PREDICTING STATUS OF CHRONIC KIDNEY DISEASE (HEALTH CARE USE CASE)

Nikhil Kothari
Master's in Applied Computing
University of Windsor
Windsor, Ontario
kotha113@uwindsor.ca

Yash Joshi
Master's in Applied Computing
University of Windsor
Windsor, Ontario
joshi24@uwindsor.ca

***Abstract*— The Chronic Kidney Disease (CKD) is increasing rapidly day by day due to many reasons. Machine learning is one of the most growing field. This field is exploding with opportunities and career prospects. Because with more advancing, this field have helped to solve many complex problems in real world problems in several sectors such as physics, computer science, banking, education, medical and economics. With the help of machine learning algorithm, we tried to predict the chronic kidney disease So, this model will help to predict whether a person has kidney disease or not**

Keywords—dataset, data purification, prediction, exploratory data analysis, logistic regression, prediction

I. INTRODUCTION

One of the most critical disease is kidney disease. Based upon available reports, there is need of machine learning model which can identify whether a person is having any kidney disease or not. The first and foremost thing required is a dataset which contains report details of people. Hence, we considered the data of those patients that have a kidney disease or not.

Basically, the chronic kidney disease occurs due to impact of other disease or conditions that affect the functioning of kidney. Later, it leads to serious damage to kidney over months and years. Some of the diseases that leads to CKD are Type 1 or type 2 diabetes, high blood pressure, glomerulonephritis which is an inflammation of kidney's filtering units, interstitial nephritis which is an inflammation of kidney's tubules and surrounding structures, polycystic kidney disease, prolonged obstruction of the urinary tract due to enlarged prostate, kidney stones and some cancers, vesicoureteral reflux which is a condition causing urine to back up into kidneys and pyelonephritis which is recurrent kidney infection.[1]

Major risk factors causing CKD are diabetes, high blood pressure, cardiovascular disease, smoking, obesity, family history of kidney disease, abnormal kidney structure or even old age.[1]

II. DESCRIPTION OF THE PROBLEM

There is need to predict whether a particular person having all different properties can have a chronic kidney disease or not using some classification algorithm. There is need to understand data. From huge chunk of data, there is need to analyse the data and extract some meaning insight format.

III. LITERATURE REVIEW

A. Materials and methods

1) Data source: A non-systematic review of literature was performed using keywords such as "machine learning", "artificial intelligence", "kidney disease", "chronic kidney disease", and "deep learning".

2) Study selection: This paper is derived based upon English articles or articles that could be availabled with English abstratcs. Even, studies of human datasets were included. Moreover, references were also mentioned from bibliographies of identified articles and the authors' files. [2]

B. Related Work

AI technologies has good advantage in warning of critical illness like acute kidney injury (AKI). The mortality rate was about 30% for people having such disease without complication and around 30 – 80% for patients having different organ failure. Hence, early recognition and prevention of AKI is very important. Traditional linear models are generally over fitting and are multicollinearity. Whereas machine learning algorithms were taken into consideration for good comparable ability to predict such disease. In 2015, Google has developed the streams program that helps to predict AKI and generate alert for doctors for early intervention. Even, Tomase had developed a model that can predict approximately 55.8% inpatient episodes of AKI and 90.2% of all AKI that needed administration of dialysis by AI. [2]

With the help of Bayesian network and AI, Eiichiro had identified factors of progressive CLD from healthy population at a health check point. They had taken into consideration high blood pressure, hypertension, and some other factors. With that experiment, they came to conclusion that ANN had better prediction results with accuracy of 99.75% than SVM which was having accuracy of 97.75% respectively. [2]

In 2019, Xiao *et al.* [3] in their research, developed and compared nine machine learning models that includes LR, k-nearest neighbor, ridge regression, lasso regression, Elastic Net, SVM, RF, XGBoost and neural network to predict the progression of CKD. They came to final decision that linear model have predictive accuracy with an average AUC more than 0.87 and precision more than 0.8 respectively.

A few investigations somewhere in the range of 2008 and 2017 have exhibited that acute kidney injury is normally reversible. Few patients may encounter fragmented recuperation of kidney work, while others hence create sped up loss of kidney work, bringing about an expanded danger of chronic kidney infection. A multivariable model was created with 9973 members and was remotely approved with 2761 members to build up a down to earth hazard separation approach that could be utilized to recognize patients at high danger of CKD after they are released. This model utilizing routine research facility information had the option to foresee progressed ongoing kidney infection following hospitalization with intense kidney injury. [4]

In 2017, Dr. Akbilgic, is a data scientist with sound knowledge of statistical analysis and machine learning

developed a model which predicted the risk of the death of patients. The model showed cardiac rhythm classification from a single lead ECG script using random forest algorithm. However, random forest gave him only 0.76% accuracy. But the model was so much successful based on the efficiency that it has more than 27 thousand patents.[5]

Dr. Chen made some remarkable changes in precision of the models in nephrology. He prepared machine learning aided risk prediction model used for immunoglobulin a nephropathy (IgAN). He made some changes in new version of this model where he used standard modelling with small number of predefined variables. Moreover, he used extreme gradient boosting (XGBoost) to get the regularity of the candidate feature which increased the prediction accuracy up to 0.84%.[6]

IV. Dataset

The dataset used for this project is available University of California, Irvine (UCI) with repository named as Chronic_Kidney_Disease Dataset. Basically, it contains record of 400 patient in which 250 were recorded with CKD and 150 without CKD. There are patients from multiple age groups. Dataset has 24 features. 11 features have data type float64 whereas rest of them have object datatype.

```
age                               float64
blood pressure                    float64
specific gravity                  float64
albumin                           float64
sugar                             float64
red blood cells                    object
 pus cell                          object
pus cell clumps                    object
bacteria                           object
blood glucose random              float64
blood urea                        float64
serum creatinine                  float64
sodium                            float64
potassium                         float64
haemoglobin                       float64
packed cell volume                float64
white blood cell count            float64
red blood cell count              float64
ypertension                        object
diabetes mellitus                  object
coronary artery disease            object
appetite                           object
pedal edema                        object
anemia                             object
class                              object
dtype: object
```

Fig. 1. Features and Datatypes

V. Description of Solution and Result

Detailed description of work is divided into sub sections and explained as below:

For detailed work, libraries used are pandas, numpy, matplotlib and seaborn. Pandas library is used for data extraction and data manipulation. Numerical python (numpy) is used to perform numerical tasks and analysis. For data visualization, matplotlib library is used. For statistical graphics in python, seaborn library is used.

A. Preparing data for analysis and modelling

The very first problem statement was to perform lots of pre-processing of data. Later, that data is used for the analysis and modelling purpose. For, working in real world scenario, data is never cleaned. Hence, data cleaning, pre-processing and techniques are used to clean that data. Later from that cleaned data, meaningful pattern is searched. Once, data is understood, data machine learning model is developed depending upon available use case whether its regression, classification, segmentation, or time series use case.

Not required feature like id was dropped from the dataset as there was already one field with unique values which could be treated as primary reference. Datatypes for fields white blood cell count, blood pressure and packed cell volume was changed based upon its values which were decimal value. Hence, it was replaced to float datatype from object datatype.

B. Applying data cleaning techniques

First, columns were divided into categorical columns and numerical columns based upon its datatype. Dirtiness in categorical data is searched by checking number of unique values in each feature.

After applying that, dirtiness is found in feature diabetes mellitus, coronary artery disease and class. There is need to correct 2 features and the target variable which contain certain discrepancy in some values. So, after obtaining dirtiness, it is replaced with “yes” or “no” value in each feature because “/t” was extra in its value.

C. Analysis Distributions of data

Distribution of numerical columns is represented as num_col list. Iteration is performed on this list. After that, there is need of sub plot and on every plot, there is need to visualize a distribution of each numerical column. For plotting the subplot, parameter is used. With respect to index of every numerical column, unique sub plot is visualized. Hence enumerate function is used. Histogram is generated based upon numeric feature.

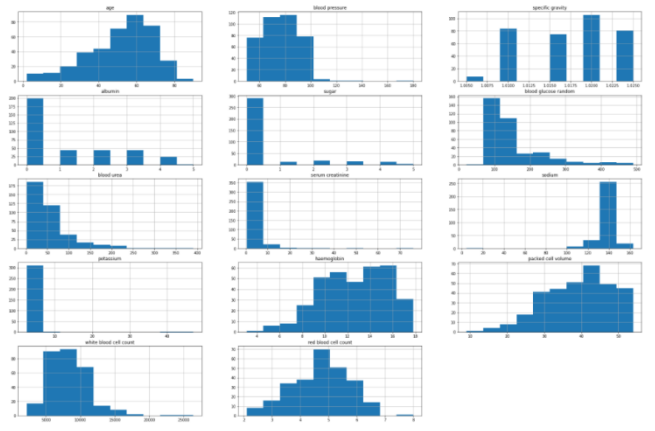

Fig. 2. Histogram of num_col

Further, observations are noted from generated graphs. Age graph (subplot – 1) looks a bit left skewed. Blood glucose random graph (subplot - 6) is right skewed. Blood Urea graph is also a bit right skewed (subplot - 7) and rest of the features are lightly skewed.

Label distribution of categorical data is done through cat_col. There are two labels in the given dataset i.e. patient with CKD and patient without CKD. There is total 11 categorical columns. With help 11 cat_col, count plot is generated.

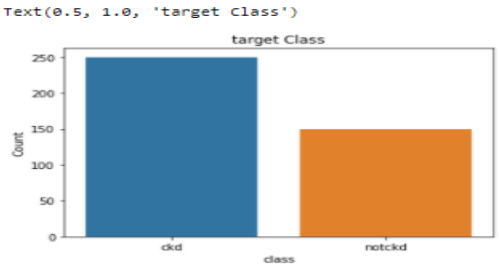

Fig. 3. Categorical Distribution

D. Checking co-relation in data

The best way to understand co-relation is using heatmap. Here, object of seaborn library helped to generate heatmap. Colour bar is present on the side of heat map. On basis of that, conclusions can be made. It indicated that specific gravity has good relationship with red blood cell count. It also indicated that as specific gravity increases, haemoglobin also increases. Sugar has positive relation with Blood glucose random. Blood Urea has it with Serum creatinine and Haemoglobin and packed cell volume has it with Red blood cell count

Even negative co-relation can be found through that heatmap Albumin, Blood Urea has negative relation with Red blood cell count, packed cell volume and Haemoglobin. Serum creatinine has it with Sodium.

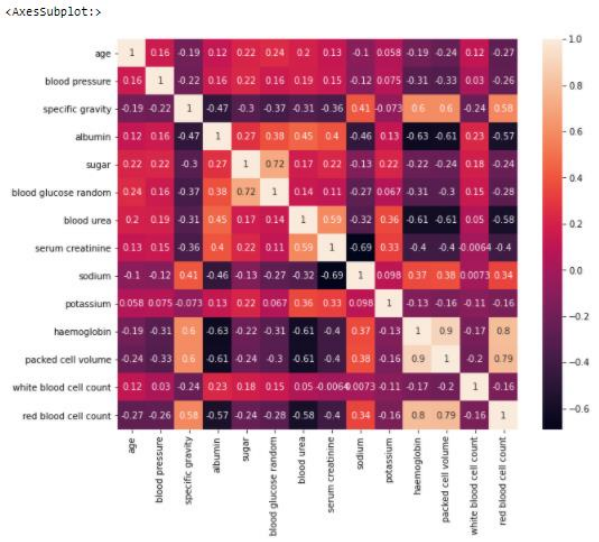

Fig. 4. Heatmap

Data is grouped on basis of red blood cell. Aggregate function is applied on it and good stats are available.

		count	mean	median	min	max
red blood cells	class					
abnormal	ckd	25	3.832000	3.7	2.5	5.6
normal	ckd	40	3.782500	3.8	2.1	8.0
	notckd	134	5.368657	5.3	4.4	6.5

Fig. 5. Stats

From stats, it can be concluded that whenever a person is not having CKD, mean of his/her red blood cell is always high. Hence, median, min and max also possesses higher values.

Violin plot is one of the good distribution plots of plotly which depends upon number of categories in X-axis. Here, red blood cell count is taken on Y-axis and class on X-axis. Colour is taken based upon class. Graph contains data regarding distribution of red blood cell whenever a person is having CKD and not having CKD.

Fig.5. Red Blood Cell Distribution for CKD/noCKD

From the graph, a conclusion can be made that maximum value for noCKD is 6.5 and minimum value of 4.4 Hence, whenever a person is not having CKD, distribution of red blood cell is not varying with respect to person having CKD.

E. Automating the analysis

Now, there is requirement find the relationship between haemoglobin and packed cell volume. Whenever concept of relationship exist, scatter plot plays a vital role. On X-axis, haemoglobin is considered and on Y-axis, packed cell volume is considered. Graph has a linear kind of trend. When haemoglobin increases, packed cell volume increases much in a linear way.

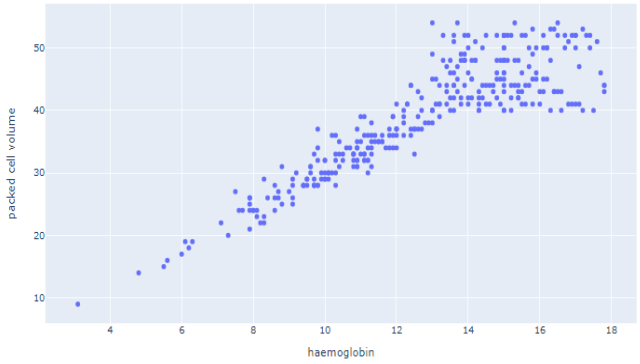

Fig.6. Relationship between Haemoglobin vs Packed cell volume

Then, analysis of distribution of red blood cell in CKD and noCKD is performed. Faceitgrid using seaborn library is used to display the analysis. Blue line indicates reed blood cell distribution among CKD and yellow line indicates among noCKD. From the figure, it can be analysed that person not having CKD have greater red blood cell distribution counts.

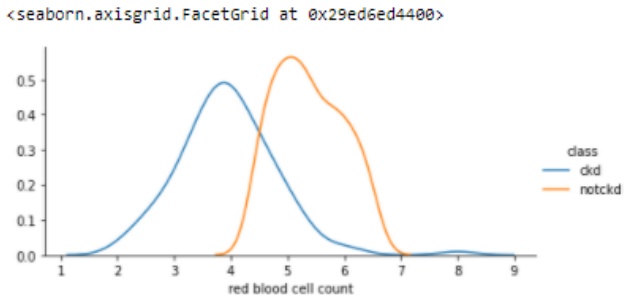

Fig.7. Analysis graph of red blood cell distribution

Similarly, analysis of distribution of haemoglobin in CKD and noCKD is performed. Similar result appears for haemoglobin too. With the help of function used for analysis, it can be used to analyse other features too. So, by changing the feature in that function, analysis of that feature can be performed without increasing space complexity.

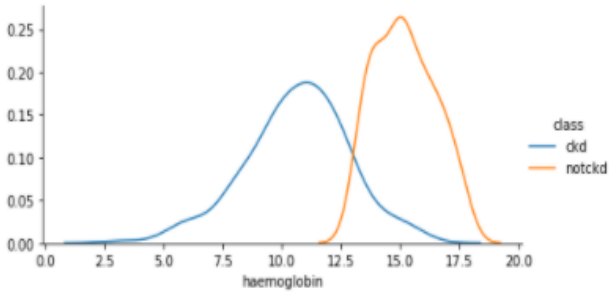

Fig.8. Analysis graph of haemoglobin distribution

F. Performing exploratory data analysis on data

Understanding relationship between red blood cell count and packed cell volume. Again, to understand relationship, scatter plot is best way.

Fig.8. Relationship between rbc count and pcv

From the scatter graph of red blood cell count vs packed cell volume, it can be analyzed that when a person is having CKD, relationship between given two features follows linear trend. Whereas that for not having CKD, it follows non – linear trend.

Similarly, to understand relationship between red blood cell count and hemoglobin, scatter graph og them is generated.

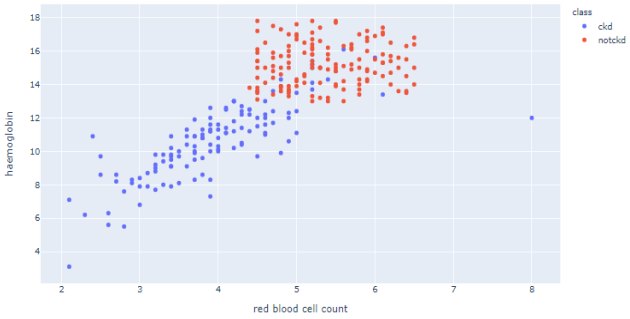

Fig.9.Relationship between hemoglobin and rbc count

Similar linear trend for CKD and non-linear trend for noCKD can be seen here too. Thus, RBC count range ~2 to <4.5 and Hemoglobin between 3 to <13 is mostly classified as positive for chronic kidney disease (i.e., ckd). RBC count range >4.5 to ~6.1 and Hemoglobin between >13 to 17.8 are classified as negative for chronic kidney disease (i.e., noCKD).

It is necessary to check for negative correlation and its impact on classes. Feature named albumin and blood urea have negative correlation with red blood cell count, packed cell volume and hemoglobin. Checking correlation of red blood cell count and albumin.

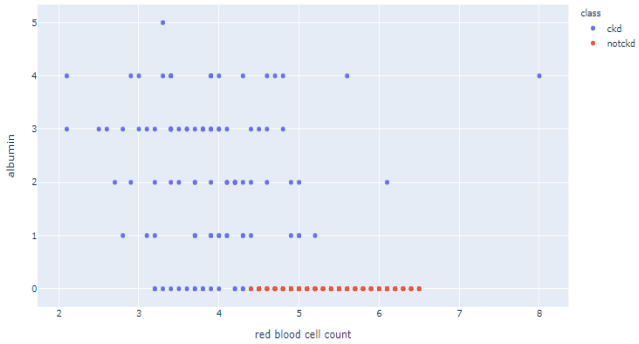

Fig. 10. Relationship between RBC count and albumin

From the scatter graph, it can be clearly analysed that albumin levels of above 0 affect CKD largely.

G. Performing Data Cleaning

The important factor after analysis is to deal with missing values in given dataset. To check whether missing values are there in data or not, “isna()” function is used. Its expression is as below which identifies all factors in descending order with total number of missing values.

dataset.isna().sum().sort_values(ascending=False)

It results all factors having missing values and factor on top contains highest missing values. Red blood cells contain highest missing values of 152 and class with 0 missing values i.e., the lowest count. Missing values can impact the model very badly.

```
red blood cells            152
red blood cell count       131
white blood cell count     106
potassium                   88
sodium                      87
packed cell volume          71
 pus cell                   65
haemoglobin                 52
sugar                       49
specific gravity            47
albumin                     46
blood glucose random        44
blood urea                  19
serum creatinine            17
blood pressure              12
age                          9
bacteria                     4
pus cell clumps              4
ypertension                  2
diabetes mellitus            2
coronary artery disease      2
appetite                     1
pedal edema                  1
anemia                       1
class                        0
dtype: int64
```

Fig. 11. Missing value count

One approach is to fill missing value with mean, median, standard deviation. But if there is huge number of missing values, use of mean will impact distribution of data badly. Normal distribution of data is most suitable for machine learning model. Hence, distribution of data must be maintained.

Another approach is to replace missing value with some random values. But it should be done with proper care. Copy of dataset is made and some sample feature like red blood cells is taken and random value from copied dataset is filled. Generally, it provides better results. For an instance, 152 random values are taken to replace missing values of red blood cells feature. For this approach, indexes must be equal. Data points get changed every time when random function is called.

```
362      normal
196    abnormal
61       normal
114    abnormal
158      normal
         ...
395      normal
387      normal
133      normal
293      normal
308      normal
Name: red blood cells, Length: 152, dtype: object
```

Fig: 12. Missing values Indexes

```
Int64Index([  0,   1,   5,   6,  10,  12,  13,  15,  16,  17,
            ...
            245, 268, 280, 290, 295, 309, 322, 349, 350, 381],
           dtype='int64', length=152)
```

Fig. 13. Random Value Indexes

With the help of “isnull()” function, indexes having missing values will appear. When random sample indexes are put in missing value, total number of indexes must be same. After filling missing values, it will generate answer 0 for number of missing values. The main reason for this process is to clean missing values without affecting the ratio. Once, number of missing values decreases, any approach like mean, median or standard deviation.

After cleaning data in features, there is need to clean categorical and numerical features. After cleaning, there is need to fix it. Making function of the process mentioned in previous section will fill all missing values in all features. Ultimately, it will optimize the code and reduce the complexity.

H. Applying feature encoding on data

The need for applying feature encoding on data raised due to data types of features. Machine learning can not understand descriptive data. Hence, descriptive data needed to be updated to numerical data. Unique labels are searched in every feature. Whenever there is a smaller number of unique labels, label encoding technique can be applied because it will not cause curse of dimensionality. For an instance, if there are two categories named with normal and abnormal, label encoding converts it to 0 and 1. If there are more categories, further consecutive numbers (2, 3, 4, …) are selected.

```
red blood cells has 2 categories
 pus cell has 2 categories
pus cell clumps has 2 categories
bacteria has 2 categories
ypertension has 2 categories
diabetes mellitus has 2 categories
coronary artery disease has 2 categories
appetite has 2 categories
pedal edema has 2 categories
anemia has 2 categories
class has 2 categories
```

Fig. 14. Unique labels

Label encoder class is imported using scikit learn library. After applying it, all categorical data is converted into numerical data and the developed machine learning model can understand the available numerical data. In case of higher unique labels, label distribution and its contribution are analyzed. Depending upon that contribution, some threshold value is generated. Based upon obtained threshold value, some categorical features are eliminated.

I. Selecting best feature for model using suitable feature importance technique

With the help of feature selection from scikit learn library, SelectKbest is imported. Chi square class is imported which checks whether probability value is less than 0.5 or not. Based upon that probability value, it will order all features needed for model building.

SelectKBest feature using chi2 as score function and fit function, it will generate ordered feature module. To generate rank of feature, score function is used.

J. Building a Cross-Validated Model and checking its accuracy

For building a Cross-Validation model and generating classification matrix, training and testing data is required. With the help of scikit learn library, training and testing dataset is generated. Random state is taken as 0 and the test size is considered 0.25.

After checking testing data for X, it resulted count 100. So, prediction is done on 100 entries. Obtained trained model is not have much imbalance condition and it can be used for machine learning model. XGBClassifier is imported from XGBoost algorithm to check which is best suitable model from with the help of obtained parameter. Hence, model needs to be cross validated.

Hyperparameters are used for the model. Hyperparameters plays vital role for obtaining better accuracy rate. So the need is for the hyper-parameters which will be used by the RandomizedSearchCV, which is a model provided by scikit

learn. The return value of XGBoost will be used by RandomizedSearchCV and on top of that we need to define some custom parameters like, learning_rate, max_depth, min_child_weight, gamma values and colsample_bytree. For good efficiency we will use 5 iterations. Classifier prediction of X-test dataset is as follow:

```
array([0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1], dtype=int64)
```

Fig.13. X-Test Classifier Prediction

VI. Discussion of Results

Using this model of random search and using X and Y values it will show a list of values which are a good fit for such types of dataset. In this case when random_search.best_estimator_ is used, it gives best parameters for our XGBoost classifier. This will increase the efficiency to a greater extent. XGBoost classifier is ready to fit and predict the values that is X and Y. Ultimately, using predict method, the accuracy obtained by this method will be **0.97** which is very dependable. Obtained confusion matrix is:

```
[[58  3]
 [0  39]]
```

VII. Future Work

- To increase the prediction and accuracy for the model.
- Working with features having higher unique labels
- Reducing the dataset without affecting the accuracy

VIII. Task Performed by Group Participants

Tasks performed by Nikhil:

- Checking correlation in data
- Automating the analysis
- Performing exploratory data analysis
- Predicting the accuracy of the test set
- Feature distribution for CKD, NO-CKD
- Generating heatmaps and dealing with the missing values indexes

Tasks performed by Yash:

- Splitting the train and test set.
- Performing data cleaning
- Scaling the dataset
- Applying feature encoding on data
- Generating the cross-validation model and generating classification matrix
- Generating histogram of num_col
- Generating scattering graphs of relationship between features.
- Generating label encoder for smaller unit labels.

Tasks performed by Both:

- Searching open available dataset (Chronic kidney disease dataset).
- Feature selection
- Documentation work

IX. Conclusion

After performing data cleaning and scaling techniques on raw data, feature encoding was successfully performed. Thus, performing cross validation techniques using XGBoost, it resulted the accuracy score of 0.97. Hence this model is a good fit for predicting the chronic kidney disease in humans.

Acknowledgment *(Heading 5)*

We are thankful to Dr. Luis Rueda for guiding us throughout the semester.

References

There are six authors.

[1] "Chronic kidney disease," *Mayo Clinic*, 15-Aug-2019. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/chronic-kidney-disease/symptoms-causes/syc-20354521. [Accessed: 21-Apr-2021].

[2] Q. Yuan, H. Zhang, T. Deng, S. Tang, X. Yuan, W. Tang, Y. Xie, H. Ge, X. Wang, Q. Zhou, and X. Xiao, "Role of Artificial Intelligence in Kidney Disease", International Journal of Medical Sciences, 2020.

[3] M. Almasoud, T. Ward, "Detection of Chronic Kidney Disease Using Machine Learning Algorithms with Least Number of Predictors", 2013.

[4] J. Xiao et al, "Comparison and development of machine learning tools in the prediction of chronic kidney disease progression," Journal of Translational Medicine, vol. 17, (1), pp. 119, 2019. R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5] S. Chaudhuri, A. Long, H. Zhang, C. Monaghan, J. W. Larkin, P. Kotanko, S. Kalaskar, J. P. Kooman, F. M. van der Sande, F. W. Maddux, and L. A. Usvyat, "Artificial intelligence enabled applications in kidney disease," *Wiley Online Library*, 13-Sep-2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1111/sdi.12915. [Accessed: 22-Apr-2021].

[6] G. Xie, T. Chen, Y. Li, T. Chen, X. Li, and Z. Liu, "Artificial Intelligence in Nephrology: How Can Artificial Intelligence Augment Nephrologists' Intelligence?," *Kidney Diseases*, 03-Dec-2019. [Online]. Available: https://www.karger.com/Article/FullText/504600. [Accessed: 22-Apr-2021].