Using Machine Learning to predict first stage of Chronic Kidney Disease "CKD"

Lama Albarqawi
MS. Computing student
Salt Lake City, UT
Lama.albarqawi@hsc.utah.edu

Yulong Liang
MS. Computing student
Salt Lake City, UT
yulong.liang@utah.edu

ABSTRACT

UPDATED—29 April 2018. This report describes how utilize usage of Machine learning magnificent algorithms to predict severe and serious health care issues. The project studies a dataset collected in an Indian hospital. It includes 250 patients that were already diagnosed with CKD and 150 individuals who do not have CKD.

Author Keywords

CKD; first stage CKD; Chronic Kidney Disease; Machine Learning; CKD prediction; predicting diseases using Machine Learning; Machine Learning and Healthcare

INTRODUCTION AND MOTIVATION

Chronic Kidney Disease (CKD) is a condition in which kidney function deteriorates, allowing blood waste to accumulate in and damage the body, CKD progresses gradually and causes a gradual loss of kidney function over time. Progression of CKD stages can be slowed or prevented by early detection and control of risk factors, such as arterial hypertension and proteinuria, by tight blood pressure control and inhibition of the renin-angiotensin system. To this purpose, early referral to a nephrologist is important to identify patients at risk and provide individualized and comprehensive care aimed to slow disease progression and limit or prevent the occurrence of CKD advanced stages and related complications.

CKD has five stages, individuals with early stages often do not experience noticeable symptoms. However, if left untreated, the disease progresses to kidney failure, at which point the only treatment options are regular and costly dialysis, or kidney transplant.

ADVANTAGES OF EARLY PREDICTION

- Enhancing the quality of life for the individuals by trying to delay the progression of CKD (e.g. controlling Blood Pressure, losing weight, controlling protein in urine, Low potassium diet, water pill, fix acid levels) and preventing complications of the advanced stages.
- Preventing the need for Dialysis and/or Kidney Transplant. Both of these treatments can result in a huge burden on the patient, physically, psychologically and economically.
- CKD often has no symptoms in its early stages and can go undetected until it is very advanced. That said, early prediction can prevent the need for an emergency, unscheduled dialysis treatment at a

hospital, which can cost around $9,900 for a single treatment. In case the patient was not diagnosed earlier and is not aware of his medical condition. Which if was predicted earlier, would dramatically decrease such costs on both individuals and insurance companies.

ULTIMATE GOAL OF CKD PREDICTION

Raising a red flag for the individuals who are at high risk, informing them with the urgent need to go and see a nephrologist, in order to prevent further complications.

THE MOTIVATION - WHY TO USE MACHINE LEARNING TECHNIQUES? WHY NOT THE TRADITIONAL OR EXISTING METHODS?

Machine learning is all about developing mathematical and computational methodologies for learning and extracting insights from data and discovering patterns hidden within these data. The more data provided for the machine learning algorithm is the better. Hence, healthcare is a fertile ground for machine learning, since its very rich with patients data. Predicting diseases by just studying bunch of data features (e.g. vital signs and other measures) and trying to discover patterns by studying same features of the already diagnosed patients, is a task that humans or ordinary automated tools would struggle with. Another point is that humans or the programs that the human brain develop, will focus on what the human already knows, and will not search for other signs and indicators, in other words, they will not find new patterns and will only focus on a number of specific signs that they expect a patient at a risk would have.

OUR SOLUTION / WORK DESCRIPTION

We were eager to explore machine learning packages with their built-in libraries, as well as trying to study the algorithms that we have already learned and implemented from scratch during the semester, and how would they act in training and testing the dataset.

DATA SHUFFLING

The raw data was segmented with positive examples in the first half (patients diagnosed with CKD) and negative in second half (patients diagnosed with noCKD). Thus, we have **shuffled** the data to ensure randomized order so that we can guarantee order will not affect the prediction of any algorithm used.

MISSING VALUE IMPUTING

The raw dataset had many features with missing values, we have followed **5 strategies** for imputing them.

1. Removing the features with most of the value missing first and then removing the data points with missing values
2. For numeric data, imputing with the mean of the whole data set. For categorical data, creating a new value to indicate missing.
3. For numeric data, imputing with the mean of the whole data set. For categorical data, imputing with the mode of the whole data set.
4. For numeric data, imputing with the mean of the whole data set. For categorical data, using a hybrid strategy. Namely imputing with new value for the features which have more missing values and mode of the whole data set for the feature which have fewer missing values.
5. For numeric data, imputing with the mean of the data with the same label. For categorical data, using a hybrid strategy. Namely imputing with new value for the features which have more missing values and mode of the data with the same label for the feature which have fewer missing values.

We wanted to explore how those strategies would affect the prediction performance. In order to do that, we chose to use **three base algorithms** to do the evaluation: Logistic Regression, Decision Tree, and

Random Forest. We used accuracy and f1 score for both training data and testing data as **metrics**.

Algorithm	Metric	S1	S2	S3	S4	S5
Logistic Regression	Training Accuracy	1	1	0.994	1	1
	Training F1	1	1	0.995	1	1
	Testing Accuracy	0.971	1	0.958	0.972	0.99
	Testing F1	0.965	1	0.966	0.978	0.992
Decision Tree	Training Accuracy	1	1	0.99	0.99	0.999
	Training F1	1	1	0.992	0.992	0.999
	Testing Accuracy	0.995	1	0.968	0.958	0.995
	Testing F1	0.995	1	0.974	0.967	0.996
Random Forest	Training Accuracy	1	1	1	1	1
	Training F1	1	1	1	1	1
	Testing Accuracy	1	1	0.978	0.995	1
	Testing F1	1	1	0.982	0.996	1

Table (1): Comparison among imputation strategies

From the table above, we made the following discoveries for this particular case:

- The strategies have the following ranking in terms of the contribution to the prediction accuracy:

  S2 > S1 > S5 > S4 > S3
- Strategy S2, which is imputing with the mean for numeric values and with a new value for categorical values works the best.
- Strategy S3, which imputing with the mean for numeric values and with the mode of the whole data set for categorical values works the worst.
- The comparison might be different for other data set. So, we need to try different strategies when dealing missing values.

DATA SCALING

We have done reasonable amount of research on Scaling Data for Machine Learning, and comparing the effect of different scalers with respect to dataset size and the cases in which the researcher might need to rescale data. After discussing if we should apply rescaling techniques and after plotting charts for our dataset features, we have agreed that there is no need for applying them into this project, since the dataset is not huge, and the features do not contain such considerable amount of data outliers, additionally, the positive-negative ratio is realistic and is not biased to specific label.

DATA DISTRIBUTION

Next, we thought that before starting with the algorithms implementations, plotting the features data might help us understand our data better, and give us some insights as well. So, we have proceeded with running 3 helpful pandas functions to help us with that, we have started by plotting the features distribution using histograms (see Figure 1), applying this to the data that we have got with the ultimate imputation strategy for missing values.

As we can see from 1, the plots met our expectations, as most of the patients were above 50 years old (around 64%) which is expected because potential of having CKD increases with aging.

Then, we have plotted Features Density (see Figure 2). These plots were more helpful than histograms in trying to imagine the population data meaning, since they give more precise domains, and more info on the exact number of patients who have the maximum values of proteins for example, or number of red blood cells, white blood cells, or sugar levels and hypertension.

After that, we have plotted features box plots (see Figure 3), in order to test outliers mainly, it helped us in making the decision of not doing rescaling for our data.

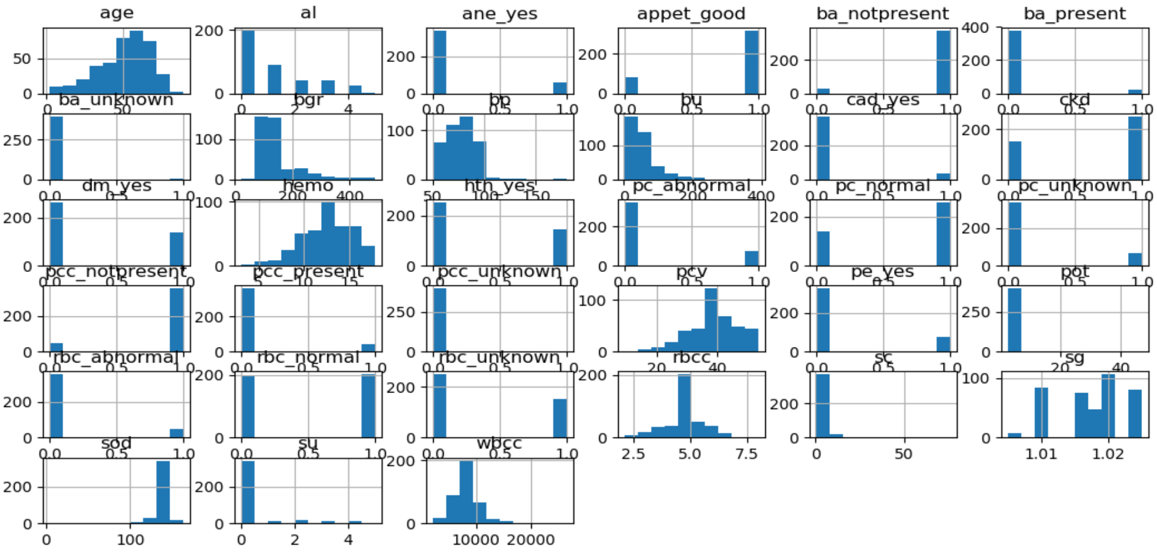

Figure 1: Features Distribution histograms for the second missing values imputation strategy

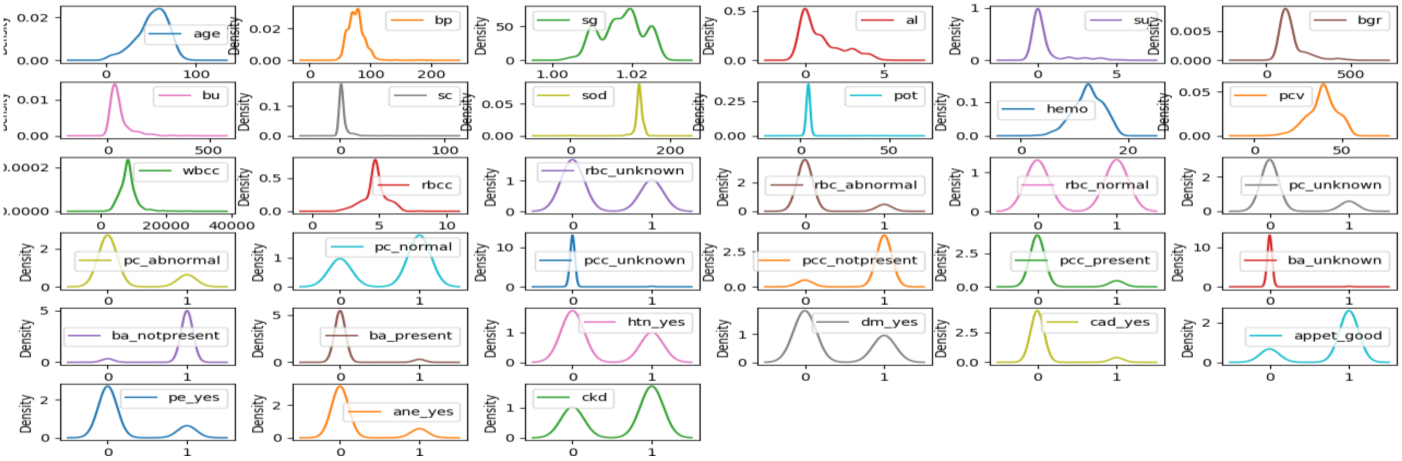

Figure 2: Features Density plots for the second missing values imputation strategy

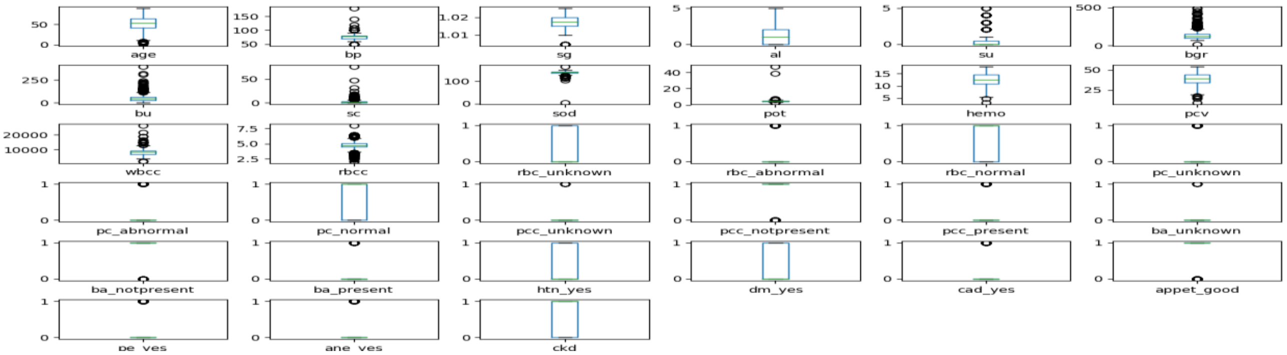

Figure 3: Features Box plots for the second missing values imputation strategy

CROSS VALIDATION

For dataset splitting, we have chosen to apply **k-fold cross validation with 5 splits**. This strategy is more accurate than the simple train test split technique, which only specify the percentage of test dataset without dynamically changing the examples of the train/test.

FITTING INTO A MODEL

We have applied the following **13 different learning algorithms** to make comparisons with the ultimate imputation strategy.

1. Logistic Regression
2. Perceptron
3. Linear SVM
4. Polynomial SVM
5. Gaussian SVM
6. Decision Tree
7. Random Forest
8. XGBoost with Decision Tree
9. XGBoost with Linear Classifier
10. AdaBoost with Decision Tree
11. AdaBoost with Linear Classifier
12. Bagging with Decision Tree
13. Bagging with Linear classifier

To ensure a comprehensive evaluation, we chose **five metrics**: fit time, training accuracy, training f1 score, testing accuracy, and testing f1 score.

	Fit Time	**Train Accuracy**	**Train F1**	**Test Accuracy**	**Test F1**
Logistic Regression	0.003	1	1	0.99	0.992
Perceptron	0.001	0.66	0.777	0.647	0.762
Linear SVM	4.658	0.997	0.997	0.978	0.982
Polynomial SVM	8.809	1	1	0.993	0.994
Gaussian SVM	0.005	1	1	0.625	0.769
Decision Tree	0.001	0.999	0.999	0.995	0.996
Random Forest	0.049	1	1	1	1
XGBoost with Decision Tree	0.012	1	1	0.997	0.998
XGBoost with Linear Classifier	0.005	0.999	0.999	0.997	0.998
AdaBoost with Decision Tree	0.061	1	1	0.997	0.998
AdaBoost with Linear Classifier	0.119	0.993	0.994	0.985	0.987
Bagging with Decision Tree	0.044	0.996	0.997	1	1
Bagging with Linear Classifier	0.016	0.994	0.995	0.982	0.986

Table (2): Comparison among machine learning algorithms

From the table above, we discovered the following facts with respect to **time complexity**:

- At least with Sciki-learn, Linear SVM and Polynomial SVM are very time-consuming.
- Ensemble methods require more time to fit than single classifier.

We also discovered the following facts regarding the **prediction performance**:

- The data is nearly linear separable.
- All the methods performed very well except Perceptron and Gaussian SVM.

- Ensemble methods always have a better result than single classifiers.
- Random Forest performs the best among all the algorithms.

FEATURE SELECTION:

EXPERT vs. STATISTICIAN vs. DATA SCIENTIST

As a last stage, we wanted to test the degree on which machine learning algorithms can discover chronic kidney disease at its first stage, and examine the extent of its ability to predict CKD.

Explicitly, we wanted to study whether feature selection can reduce the time consumed while keeping the prediction performance. We also wanted to compare the feature selection strategies from an expert (a nephrologist), a statistician and a data scientist.

The approach is as follows:

- First, features selection from the perspective of **a nephrologist**.
  Based on our previous research that we have done to study CKD along with its most dominant factors and main medical predictors, we have deleted the features that are less dominant. As a result, we have kept 16 features left, after they were 31 features, which means we have decreased the number of features to almost the half.
- Second, feature selection with **Filter Method**.
  Then, we selected the same amount (16) features using a statistical approach – Filter Method with p-value, which is the most commonly used feature selection technique. The idea is to study the correlation between the features and label and select the features with higher correlations.
- Last, feature selection with **Wrapper Method**.
  Last, we selected the same amount (16) features using a data science approach – Wrapper Method with Recursive Feature Elimination, which is believed to be most accurate approach for feature selection. The idea is to run the base algorithm several rounds. Within each round, select the best feature or eliminate the worst feature until we get the desired number of features.

Algorithm	Metric	All features	Expert	Statistician	Data Scientist
Logistic Regression	Training Accuracy	0.99	0.988	0.99	0.988
	Training F1	0.992	0.99	0.992	0.99
	Testing Accuracy	1	1	1	1
	Testing F1	1	1	1	1
Decision Tree	Training Accuracy	0.997	0.977	0.995	0.995
	Training F1	0.998	0.982	0.996	0.996
	Testing Accuracy	0.999	0.989	0.999	0.999
	Testing F1	0.999	0.991	0.999	0.999
Random Forest	Training Accuracy	1	0.997	1	1
	Training F1	1	0.998	1	1
	Testing Accuracy	1	1	1	1
	Testing F1	1	1	1	1

Table (3): Comparison among feature selection techniques

From the table above, we can draw the following conclusions:

- Experts do not work as well as machine learning guys in feature selection.
- Statisticians and data scientist are equally good. Namely, Filter Methods and Wrapper Methods have very similar results.
- With the help of Filter Methods and Wrapper Methods, we can almost get the same prediction performance but with much smaller data dimension and much less time complexity.

- For small data set, we don’t need to bother selecting the features. However, for large scale data set, it is recommended to do feature selection before fitting into a machine learning model to reduce computational complexity.

FUTURE PLAN

1. Trying to get richer and bigger dataset, from different location and with more comprehensive features, trying to figure out and discover new factors, and examine the ability of machine learning and artificial intelligence to read between the lines and see what humans couldn’t.
2. Applying more splitting strategies and test when algorithms perform at their best and where do they fail more.
3. Making comparisons among all the scaling approaches, e.g., min-max scaling, quantile transformation, standardization and normalization and seeing how scaling will help to improve the machine learning model performance.