STATS 4M03/6M03: Multivariate Analysis

Final Project

Chronic Kidney Disease Analysis

Submitted to

Dr. Eman M.S. Alamer

Department of Mathematics and Statistics

McMaster University

Hamiltion, Ontario, Canada L8S 4K1

November 24, 2023

Reported by

Chenning Chu (400272675)

Weipeng Hu (400292876)

Ying Cui (400296443)

Yifei Li (400276294)

1 Introduction

1.1 Abstract

Ammirati, A.L. describes chronic kidney disease as a long-term condition where the kidneys lose their ability to filter wastes and excess fluids (2020). The dataset on chronic kidney disease used in this project was obtained from Kaggle. It contains multiple interrelated response and predictor variables. Through multivariate analysis, it is possible to discern which variables are closely correlated with kidney failure. This analytical approach contributes to a deeper understanding of the disease and aids individuals in predicting and then preventing its onset. Furthermore, for patients, treatment plans can be optimized and tailored to their specific variables, enhancing personalized care effectiveness.

1.2 The Data

The data comprise 26 columns, 400 observations, and variable types that are numeric and character. Figure 1 provides a data description.

age: age	bp: blood pressure	sg: specific gravity	al: albumin	su: sugar
rbc: red blood cells	pc: pus cell	pcc: pus cell clumps	ba: bacteria	bgr: blood glucose random
bu: blood urea	sc: serum creatinine	sod: sodium	pot: potassium	hemo: haemoglobin
pcv: packed cell volume	wc: white blood cell count	rc: red blood cell count	htn: hypertension	dm:diabetes mellitus
cad: coronary artery disease	appet: appetite	pe: peda edema	ane: aanemia	classification: class

Figure 1: data description

1.2.1 Exploratory Data Analysis(EDA)

The statistical summary for numeric variables includes key metrics like the minimum, maximum, median, mean, standard deviation, and the count of missing values. For character variables, it represents information on length, class, and mode. Histograms in the diagonal of the pairs plot

depict the distribution of numeric variables. Histograms for variables such as age, sg, hemo, pcv, and rc show nearly symmetric shapes, suggesting the use of mean values for data cleaning. Conversely, variables like bp, al, su, bgr, bu, sc, sod, pot, and wc exhibit skewed histograms, indicating the suitability of using median values. The histograms for su, hemo, pcc, and rc are effectively presented. In the upper triangular section of the pairs plot, the correlation plot illustrates relationships between variables, showing strong positive correlations between variables like bgr and su, hemo and sg, etc., and strong negative correlations between variables like pcv and al, sod and sc, etc. Figure 3's bar plot, by comparing numeric variables to the classification of chronic kidney disease, explains the findings. There is insufficient evidence to suggest that variables like sg, sod, hemo, and pcv significantly impact chronic kidney disease. Age, bp, and rc appear to have a weak effect on the disease. Conversely, bgr, bu, sc, pot, and wc have a strong influence. The plot provides sufficient evidence to indicate that al and su affect the disease. Figure 4's pie chart illustrates the distribution of chronic kidney disease classification in the dataset: 38% without and 62% with the condition.

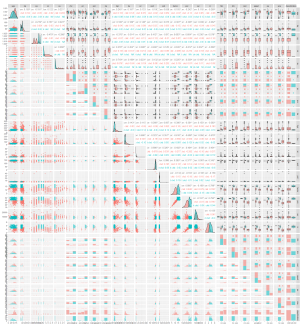


Figure 2: Pairs Plot

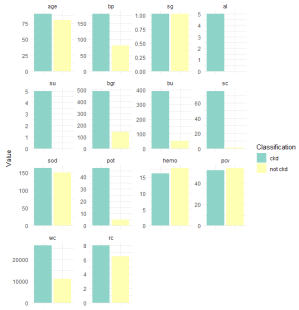


Figure 3: Bar Plot

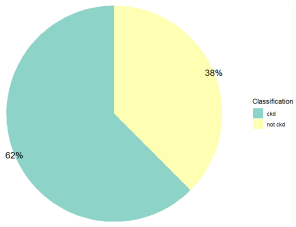


Figure 4: Pie Chart

1.2.2 Data Preparation

The initial data review for the kidney disease dataset focused on understanding the dataset structure by examining the head of the data, which revealed the 'id' column as a mere row identifier. This column was deemed irrelevant for analysis and subsequently removed using an R command. Additionally, renaming the columns from abbreviations to their full descriptive names aid in better comprehension for subsequent analysis.

Further inspection of the dataset structure was undertaken to verify the correct data types. The discovery that columns labeled as 'pcv', 'wc', and 'rc' were character types rather than numerical prompted their conversion using R's data manipulation capabilities.

The final stages of data preparation included handling missing values through appropriate imputation strategies for both numerical and categorical columns, informed by an analysis of pairs plots in the exploratory data analysis (EDA) segment. Feature engineering and dataset refinement were performed to further tailor the dataset for analysis, with additional steps like scaling or normalization. The dataset was then split into training and testing sets, using stratified and random sampling techniques to ensure a balanced representation, thus preparing the data for the crucial task of model training. Then we are ready for in-depth analysis and machine learning applications. We are using R statistical with version 2023.06.2+561.

2 Methodology

2.1 KNN

Our first method is the k-Nearest Neighbours Classification (KNN). KNN classifies unlabelled observations by assigning them to the class that has the most labeled observations in its neighbourhood with size k. The parameter k denotes the number of nearest labelled observations surrounding the unlabelled observation we wish to classify. We begin by selecting the value of k through 10-fold cross-validation, which is a method that helps estimate the error rate of a method in order to evaluate its performance and also assists in choosing appropriate parameters. The output summary and the plot of cross-validation (Figure 5) indicate that the best parameter is $k = 7$, corresponding to the lowest error rate of 0.02666667. With $k = 7$ set as the parameter, we apply KNN classification to make predictions on the test set.

2.2 Bagging

We are using Bagging (Bootstrap Aggregating) as our second method. Bagging can be used to fit a learning method to each bootstrap sample and subsequently averaging or combining the resulting predictions. Its purpose is to enhance the stability and accuracy of models by mitigating

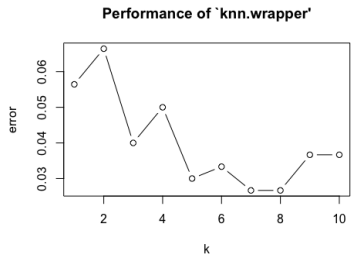


Figure 5: KNN error rate

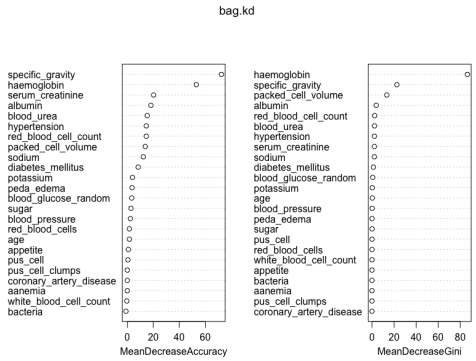
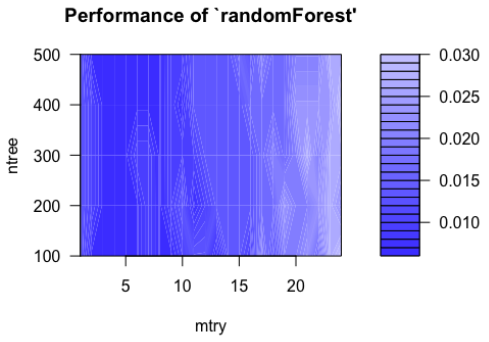


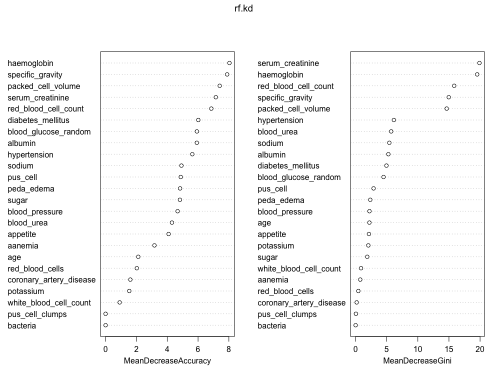
Figure 6: Variable Importance Plot for Bagging

variance and overfitting. We are using the RandomForest algorithm for Bagging and take mtry equal to 24 since we have 24 features in our dataset. We also draw the Variable Importance Plot (Figure 6) which is a dotchart of how important each predictor variable is when doing predictions. In this VarImpPlot, we can see the three most important variables are "specific_gravity", "haemoglobin" and "serum_creatinine" (highest MeanDecreaseAccuracy); the two least important variables are "bacteria" and "white_blood_cell_count" (lowest MeanDecreaseAccuracy).

2.3 Random Forest



(a) Random Forest error rate



(b) Variable Importance Plot for Random Forest

Figure 7: Random Forest analysis results

We use Random forest as the third method which is an extension of bagging. The random forest model combines multiple decision trees. Random forest takes a random sample of M predictors at

each split, for each tree. So random forest involves randomness and also decorrelates the trees in the ensemble. This randomness and the uncorrelated trees made this method less likely to overfit. We start with using cross-validation to select the value of parameters mtry and ntree which represent the number of variables randomly sampled at each split and the number of trees respectively. The 10-fold cross-validation gives us the best parameters is mtry = 2 and ntree = 100 with the lowest error of 0.006666667. The plot given by cross-validation (Figure 7a) shows the same thing visually. After getting the parameter, we start to build the Random Forest model. Then we also draw the Variable Importance Plot (Figure 7b). We can see the three most important variables are "haemoglobin", "specific_gravity" and "packed_cell_volume"; the two least important variables are "bacteria" and "pus_cell_clump" (lowest MeanDecreaseAccuracy).

2.4 Logistic Regression

The last method we used is fitting a logistic regression model in order to explore the relationship between the response variable "class" and predictor variables. Logistic regression estimates the probability of a binary outcome by transforming the linear combination of predictor variables using logistic function.

We build one model with all variables (full model) and one only uses several of the most and least important predictor variables (reduced model) which are obtained from the VarImpPlots we did above. The reduced model demonstrates that variables "specific_gravity" and "serum_creatinine" have three asterisks which means they are significant at the 0.001 level. The variables "haemoglobin" and "packed_cell_volume" have two asterisks which means they are significant at the 0.01 level. In addition, the odd ratio tables show the estimated odd ratio and confidence intervals for each predictor variable. To conclude, the patient is more likely to get chronic kidney disease as the value of specific gravity, serum creatinine, haemoglobin and packed cell volume increases. In the end, we compare the full model and the reduced model.

3 Discussion

To find the best performance classification methods for our dataset, we could compare the Adjusted Rand index (ARI) and the misclassification rate (MCR). Since ARI = 1 and a low mis-

Method	Parameter	ARI	MCR
KNN	k=7	0.9594986	0.0101
Bagging	mtry=24	0.9594532	0.0101
Random Forest	mtry = 2,ntree=100	0.9594986	0.0101

Figure 8: data description

classification rate means perfect classification, we want ARI as close to 1 and MCR as low as possible. The table above compares ARI and MCR of our three classification methods. By observing the result, we find that all three ARIs are close to 1, where KNN and RandomForest appear the same as highest, and all the MCRs are pretty low. KNN and Random Forest methods have slightly higher ARI, they are slightly better well-performing classification models than Bagging, and they all made accurate predictions on our dataset. Variables "specific_gravity", "serum_creatinine", "haemoglobin" and "packed_cell_volume" have a strong relationship with the response variable by Logistic Regression. Therefore, we could consider these variables as the major factors in further research. Also, we compare the Residual deviance between full and reduced models. The p-value is 1.299745e-05, which quite low. So that we reject the null hypothesis(the reduce model performs the same as full model). So we choose to use the reduce model to make prediction because it is better.

4 Conclusion

Throughout this project, we have deployed several machine learning algorithms to predict chronic kidney disease, conducting a thorough analysis of their efficacy. Meticulous data preprocessing, application of algorithms, and performance evaluation revealed that the k-Nearest Neighbors, Bagging, and Random Forest methods all demonstrated high precision in predicting chronic kidney disease. The calculated Adjusted Rand Index and Misclassification Rate results indicated the effectiveness of our chosen models in handling this type of medical data. This finding is not only significant for early diagnosis and treatment of the disease but also serves as a valuable reference for further research. Future studies can build upon this foundation to refine the models and extend their application to other medical datasets.

5 Bibliography

References

[1] Ammirati, A. L. (2020). Chronic kidney disease. *Revista da Associação Médica Brasileira*, 66, s03-s09.

[2] Chronic Kidney Disease dataset. (n.d.). Www.kaggle.com. Retrieved November 24, 2023, from https://www.kaggle.com/datasets/mansoordaku/ckdisease/data

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

[4] R Core Team (2023.06.2+561). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

[5] GeeksforGeeks. (n.d.). Taking Input from User in R Programming. Retrieved from https://www.geeksforgeeks.org/taking-input-from-user-in-r-programming/

[6] DataMentor. (n.d.). R Programming Examples – User Input. Retrieved from https://www.datamentor.io/r-programming/examples/user-input

[7] DataMentor. (n.d.). Plot Function in R. Retrieved from https://www.datamentor.io/r-programming/plot-function

[8] Hubert and Arabie. (1985). Comparing partitions | semantic scholar. (n.d.). https://www.semanticscholar.org/paper/Comparing-partitions-Hubert-Arabie/b756db6dd3928f985275bf77a36f4e6e2d8f4cfd/

[9] R-bloggers. (2021, April). How to Clean the Datasets in R. Retrieved from https://www.r-bloggers.com/2021/04/how-to-clean-the-datasets-in-r/

[10] Zhou, Z.-H. (2012), *Ensemble Methods: Foundations and Algorithms*, Boca Raton: Chapman & Hall/CRC Press.