into the distributions of various continuous variables such as age, blood pressure (bp), specific gravity (sg), albumin (al), and sugar (su).

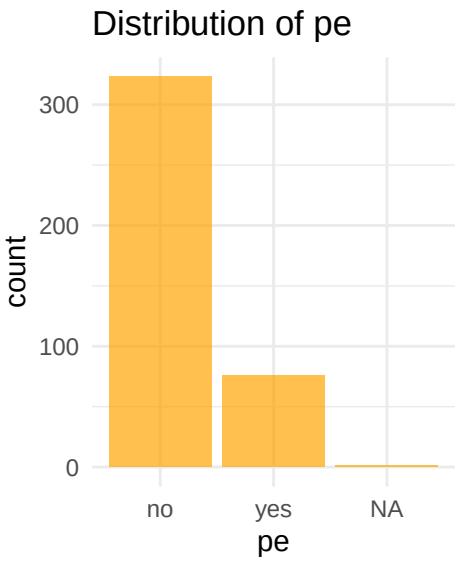
Detection of Outliers: The boxplots and scatter plots for numerical columns help in identifying potential outliers. Outliers are the data points that significantly deviate from the rest of the data and can impact the analysis.

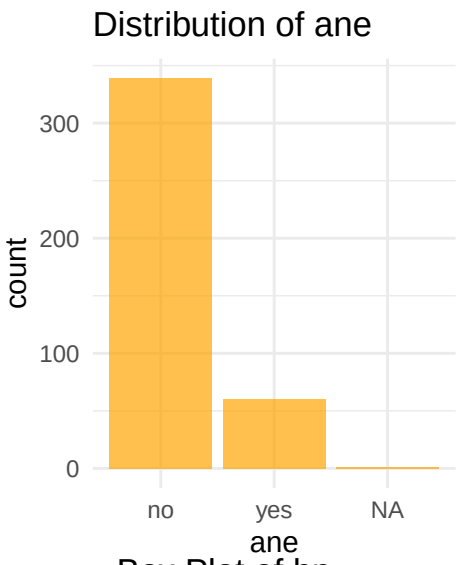


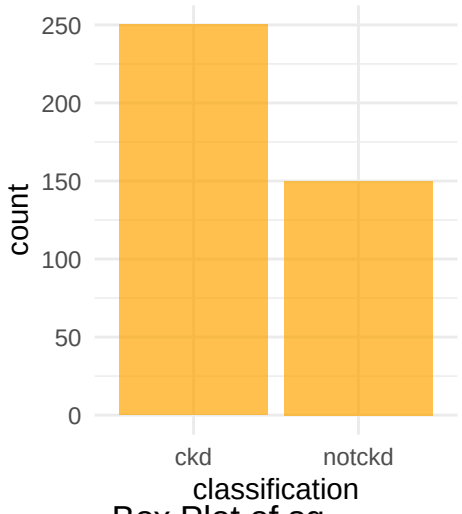

Distribution of al
count
al
Distribution of su
count
su
Distribution of bgr
count
bgr
Distribution of bu
count
bu
Distribution of sc
count
sc
Distribution of sod
count
sod
Distribution of pot
count
pot
Distribution of hemo
count
hemo
Distribution of pcv
count
pcv

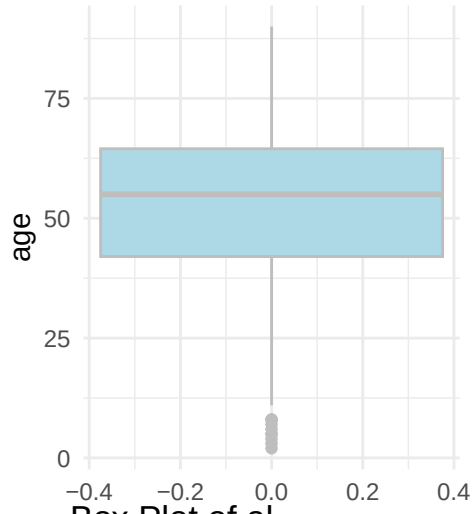

Distribution of wc
count
wc
Distribution of rc
count
rc
Distribution of rbc
count
abnormal
normal
NA
rbc
Distribution of pc
count
abnormal
normal
NA
pc
Distribution of pcc
count
notpresent
present
NA
pcc
Distribution of ba
count
notpresent
present
NA
ba
Distribution of htn
count
no
yes
NA
htn
Distribution of dm
count
no
yes
yes
no
yes
NA
dm
Distribution of cad
count
no
no
yes
NA
cad

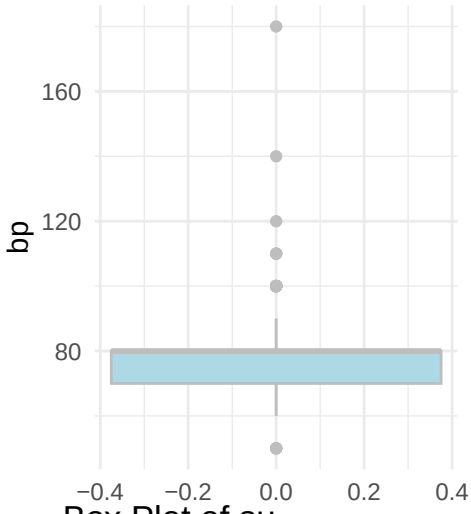

Distribution of appet
count
300
200
100
0
good
poor
NA
appet

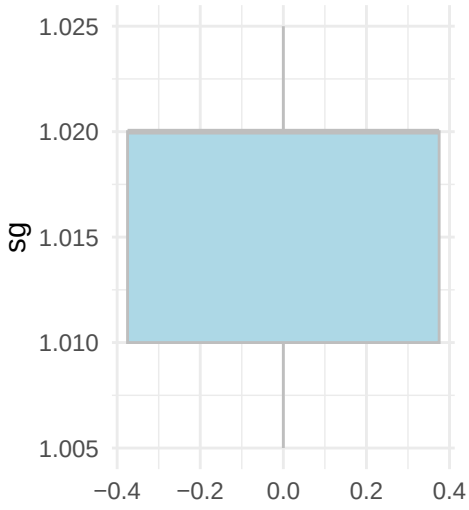

Distribution of pe
count
300
200
100
0
no
yes
NA
pe

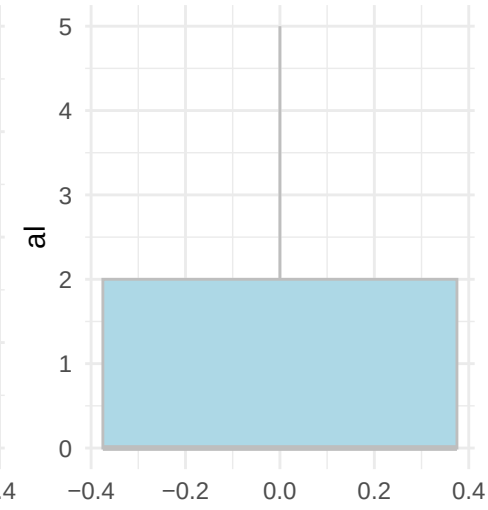

Distribution of ane
count
300
200
100
0
no
yes
NA
ane

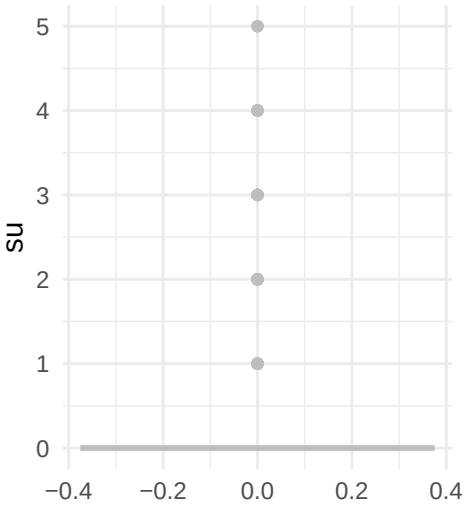

Distribution of classificati
count
250
200
150
100
50
0
ckd
notckd
classification

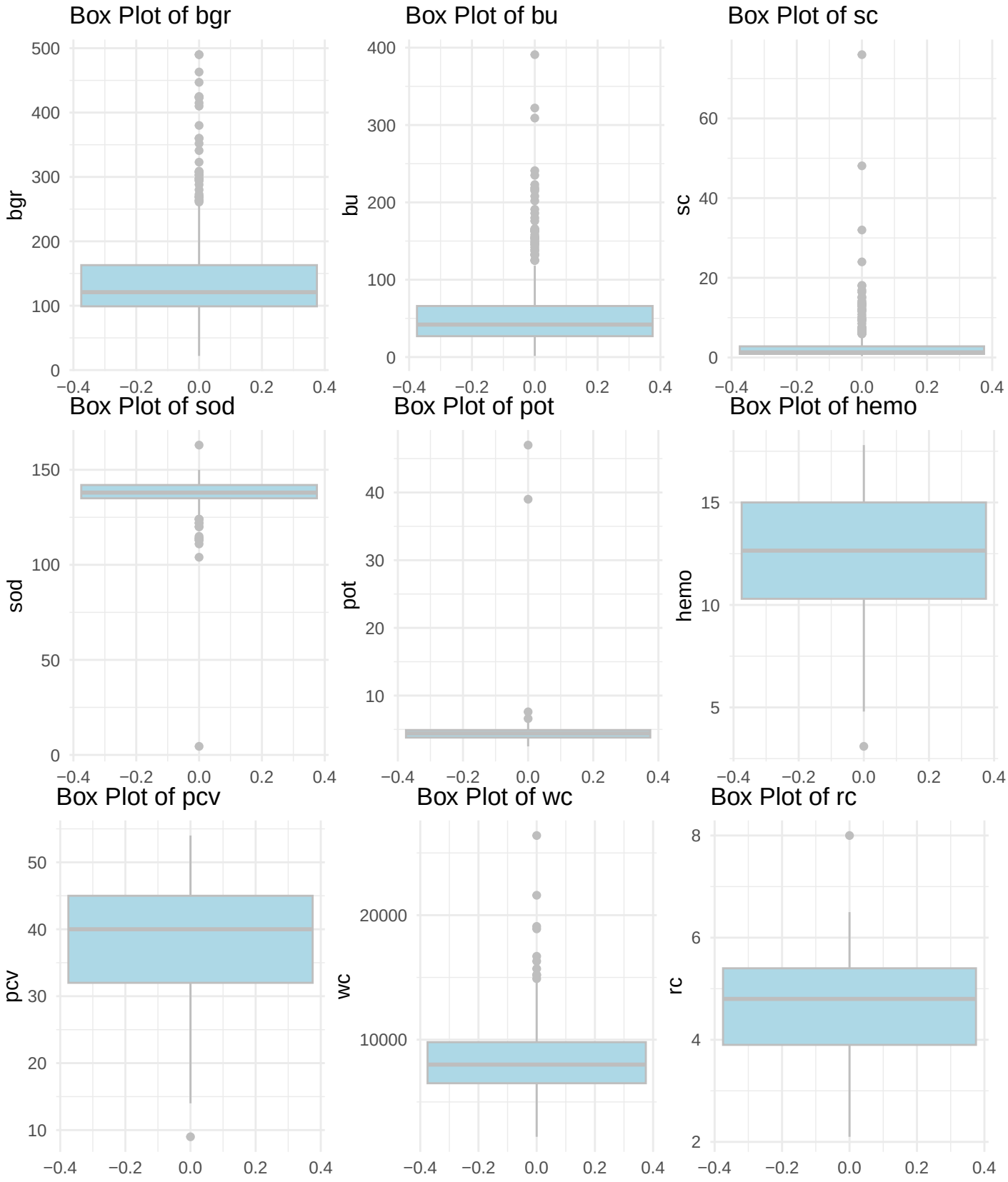

Box Plot of age
age
75
50
25
0
−0.4
−0.2
0.0
0.2
0.4

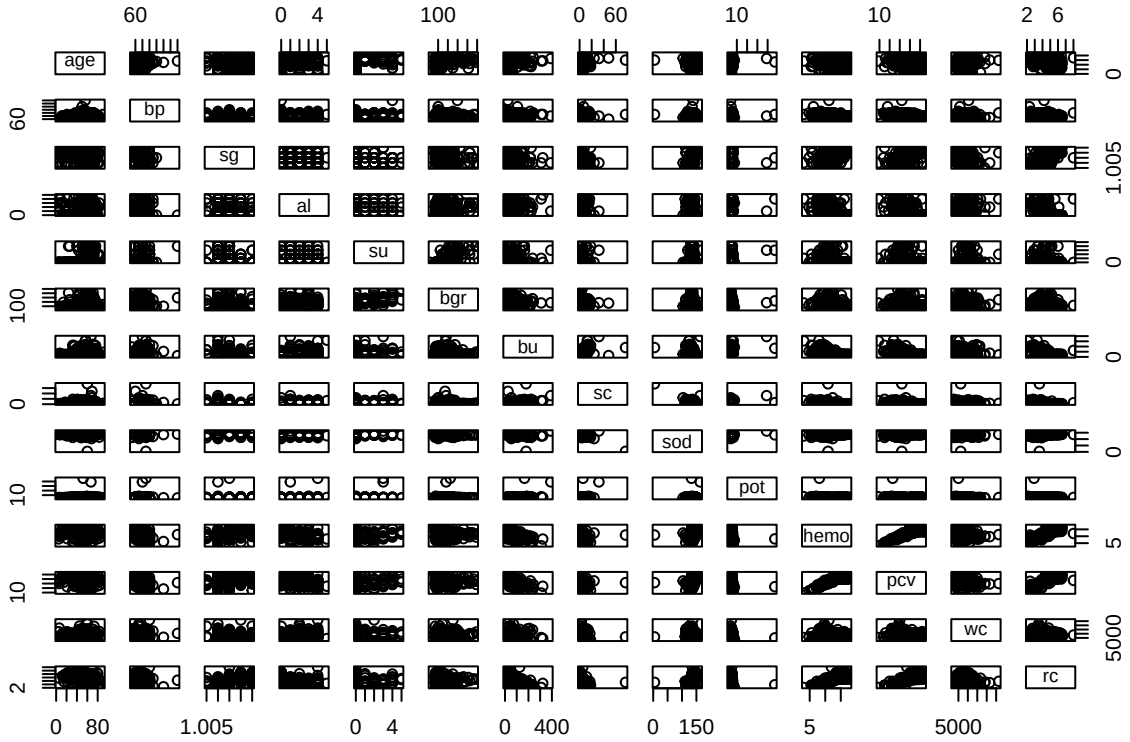

Box Plot of bp
bp
160
120
80
−0.4
−0.2
0.0
0.2
0.4


Box Plot of sg
sg
1.025
1.020
1.015
1.010
1.005
−0.4
−0.2
0.0
0.2
0.4


Box Plot of al
al
5
4
3
2
1
0
−0.4
−0.2
0.0
0.2
0.4


Box Plot of su
su
5
4
3
2
1
0
−0.4
−0.2
0.0
0.2
0.4


Box Plot of bgr
bgr
Box Plot of bu
bu
Box Plot of sc
sc
Box Plot of sod
sod
Box Plot of pot
pot
Box Plot of hemo
hemo
Box Plot of pcv
pcv
Box Plot of wc
wc
Box Plot of rc
rc



Exploratory Data Analysis Results and Evaluation of Data Distribution:

Numerical Data Distribution:

Blood Pressure (bp), Blood Urea (bu), Serum Creatinine (sc): These features exhibit a right-skewed distribution, indicating a concentration of lower values and a long tail towards higher values.

Blood Glucose Random (bgr), Sodium (sod), Potassium (pot): Similar to bp, bu, and sc, these features also show a right-skewed distribution.

Hemoglobin (hemo), Packed Cell Volume (pcv), Red Blood Cell Count (rc): These features appear more normally distributed but still show some skewness.

Age: The distribution of age is relatively more uniform but slightly right-skewed.

Overall, many numerical features exhibit skewness, which may require normalization or transformation. Categorical Data Distribution:

Red Blood Cells (rbc), Pus Cell (pc), Pus Cell Clumps (pcc), Bacteria (ba): These features show a significant imbalance in their categories.

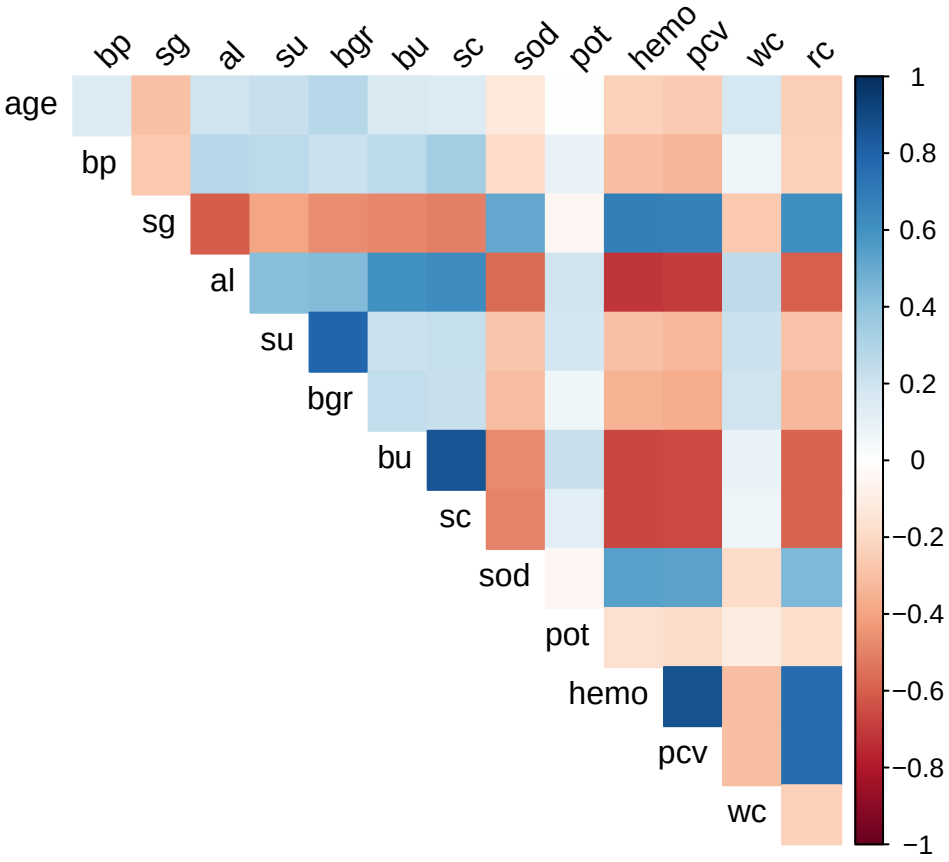
Hypertension (htn), Diabetes Mellitus (dm), Coronary Artery Disease (cad), Appetite (appet), Pedal Edema (pe), Anemia (ane) also exhibit imbalance, with one category being more prevalent than the other.

Here are the observations from the box plots for features in the dataset:

- Blood Pressure (bp): Some values are significantly higher than the majority, indicating potential outliers.
- Blood Glucose Random (bgr): This feature also shows a number of outliers on the higher side.
- Blood Urea (bu): There are outliers present, especially higher values.
- Serum Creatinine (sc): This feature has several high-value outliers.
- Sodium (sod): There are outliers on both the lower and higher ends.
- Potassium (pot): Numerous high-value outliers are present.
- Hemoglobin (hemo): A few low-value outliers can be seen.

- Packed Cell Volume (pcv), White Blood Cell Count (wc), and Red Blood Cell Count (rc): These features also display outliers, mostly on the higher side for wc and on both sides for pcv and rc.

Correlation/Collinearity Analysis using heatmap



The heatmap above shows the correlation matrix for the numerical features in the dataset:

Strong Correlations: There are pairs of features that exhibit strong positive or negative correlations. For example, features like 'hemo' (hemoglobin) and 'pcv' (packed cell volume) show a strong positive correlation.

Weak Correlations: Some features display weak correlations with others, indicating less direct linear relationships.

Data Cleaning & Shaping:

identification of missing values

I observed that there are quite a few missing values in the dataset so to count missing values for each column:

```
##             age              bp              sg              al              su
##               9              12              47              46              49
##             rbc              pc             pcc              ba             bgr
##             152              65               4               4              44
##              bu              sc             sod             pot            hemo
##              19              17              87              88              52
##             pcv              wc              rc             htn              dm
##              71             106             131               2               2
##             cad           appet              pe             ane classification
##               2               1               1               1               0
```

Imputing missing values

To Impute the missing values I used the MICE library available for R. Now we can recheck to see that there are no missing values:

```
##            age             bp             sg             al             su
##              0              0              0              0              0
##            rbc             pc            pcc             ba            bgr
##              0              0              0              0              0
##             bu             sc            sod            pot           hemo
##              0              0              0              0              0
##            pcv             wc             rc            htn             dm
##              0              0              0              0              0
##            cad          appet             pe            ane classification
##              0              0              0              0              0
```

Here we can see that there are no missing data. Hence proceeding with data normalization:

- For columns with a small proportion of missing data (say, less than 10%), simple imputation methods like using the mean (for numerical features) or mode (for categorical features) can be effective. This is because the risk of introducing significant bias is relatively low when the amount of missing data is minimal.
- MICE for High missingness and Complexity: For columns with a higher degree of missingness (over 20%), especially those like rbc, rc, wc, pot, sod, which also potentially have complex relationships with other variables, MICE would be more appropriate. MICE can better account for the underlying patterns and relationships in the data, which is crucial when dealing with significant missingness.

Dataset Characteristics: Given that my dataset is related to kidney disease, it's likely that many features are interrelated, and their relationships might be important for understanding the disease and making predictions. This complexity makes a strong case for using MICE for columns with more substantial missingness or complex interactions.

Normalization

PCA

```
## Importance of components:
##                           PC1   PC2     PC3     PC4     PC5     PC6    PC7
## Standard deviation     2.195 1.296 1.13343 1.08483 0.97376 0.95952 0.9211
## Proportion of Variance 0.344 0.120 0.09176 0.08406 0.06773 0.06576 0.0606
## Cumulative Proportion  0.344 0.464 0.55578 0.63984 0.70757 0.77333 0.8339
##                           PC8     PC9   PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.75030 0.74119 0.6523 0.53253 0.47341 0.42980 0.30772
## Proportion of Variance 0.04021 0.03924 0.0304 0.02026 0.01601 0.01319 0.00676
## Cumulative Proportion  0.87414 0.91338 0.9438 0.96403 0.98004 0.99324 1.00000
```

Data Partitioning and Preprocessing:

Split ratio of 80% Training and 20% Testing:

Training the Models with split ratio of 80% Training and 20% Testing :

The performance of the models is summarized below:

- Logistic Regression: Accuracy - 0.9875, Specificity - 1, Sensitivity - 0.98.
- Decision Tree: Accuracy - 1, Specificity - 1, Sensitivity - 1.
- SVM: Accuracy - 1, Specificity - 1, Sensitivity - 1.

Training the Models with split ratio of 70% Training and 30% Testing :

The performance of the models is summarized below:

- Logistic Regression: Accuracy - 0.9833333, Specificity - 1, Sensitivity - 0.9733333.
- Decision Tree: Accuracy - 0.9916667, Specificity - 0.9777778, Sensitivity - 1.
- SVM: Accuracy - 1, Specificity - 1, Sensitivity - 1.

Training the Models with split ratio of 75% Training and 25% Testing :

In the 75-25 split evaluation of kidney disease data, the models of Logistic Regression, Decision Tree, and SVM demonstrated high efficacy. The Logistic Regression model achieved an accuracy of 0.989899, with sensitivity at 0.983871 and specificity at 1. This indicates its robust capability in correctly classifying both 'ckd' and 'notckd' cases. The model's positive predictive value (PPV) and negative predictive value (NPV) were exceptionally high, reinforcing its reliability in making predictions. The Decision Tree model showed an accuracy of 0.989899, a sensitivity of 1, and a specificity of 0.972973. These figures represent a strong performance, particularly in correctly identifying 'ckd' cases (since high sensitivity). The SVM model achieved a perfect score across all metrics, with an accuracy, sensitivity, specificity, PPV, and NPV all at 100%. This indicates its really good ability to classify the dataset without any errors.

These results suggest that all three models are highly effective for this particular dataset. Here the SVM model seems to be the best model with the best accuracy. However, the perfect scores in SVM, as in the previous split ratios, raise a potential concern for overfitting. This underlines the importance of further validation and testing, particularly for the SVM model, so we have to ensure its generalizability and robustness in different datasets and conditions.

Given these observations, the 75-25 split appears to be marginally better for Logistic Regression and SVM, while the 80-20 split seems slightly more favorable for the Decision Tree. However, the differences are minimal, suggesting that all split ratios are generally effective for this dataset. Since Random forest is essentially an ensemble of decision tree models I will go ahead with a 80-20 split for this model.

Random Forest as an ensemble model wiht 80:20 split dataset:

```
## Random Forest
##
## 320 samples
##  24 predictor
##   2 classes: 'ckd', 'notckd'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 288, 288, 288, 288, 288, 288, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa
##   1     0.990625  0.9803279
##   2     0.996875  0.9934426
##   3     0.996875  0.9934426
##   4     1.000000  1.0000000
##   5     0.996875  0.9934426
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

The Random Forest model evaluation for the kidney disease dataset with 320 samples, 24 predictors, and two classes ('ckd' and 'notckd') shows really good performance. This evaluation involves a 10-fold cross-validation

process without any data pre-processing. The use of 10-fold cross-validation, is a robust method for assessing model performance and ensuring that the evaluation is not biased towards a specific subset of the data.

Key observations from the resampling results across different tuning parameters are as follows:

Performance Metrics: The model achieved near-perfect to perfect accuracy across different 'mtry' values: With different mtry values, the accuracy was as displayed in the output with their corresponding Kappa statistic until the model reached a perfect accuracy of 100% and a Kappa statistic of 1.0000.

Model Selection: The model with the highest accuracy was selected as the optimal model. Despite several 'mtry' values yielding a perfect accuracy of 100%, the final model chosen as per the selection criteria for the optimal model using the largest accuracy value.

Interpretation: The high accuracy and Kappa values indicate that the Random Forest model is extremely effective in distinguishing between the two classes of the dataset. The suggested Kappa statistic near to the chosen mtry value suggests that the model's predictions are not only accurate but also significantly better than chance-level predictions.

Creating an ensemble model function :

```
## [1] "Ensemble Model Results:"
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ckd notckd
##     ckd     50      0
##     notckd   0     30
##
##                Accuracy : 1
##                  95% CI : (0.9549, 1)
##     No Information Rate : 0.625
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.000
##             Specificity : 1.000
##          Pos Pred Value : 1.000
##          Neg Pred Value : 1.000
##              Prevalence : 0.625
##          Detection Rate : 0.625
##    Detection Prevalence : 0.625
##       Balanced Accuracy : 1.000
##
##        'Positive' Class : ckd
##
```

The approach here is to use an ensemble model for classification task is to use a voting system where the final class is determined by the majority vote from the individual models.

Here the function Trains the three models (Logistic Regression, Decision Tree, and SVM) on the training data and defines a function ensemble_predictions that:

- Takes test_data as input.
- Generates predictions from each model.

- Combines these predictions and decides the final prediction based on majority voting.
- Evaluates the performance of the ensemble model using a confusion matrix.

Interpretation of the ensemble model prediction :

- Accuracy: The model achieved an accuracy of 100% (1.0). This means that every prediction made by the model, whether for the 'ckd' or 'notckd' class, was correct.
- 95% Confidence Interval: The 95% confidence interval for accuracy is between 95.49% and 100%. This high interval indicates strong confidence in the model's accuracy.
- No Information Rate (NIR): The NIR is 62.5%, and the model's accuracy is significantly better than this rate (p-value $<$ 2.2e-16), suggesting that the model's predictions are highly reliable and not due to chance.
- Kappa: The Kappa statistic is 1, indicating perfect agreement between the model's predictions and the actual values and signifies that the model's performance is not due to random chance.
- Sensitivity and Specificity: Both are at 100% (1.000). Sensitivity (or True Positive Rate) measures the proportion of actual positives correctly identified, while Specificity (or True Negative Rate) measures the proportion of actual negatives correctly identified.

Since this model uses majority voting and is a combination of individual base models, this ensemble model has better metrics for the classification task at hand compared to any of the models developed above (although most of the models did perform exeptionally well). The ensemble model combining Logistic Regression, Decision Tree, and SVM shows superior performance with an accuracy of 100%. This suggests that the ensemble approach is effective in this case, potentially offering better predictive performance than individual models.

Conclusion:

After evaluation of each model performance over various train and test split ratios, I decided the split ratio of 80:20 for training the Random Forest model (which is an ensemble of various decision tree models) and also the ensemble model using logistic regression model, decision tree model and SVM models. Here we finally developed an ensemble model which can now classify and predict whether an individual has chronic kidney disease or not, ensuring early detection and intervention of chronic kidney disease (CKD) which can significantly improve patient outcomes.

Reference:

Rubini,L., Soundarapandian,P., and Eswaran,P.. (2015). Chronic_Kidney_Disease. UCI Machine Learning Repository. https://doi.org/10.24432/C5G020.