

Chronic kidney Disease: Part 1

By: Davis Hanai, Michael Cali, Quynh-Tram Tran, and Wynter Gray



Today’s world is data-driven. Data is essential to all organizations. In the case of Chronic Kidney Disease (CKD) and the healthcare industry, data analysis saves lives. According to the National Kidney Foundation, this silent killer is “the 8th leading cause of death in the U.S. [and] about 9 in 10 adults with kidney disease (≈90%) do not know they have it.” (kidney.org, 2025) Routine screenings often fail to detect this diagnosis. CKD is an incurable disease and therefore, early detection is of utmost importance.

Healthcare professionals can leverage the power of predictive modeling to help predict whether a patient has CKD or not. This can be achieved by taking a holistic view of the patients’ health records, utilizing a combination of their age, vital signs, lab results from their metabolic function panel, complete blood count panel, urinalysis, and other relevant health markers. This would enable the patients to receive better care and early detection of the disease. Review of current disease state knowledge identifies the following as most often accompanying a diagnosis of CKD: increased serum creatinine and blood urea nitrogen (BUN), anemia, albuminuria, electrolyte imbalances (later-stage CKD), hypertension, and diabetes.

The data set for Project Kidney Disease contains 400 instances and twenty-five columns that identify patients with or without chronic kidney disease (CKD), as well as accompanying laboratory results and comorbid conditions from a hospital in nearly a 2-month period. To allow for more collaboration, the original dataset from UC Irvine is used. Multiple distribution graphs (Figure A) are created to understand the data within. In the Roadmap to Machine Learning, eight steps are listed: Framing the Problem, Get the Data, Explore the Data, Wrangle the Data, Identify Promising Models, Tune the Models, Present Solution, and Launch. Through this roadmap, a successful predictive model can be built.

Goal setting is the first step toward Framing the Problem. The goal of the machine learning model is to accurately predict the presence of CKD and identify patients quickly and more efficiently based on available data. The data is run through several machine learning techniques in order to achieve this goal. A successful model will be defined as one that achieves an accuracy score greater than 0.90 on the test dataset, an F1 score, and a confusion matrix to determine levels of false positives and false negatives. The dataset is assumed to contain missing data, possibly incorrect data types, outliers, and correlated fields. A preliminary review of the dataset revealed numerous categories with a significant number of null values, especially noted in rbc, wbcc, and rbcc.

Additionally, many binary and categorical variables are encoded and mapped to both the training and testing set as text, which will require standardization before they can be used effectively in modeling. Imputation for categorical columns is also used. Chronic kidney disease is a multifactorial disease state that affects many other aspects of overall health. It can also be assumed that not all lab values in the dataset are equally associated with the presence of CKD. Certain comorbid conditions or laboratory findings, such as serum creatinine, age, albumin, and hypertension will be more strongly associated with CKD than others. By using the .info() function, missing data can be identified and are replaced with the median values; however, all data types seem to be correct. After, outliers were quantified using the first, third and interquartile ranges (IQR) and were imputed. Next, a correlation matrix (Figure B) is used in efforts to avoid multicollinearity. Several columns are found to be interdependent or highly correlated. These columns, including ‘rbcc’, ‘rbc’, ‘wbcc’, ‘pcv,’ are dropped to create a better and more stable model. After that, Step Two: Get the Data begins.

A shared Jupyter Notebook is used to further foster collaboration in addition to Pandas read\_csv function is utilized. All authorizations are given, the workspace is created, and sensitive information is protected. To further prove sensitive information is protected, according to the *Kidney Disease: Improving Global Outcomes (KDIGO)* guidelines, CKD is most commonly diagnosed when estimated glomerular filtration rate (eGFR) or Creatinine Clearance (CrCl) is below 60ml/min/1.73m^2 for >=3months. However, this dataset does not include key variables such as sex, weight, and sometimes race, which are necessary for calculating eGFR or CrCl. Therefore, we are unable to derive these clinical indicators directly from the data. Finally, the train\_test\_split package from the sklearn.model\_selection library is used with the ‘class’ column used as the target with an 80% (Training)/ 20% (Testing). There seems to be more positive CKD records than negative, however, through stratification, this imbalance is addressed, prevents skewed results, and balances the proportions of positive and negative results. Data visualizations are used extensively.

Some data visualizations include ROC curves, which will be explored more in Part 2, various boxplots to detect outliers (Figure C), and a confusion matrix after modeling were generated from the notebook and further confirmed high levels of accuracy. These figures can be seen in the notebook. Confusion Matrix gives accuracy percentages to get details on how accurate the model is and using the different machine learning methods to gain insights into what method is best. The settings could be changed to help improve the model. However, these are all factors and methods to be considered. This concludes Step 3 (Explore the Data) and allows for the progression to Step 4 (Wrangle the Data).

The Chronic Kidney Disease dataset is wrangled in a shared Jupyter notebook for all team members and various functions are created. Missing data, outliers, feature engineering, and unnecessary information are wrangled and handled in the notebook for better analysis. The .describe() function is used to understand the data after the imputation and data manipulation, as seen in Figure D. Some tools and methods used include Pandas, Seaborn, and Matplotlib libraries, imputation methods, correlation matrix to verify multicollinearity, and train\_test\_split for machine learning techniques. This allows for a more accurate model to be built. By utilizing these techniques, Logistic Regression, SVM, and Random Forest are set for use.

Appendix:

Figure A:

A group of blue and white graphs

AI-generated content may be incorrect.

A graph of a number of numbers and a line

AI-generated content may be incorrect.

Figure B:

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure C:

A comparison of a graph

AI-generated content may be incorrect.A screenshot of a graph

AI-generated content may be incorrect.A diagram of a box plot

AI-generated content may be incorrect.A comparison of a box plot

AI-generated content may be incorrect. A diagram of a box plot

AI-generated content may be incorrect.A comparison of a box plot

AI-generated content may be incorrect.A diagram of a box plot

AI-generated content may be incorrect.

Figure D:

A screenshot of a computer

AI-generated content may be incorrect.

References:

Shmueli, Galit, et al. Data Mining for Business Analytics: Concepts, Techniques, and Applications with JMP Pro. John Wiley & Sons, Inc., 2017. Accessed 30 Jul. 2025.

“Kidney Disease: Fact Sheet.” *National Kidney Foundation*, 10 Dec. 2024, www.kidney.org/about/kidney-disease-fact-sheet.