PROJECT: Medical prediction for patients of kidney disease

I. Introduction

1. Motivation and problem definition

Almost a large number of the population worldwide is affected with a major health problem, chronic kidney disease. As a result, early detection and characterization are considered to be critical factors in the management and control of this long-lasting kidney disease. These tasks have been traditionally performed by well-trained healthcare professionals; however, they are still some of the most challenging work due to the subtle signs and difficult to detect symptoms hidden in data set. Herein, use of well-organized data mining techniques is shown to expose hidden information from clinical and laboratory patient data, which can be helpful to assist physicians in maximizing accuracy for identification of disease severity stage. The works in the past without the use of the machine learning algorithms fail to provide the accuracy of prediction to the needed extent. So, our project will try to indicate that applying different machine learning algorithms, provide better classification and prediction performance for determining whether one patient has chronic kidney disease. The project will try to predict the chronic kidney disease (ckd) of patients using systematic and automatic methodologies. Among the methodologies, the machine learning algorithm and feature selection are some of the very kinds.

2. Challenge

The identification of ckd from the patients' data can be treated as a data classification issue. The classification task is generally a supervised learning process that doesn't give much insight about the connection between features and class labels. At the beginning, with 24 medical attributes data set, our challenge was to find an approach to make the prediction easier while assuring the high accuracy of our machine learning model. Also, in our project, we need to find a way to provide the interpretability of our machine learning model.

II. Proposed solution

To give a solution for the challenges of medical prediction for kidney disease patients, we will work though 3 consecutive phases:

1. Choosing and preprocessing dataset

We choose to use the Chronic_Kidney_Disease Dataset because it's openly available online for public access on the Machine Learning Repository website[*]. The data was retrieved from an Indian hospital over approximately a 2-month period. It contains 24 medical attributes and 400 records. The first 24 columns are the risk factors of chronic kidney disease while the 25th column is the classification of the disease (ckd or notckd). There are 11 numeric columns and 14 nominal columns. This dataset has rich resourceful information and each record is the statistic about the medical situation of an individual. However, this dataset contains a considerable amount of missing values and mistyped characters, a preprocessing data phase needs to be implemented to overcome this drawback.

2. Features selection

2.1. Motivation

Because the original dataset has 24 attributes and not all of them give relevant information about the medical situation of an individual. If we keep the irrelevant attributes, they may cause some noise to the predicted results. Also, because the considered attributes come from the patients, which reflect their health situation, there must be some correlations between some attributes. Keeping all the attributes will lead to the existence of redundant information. Another important reason to have the feature selection comes from the interpretability property of machine learning models. Medical experts need to have a clear view about the strong predictors as the medical metrics so that they can analyze and explain the situation to the patients. As such, this phase is important.

In medical diagnosis, it is very important to identify most significant risk factors related to disease. Relevant feature identification helps in the removal of unnecessary, redundant attributes from the disease dataset which, in turn, gives quick and better results. So, in our project we used feature selection, also known as Variable Selection, as an extensively used data preprocessing technique in data mining which we basically used for reduction of data by eliminating insignificant and superfluous attributes from our ckd dataset. Moreover, this technique enhances the comprehensibility of the data, facilitates better visualization of the data, reduces training time of learning algorithms and improves the performance of prediction.

In some related works, such as the research presented in (11) has considered 5 attributes: blood pressure, serum creatinine, packed cell volume, hyper- tension, and anemia to calculate the L-factor and clustered Chronic Kidney Disease (ckd) and non-ckd patients based on the L-factor value. According to their evaluation ckd cannot be detected based on their L-factor classifiers. Other works (5), (14) have evaluated machine learning algorithms such as back propagation neural networks, radial basis functions, random forests and SVMs and achieved up to 85.3% accuracy on identifying ckd. Also, (17) performs feature selection techniques such as information gain, gain ratio, or attribute evaluation and fusion based feature selection to identify relevant features, but their evaluation has not presented the relevant selected features. Hence, in our project, we tried to use another approach.

2.2. Methods

At the beginning our team did want to use PCA to reduce dimensionality of our project's dataset, but the disadvantage of PCA become more evident when we consider about the interpretability property of our prediction model. PCA transforms the original features into new space with less dimensionality, but as such, this technique can't give us some kind of insight about the importance of each original medical feature. That's why we decide to use only feature selection techniques for better interpretability.

There exist numerous applications of relevant feature identification techniques in the healthcare sector. We use the following methods:

1. **Univariate Feature**: Visiting every feature and checking its importance with the target.

2. **Wrapper method - Recursive Feature Elimination- RFE:** Recursively removing attributes and building a model on those attributes that remain. An external estimator is used to assign weights to features to identify which attributes contribute the most to predicting the target attribute.

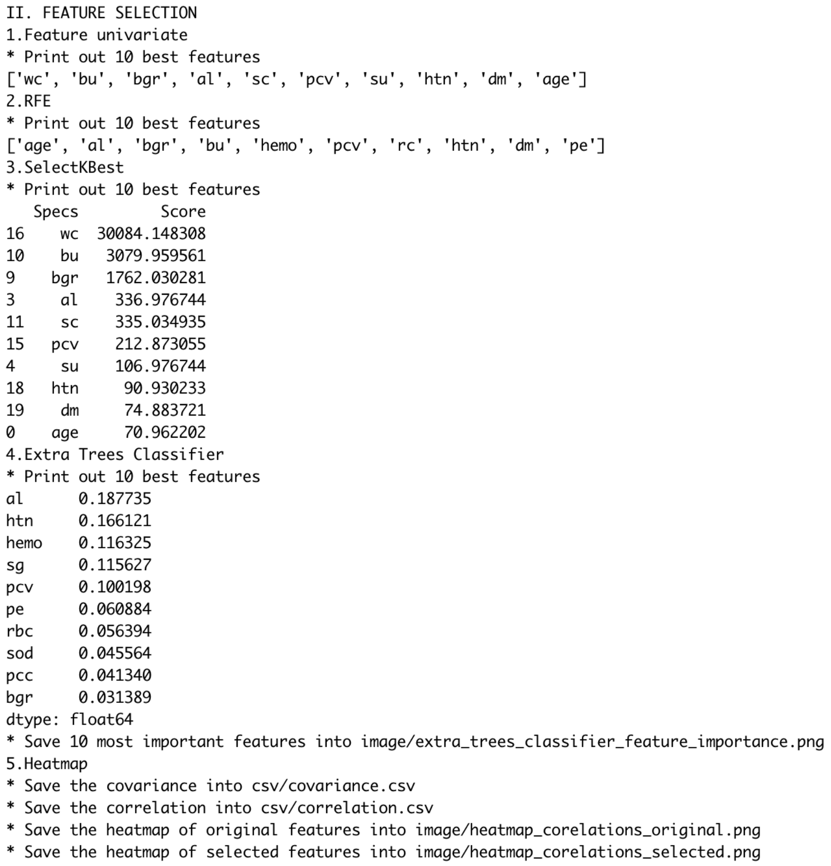
3. **Embedded method - Select K-best:** The SelectKBest class just scores the features using a function and then removes all but the k highest scoring features

4. **Extra Trees Classifier:** ensemble learning technique to aggregate the results of multiple de-correlated decision trees collected in a "forest" and give the classification result.

5. **Heatmap:** This makes it easy to identify which features are most related to the target variable by visualize the Correlation Matrix

At the beginning, because we have 24 features so we will use 5 methods above to select 10 best features for each method. We argue that, if an attribute is a strong predictor, it should appear in the result of various Feature Selection techniques, therefore by doing experiments with various aforementioned algorithms, we will detect which attributes appear frequently. If all the 10 best features of 5 methods are the same, that means all the 10 features are strong predictors. In the case, we will then increase the number of best features for each method and compare among the result until we find the best number of selected strong predictors. In the opposition case, if we compare 10 best features of each method and we find out the number of selected common strong predictors is less than 10, we can immediately select them.

2.3. Result



```
II. FEATURE SELECTION
1.Feature univariate
* Print out 10 best features
['wc', 'bu', 'bgr', 'al', 'sc', 'pcv', 'su', 'htn', 'dm', 'age']
2.RFE
* Print out 10 best features
['age', 'al', 'bgr', 'bu', 'hemo', 'pcv', 'rc', 'htn', 'dm', 'pe']
3.SelectKBest
* Print out 10 best features
   Specs         Score
16    wc  30084.148308
10    bu   3079.959561
9    bgr   1762.030281
3     al    336.976744
11    sc    335.034935
15   pcv    212.873055
4     su    106.976744
18   htn     90.930233
19    dm     74.883721
0    age     70.962202
4.Extra Trees Classifier
* Print out 10 best features
al      0.187735
htn     0.166121
hemo    0.116325
sg      0.115627
pcv     0.100198
pe      0.060884
rbc     0.056394
sod     0.045564
pcc     0.041340
bgr     0.031389
dtype: float64
* Save 10 most important features into image/extra_trees_classifier_feature_importance.png
5.Heatmap
* Save the covariance into csv/covariance.csv
* Save the correlation into csv/correlation.csv
* Save the heatmap of original features into image/heatmap_corelations_original.png
* Save the heatmap of selected features into image/heatmap_corelations_selected.png
```

For each method, we find out the 10 best features as following:

1. **Univariate Feature** : Set1 = ['wc', 'bu', 'bgr', **'al'**, 'sc', 'pcv', 'su', **'htn'**, **'dm'**, 'age']

2. **RFE** : Set2 = ['age', **'al'**, 'bgr', 'bu', 'hemo', 'pcv', 'rc', **'htn'**, **'dm'**, 'pe']

3. **Select K-best** : Set3 = ['wc', 'bu', 'bgr', **'al'**, 'sc', 'pcv', 'su', **'htn'**, **'dm'**, 'age']

4. **Extra Trees Classifier**: Set4 = [**'dm'**, 'sc', 'hemo', 'pc', 'pe', 'appet', 'rbc', **'al'**, 'sod', **'htn'**]

5. **Heatmap:** Look at the heatmap of original features (saved in *image/heatmap_correlations_original.png*), we select the 10 features which has the highest correlation with the target label "classification"
   Set5 **=** [**'al'**, **'htn'**, **'dm'**, 'sc', 'bu', 'pe', 'bgr', 'ane', 'pcc', 'su']

Therefore, at the end we have the strong features is:

$Strong$ features = $set1 \cap set2 \cap set\ 3 \cap set\ 4 \cap set\ 5$= [**'htn', 'dm', 'al'**]

The heatmap for the selected features (saved in file *image/heatmap_corelations_selected.png*).

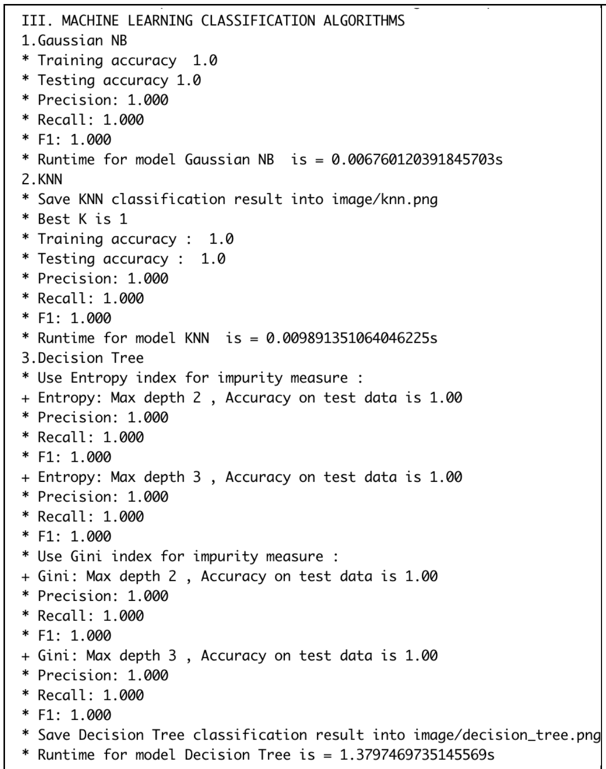
With this result of feature selection, we will apply various classification technique in machine learning to see if they can predict well the kidney disease patients. If it gives us good result, that will be very helpful to the doctors because they will need only 3 medical attributes for the interpretability of the prediction.

3. Apply various machine learning models with trial-and-test strategy

3.1. Methods

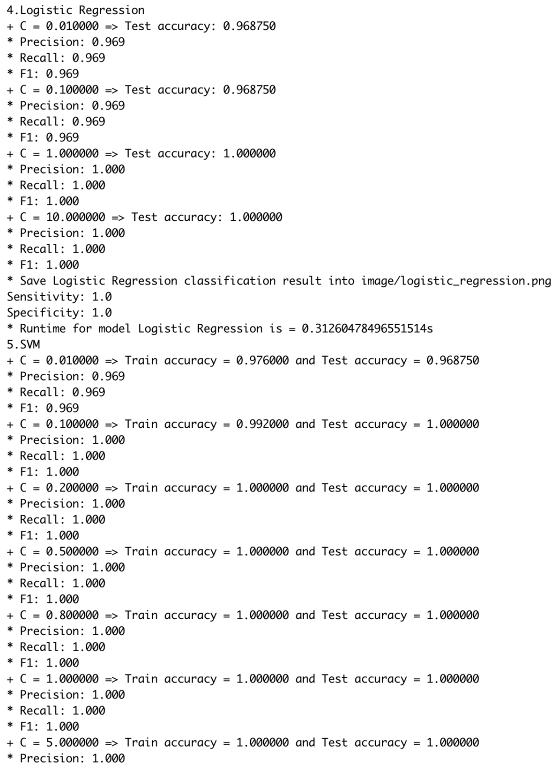
We use 9 different machine learning classification methods to make prediction of kidney disease patients and observe the result. We use Gaussian NB, K Nearest Neighbor, Decision Tree, Logistic Regression, SVM (linear and non-linear approach), Random Forest, Bagging, AdaBoost. For each method, to observe the performance, we print out the training accuracy, testing accuracy, precision, recall and F1 score. The plots are used also to visualize the result and they are saved in folder */image* with file names as indicated in the printed screen.

3.2. Result and analysis



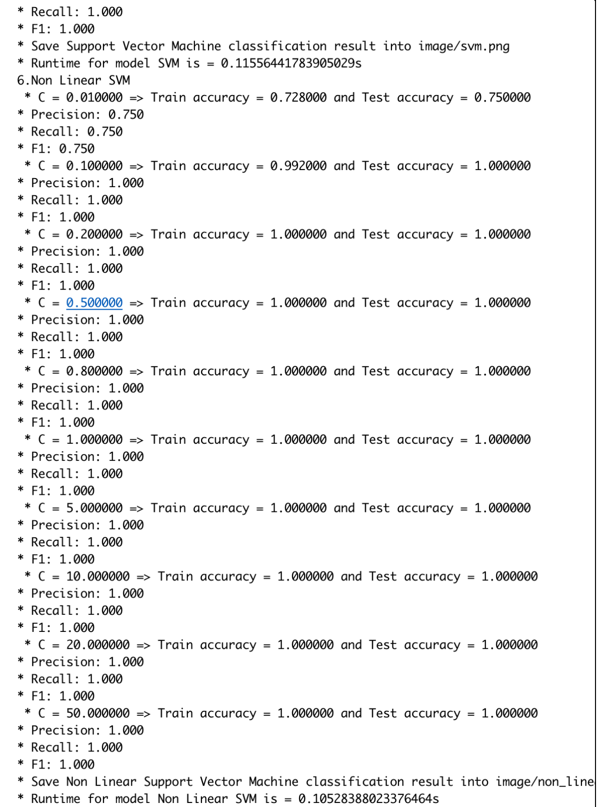
```
III. MACHINE LEARNING CLASSIFICATION ALGORITHMS
1.Gaussian NB
* Training accuracy  1.0
* Testing accuracy 1.0
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Runtime for model Gaussian NB  is = 0.006760120391845703s
2.KNN
* Save KNN classification result into image/knn.png
* Best K is 1
* Training accuracy :  1.0
* Testing accuracy :  1.0
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Runtime for model KNN  is = 0.009891351064046225s
3.Decision Tree
* Use Entropy index for impurity measure :
+ Entropy: Max depth 2 , Accuracy on test data is 1.00
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ Entropy: Max depth 3 , Accuracy on test data is 1.00
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Use Gini index for impurity measure :
+ Gini: Max depth 2 , Accuracy on test data is 1.00
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ Gini: Max depth 3 , Accuracy on test data is 1.00
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Save Decision Tree classification result into image/decision_tree.png
* Runtime for model Decision Tree is = 1.3797469735145569s
```

Figure 1



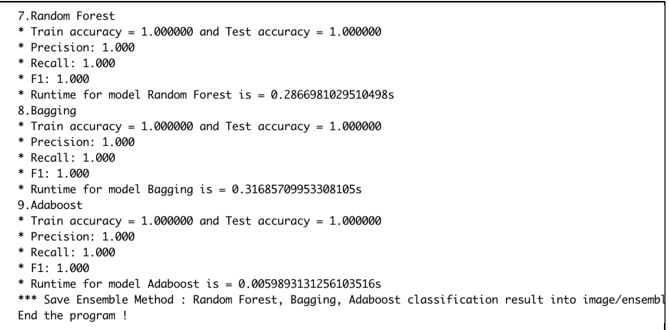
```
4.Logistic Regression
+ C = 0.010000 => Test accuracy: 0.968750
* Precision: 0.969
* Recall: 0.969
* F1: 0.969
+ C = 0.100000 => Test accuracy: 0.968750
* Precision: 0.969
* Recall: 0.969
* F1: 0.969
+ C = 1.000000 => Test accuracy: 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ C = 10.000000 => Test accuracy: 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Save Logistic Regression classification result into image/logistic_regression.png
Sensitivity: 1.0
Specificity: 1.0
* Runtime for model Logistic Regression is = 0.31260478496551514s
5.SVM
+ C = 0.010000 => Train accuracy = 0.976000 and Test accuracy = 0.968750
* Precision: 0.969
* Recall: 0.969
* F1: 0.969
+ C = 0.100000 => Train accuracy = 0.992000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ C = 0.200000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ C = 0.500000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ C = 0.800000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ C = 1.000000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
+ C = 5.000000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
```

Figure 2



```
 * Recall: 1.000
 * F1: 1.000
 * Save Support Vector Machine classification result into image/svm.png
 * Runtime for model SVM is = 0.11556441783905029s
 6.Non Linear SVM
  * C = 0.010000 => Train accuracy = 0.728000 and Test accuracy = 0.750000
 * Precision: 0.750
 * Recall: 0.750
 * F1: 0.750
  * C = 0.100000 => Train accuracy = 0.992000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 0.200000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 0.500000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 0.800000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 1.000000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 5.000000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 10.000000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 20.000000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
  * C = 50.000000 => Train accuracy = 1.000000 and Test accuracy = 1.000000
 * Precision: 1.000
 * Recall: 1.000
 * F1: 1.000
 * Save Non Linear Support Vector Machine classification result into image/non_line
 * Runtime for model Non Linear SVM is = 0.10528388023376464s
```

Figure 3



```
7.Random Forest
* Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Runtime for model Random Forest is = 0.2866981029510498s
8.Bagging
* Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Runtime for model Bagging is = 0.31685709953308105s
9.Adaboost
* Train accuracy = 1.000000 and Test accuracy = 1.000000
* Precision: 1.000
* Recall: 1.000
* F1: 1.000
* Runtime for model Adaboost is = 0.0059893131256103516s
*** Save Ensemble Method : Random Forest, Bagging, Adaboost classification result into image/ensembl
End the program !
```

Figure 4

In the 4 above figures, we observe that with 3 selected best features, all 9 models of prediction give us very good result. We have test accuracy is 1.0, precision is 1.0, recall is 1.0 and F1 score is 1.0 also. This very good result is explained by the fact that the medical experts who has the knowledge about kidney disease has collected the 24 meaningful medical attributes from the beginning. Also, this very good result confirms that our approach in choosing the best features is successful in reducing the number of must-collected features from 24 to 3. This result is very helpful in medicine because both potential patients and the medical doctors will save time and money in doing medical tests in regard to collecting necessary information for kidney disease prediction. Among the 9 models that we use, although all give us very good performance, Decision Tree model will help us to have the best interpretability because with only 3 features, It's very simple to understand how to make the decision for prediction. (Look the decision schemas in *image/tree_entropy_x.png; image/tree_gini_x.png*)

III. Conclusion

The diagnosis of the Chronic Kidney Disease (ckd) is a cumbersome problem. Researches have reported that ckd results into the tens of thousands human death toll. It is therefore of great interest to find out ways to diagnose the ckd in easier and less expensive way. At the beginning, with 24 medical attributes data set, our challenge was to find an approach to make the prediction easier while assuring the high accuracy of our machine learning model. Also, in our project, we need to find a way to provide the interpretability of our model. The result shows that with the methodology illustrated in feature selection section, we succeed in reducing the number of considered features from 24 to only 3 and still keep the high accuracy and precision of our model. The ckd prediction results of experiments show very good performance for Albumin (al), Hypertension (htn), and Diabetes Mellitus (dm) on an average using different evaluation metrics. With only 3 selected features instead of 24, the medical doctors and patients can both save time and money for only the necessary medical tests. This project work is going to be vital for benefitting the well-being of people and identifying diseases at early stages. Also, Decision Tree model gives us the best interpretability among experimented models. In the future more compact and autonomous tools can be developed.

[*] Machine Learning Repository - Center for Machine Learning and Intelligent Systems.

Retrieved from

https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease?fbclid=IwAR2bJXrFFo9VK

References

1. Z. Masetic and A. Subasi, "Congestive heart failure detection using random forest classifier," *Computer Methods and Programs in Biomedicine,* vol. 130, pp. 54-64, 2016. Google Scholar
2. E. Alickovic, J. Kevric and A. Subasi, "Performance evaluation of EMD, DWT, and WPD for automated epileptic seizure detection and prediction," *Submitted to Biomedical Signal Processing and Control,* 2016. Google Scholar
3. M. J. Pérez-Sáeza, . D. Prieto-Alhambra, C. Barrios, M. Crespo, D. Redondo, X. Nogués, A. Díez-Pérez and J. Pascual, "Increased hip fracture and mortality in chronic kidney disease individuals: The importance of competing risks," *Bone,* vol. 73, p. 154–159, 2015. Google Scholar
4. M. Cueto-Manzano, L. Cortes-Sanabria, H. R. Martinez-Ramirez, E. Rojas-Campos, B. Gomez-Navarro and M. Castillero-Manzano, "Prevalence of Chronic Kidney Disease in an Adult Population," *Archives of Medical Research,* vol. 45, pp. 507-513, 2014. Google Scholar
5. P. Sinha and P. Sinha, "Comparative study of chronic kidney disease prediction using knn and svm," *International Journal of Engineering Research and Technology*, vol. 4, no. 12, 2015.
6. Levin and P. E. Stevens, "Summary of KDIGO 2012 CKD Guideline: behind the scenes, need for guidance, and a framework for moving forward," *Kidney International,* vol. 85, no. 1, p. 49–61, 2014. Google Scholar
7. Z. Chen, Z. Zhang, R. Zhu, Y. Xiang and P. B. Harrington, "Diagnosis of patients with chronic kidney disease by using two fuzzy classifiers," *Chemometrics and Intelligent Laboratory Systems,* vol. 153, p. 140–145, 2016. Google Scholar
8. P. Muthukumara and G. S. S. Krishnan, "A similarity measure of intuitionistic fuzzy soft sets and its applicationin medical diagnosis," *Applied Soft Computing,* vol. 41, p. 148–156, 2016. Google Scholar
9. B. Widrow, D. E. Rumelhart and M. A. Lehr, "Neural networks: applications in industry, business and science," Communications of the ACM, vol. 12, pp. 93-105, 1994. Google Scholar
10. S. Armin, Data Mining and Knowledge Discovery Handbook, 2nd ed., O. Maimon and L. Rokach, Eds., New York: Springer, 2010.Google Scholar
11. A. Dubey, "A classification of ckd cases using multivariate k-means clustering." International Journal of Scientific and Research Publica- tions (IJSRP), vol. 5, August 2015.
12. L. Rokach and O. Maimon, Data Mining and Knowledge Discovery Handbook, 2nd ed., M. Oded and R. Lior, Eds., New York: Springer, 2010. Google Scholar
13. L. Breiman, J. Friedman, R. Olshen and C. Stone, Classification and Regression Trees, Wadsworth Int. Group, 1984. Google Scholar
14. J. R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, pp. 81-106, 1986. Google Scholar
15. J. R. Quinlan, C4.5: Program for Machine Learning, CA, Morgan Kaufman Publishing, 1993. Google Scholar
16. L. Breiman, "Random Forests," Machine Learning, vol. 45, p. 5–32, 2001. Google Scholar
17. E. Alickovic and A. Subasi, "Breast cancer diagnosis using GA feature selection and Rotation Forest," Neural Computing and Applications, pp. 1-11, 2015. Google Scholar