Exploratory and Predictive Analysis of Chronic Kidney Disease

[1]Srushti Wadekar
Computer Science
Lakehead University
Thunder Bay, Canada
swadekar@lakeheadu.ca

[2]Komal Barge
Computer Science
Lakehead University
Thunder Bay, Canada
bargek@lakeheadu.ca

[3]Nidhi Patel
Computer Science
Lakehead University
Thunder Bay, Canada
pateln@lakeheadu.ca

***Abstract*—Data analytics and machine learning in healthcare systems are rapidly expanding for improving patients care, chronic disease management, hospital administration and supply chain efficiencies. This paper solely gives you the details about to know the occurrence of Chronic Kidney Disease (CKD) at early stages in advance by studying the different traits of the patient. Chronic kidney disease (CKD) is a condition characterized by a gradual loss of kidney function over time. It damages your kidneys and decrease their ability to keep you healthy. CKD symptoms are often nonspecific, meaning they can also be caused by other illnesses. So, early detection of such disease would be certainly beneficial. We have done the prediction analysis using different algorithms which gave us quite good results.**

***Index Terms*—CKD, prediction, analysis, machine learning**

I. Introduction

Data Analytics plays an important role in decision making in different fields, as it provides insights from large datasets with multiple disciplines. Healthcare predictive analytics uses historical data to forecast the future, personalizing care to each patient. The past medical history, demographic information, and behaviors of a person can be used together with the expertise and experience of healthcare professionals to predict the future.

Chronic kidney disease (CKD) is a global health crisis with a high economic cost to health systems and is an independent risk factor for cardiovascular disease (CVD). All stages of CKD are associated with increased risks of cardiovascular morbidity, premature mortality, and/or decreased quality of life. Chronic kidney disease also known as chronic kidney failure describes the progressive loss of function of the kidneys. Your kidneys filter waste from your blood and excess fluids, which are then excreted into your urine. When chronic kidney disease reaches an advanced stage, dangerous fluid, electrolyte, and waste levels can build up within your body. In early stages of CKD, you may have only a few signs or symptoms. CKD cannot become evident until it seriously impairs the kidney function. Chronic kidney disease treatment focuses on slowing the progression of damage to the kidneys, usually by controlling the underlying cause. CKD can progress to fatal end-stage kidney failure if not treated with artificial filtration (dialysis) or kidney transplant.

Chronic kidney disease may be caused by diabetes, high blood pressure and other disorders. Early detection and treatment can often keep chronic kidney disease from getting worse. CKD data is used for early prediction of causes of CKD and different health conditions that might have led to CKD in patients. This will be useful to gain more insights into what caused these diseases and how we can prevent it from getting worse.

II. RELATED RESEARCH WORK

In the field of healthcare informatics, latest findings and research are continuously going on. J. Sarada and Dr. N. V. Muthu Lakshmi have done the analysis on the dataset we have used for predicting chronic kidney disease [1]. They have used classification technique. For classification, C4.5 algorithm is used to generate decision tree and classify the disease. The implementation was done in R and Weka. They have splitted data into 50% for getting training and testing set. By using implemented models, the authors were able to get higher accuracy and lower error rate.

Another paper is published by I.A. Pasadana, D. Hartama, M. Zarlis, A.S. Sianipar, A. Munandar, S. Baeha and A.R.M. Alam in which they have comparative analysis of different decision tree techniques is done [2]. The decision tree algorithm used by them are DecisionStump, HoeffdingTree, J48, CTC, J48graft, LMT, NBTree, RandomForest, RandomTree, REPTree, and SimpleCart. 10-fold cross validation was also performed. To compare the performance of these algorithms seven performance metrics were used such FACC, MAE, PRE, REC, FME, Kappa Statistics and Runtime. As per their analysis, Random Forest performed best among all the algorithms.

Basma Boukenze, Hajar Mousannif and Abdelkrim Haqiq did their research on the same dataset in which comparison of C4.5, SVM and NB learning algorithms is done [3]. The authors have also used 10-fold cross validation. According to the analysis of experimented algorithms, C4.5 came up as the best algorithm with accuracy of 63%.

Abhishek et.al, Gour Sundar Mitra Thakur and Dolly Gupta did their analysis on the different dataset related to kidney disease by comparing Back Propagation Algorithm (BPA), Radial Basis Function (RBF), and Support Vector Machine (SVM) models [4]. They have done the implementation of their models in Weka. The split ratio for train-test on their data was 75:25. They have measured the performance of model based on Accuracy, mean absolute error (MAE) and Root-mean-square error (RMSE). According to the performance of the models developed by them, the best performed model is back propagation model with accuracy of 81%.

Parul Sinha and Poonam Sinha have published their paper on comparison of KNN and SVM on the same dataset [5]. For the evaluation of results of the models, accuracy, precision and F1-measure are used. Their analysis is performed in MATLAB. Without any scaling or class balance on data, the KNN performed best with 78% of accuracy.

III. METHODOLOGY

In this paper, we have described the comparative analysis of K Nearest Neighbors, Decision Tree, Random Forest, Support Vector Classifier and Neural Network methods as classifiers. We will classify patients into two categories which are chronic kidney disease (ckd) and not chronic kidney disease (notckd).

A. Data Preprocessing

For this dataset, target variable is a 'classification' column in which we have some values of "ckd" class with tab character so first it has been replaced with "ckd". We have replaced null values with the median of the columns because our dataset has only 400 rows so dropping rows with null values is certainly not a solution. Because of the smaller number of total instances, the dataset has been split into 75:25 ratio.

The dataset has 250 sample from 'ckd' class and 150 samples from 'notckd' class so to overcome the issue of class imbalance, resampling has been performed. Resampling consists of drawing repeated samples from original samples of the data. For resampling, we have used Bootstrap method which does random sampling with replacement. Bootstrap method is one of the resampling techniques that we can use to control and check stability of the results. We have also tried Synthetic Minority Over-sampling Technique (SMOTE) to balance both the classes. SMOTE adds new synthetic instances in the dataset which are similar instances with a different variety that resides in data. Both the methods are different from each other as after Bootstrap, we can have same samples that appear more than once in data whereas SMOTE will provide us replicated instances that is not repeated. It has been observed that the models were performing similar after using SMOTE and Bootstrap. After class balance, each class was having 186 samples.

Feature scaling is done on the data to normalize the range of features of data. In our dataset, we have some features that has values with some difference. For example, in column 'bgr' we have minimum value is 22 and maximum value is 490. The varying values will affect the overall performance. Thus, we have done feature scaling on our dataset. Then, all the models are fitted on training set and tested on test set. For, performance measure of these models, we have used Accuracy, F-1 score, Precision, Recall and confusion metrics. The results of these models are discussed in the results section of the paper.

B. Neural Network Model

Neural Network can also be called as Artificial Neural Network (ANN). Neural Networks is a network that is designed to recognize hidden patterns. With the help of neural network, we can easily classify the data. Neural Network consists neurons which are aggregated into layers. In neural network, we can have multiple hidden layers between the input layer and the output layer. In this network, each neuron is connected via edges with neurons of another layer which is known as edges. Here, neurons and edges have parameter named as weight that changes as we start training the network.

For our dataset, we have implemented the Neural Network because it recognizes hidden pattern in data and can efficiently predict the target which gave us comparatively better results. The input layer has 64 units, two hidden layers has 32 and 16 units and finally the output layer is with 2 units and a softmax layer as an activation function. For this model, 'adam' optimizer is used. To train the model, the batch size of 128 and 100 epochs used. ANN has provided better results which is discussed in results section.

C. KNN Model

KNN (K-Nearest Neighbors) is one of many (supervised learning) algorithms used in data mining and machine learning, it's a classifier algorithm where the learning is based "how similar" is a data (a vector) from other. It is called Lazy algorithm because it does not need any training data points for model generation. All training data is used in the testing phase which makes training faster and testing phase slower and costlier.

The authors have used KNN as discussed in the section of literature review, but they have not done scaling or resampling on the data [5]. We wanted to do some experiment on the data by doing scaling and class balance which gave us comparatively better performance.

D. Random Forest Model

Random forest is a tree-based algorithm which involves the construction of several trees (decision trees) and then the integration of their output to enhance the model's generalization. The way trees are coupled is known as an ensemble method. Assembly is only a mixture of weak students (single trees) which produces a strong learner.

A random forest is a classifier consisting of a collection of tree structured classifiers h(x, k), k = 1,... where the k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input. Thus, Random forest will help to merge these multiple decision trees so that a more accurate and stable classification can be achieved. We have used random search grid to get best hyperparameters and used those parameters to get optimal performance of the model. The results of random forest are further discussed in results section.

E. Decision Tree Model

Decision Tree is a structure that looks like a tree in which each node has child with some information. Each branch of decision tree represents the result of some operation or test and the leaf node represents the label of class/target. Decision Tree is one of the supervised machine learning methods. The algorithms related to decision tree are referred as CART (Classification and Regression Trees). We do not really need to tune hyperparameters for Decision Tree, so it is very easy to implement the model.

Different algorithms of decision tree were implemented earlier on the same dataset as discussed in the section of literature review. We have tried to implement the basic decision tree classifier which is able to classify the target.

F. Support Vector Machine

The Support Vector Machine (SVM) is a supervised learning technique that classifies the data points into two classes based on a hyper plane. To implement this method, we used “SVC” class which is known as support vector class from library “svm”. For SVM, there are three different kernels which are linear, polynomial and gaussian. The default kernel is gaussian which is also known as Radial Basis Function (RBF). We have used rbf kernel for the model.

We have expanded the experiments of Parul Sinha and Poonam Sinha by doing some operations on data to improve the performance of SVM model [5].

IV. Exploratory Data Analysis

For data visualization, seaborn and matplotlib libraries are used from python. First analysis is on the distribution of class labels. The classification label here is if the patient has chronic kidney disease or not. Below is the count plot that shows the distribution of class labels. The dataset has 250 CKD and 150 NonCKD patient instances. Fig.1 shows the countplot of the distribution of class labels.

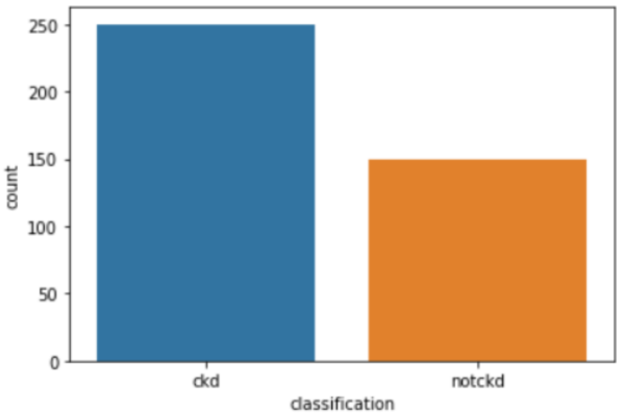


Fig. 1. Distribution of Class Labels

The count plot shows the data consists of class imbalance issue. Therefore, this issue has been resolved during implementation by random sampling method which resampled the dataset with replacement of NonCKD instances. Another analysis is done on the null values of dataset. There are so many arbitrary, erroneous and unknown values which has been rectified during preprocessing of the dataset. Fig.2 shows the heatmap of the number of null values of overall dataset.

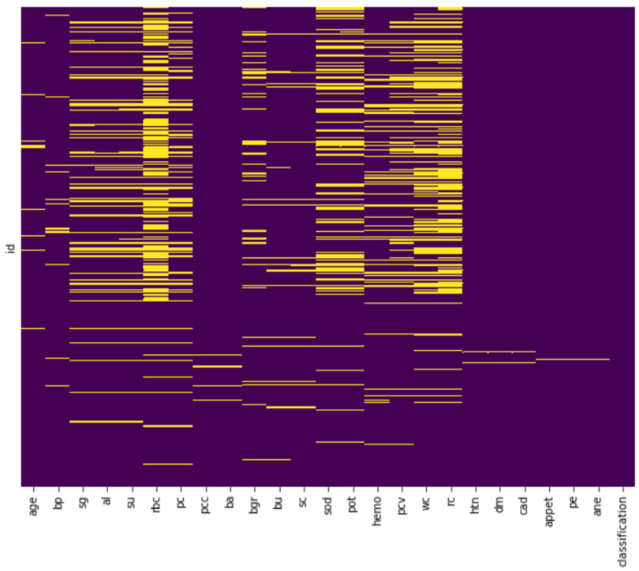


Fig. 2. Null values in Overall Dataset

For a classification problem, correlation between the features and class labels is very crucial to perform the feature engineering. The below heatmap shows the correlation between features and the class labels. As, you can see from this heatmap that diabetes(dm) and hypertension(htn) are highly correlated to target classification feature variable. Also, albumin(al), blood glucose random(bgr), blood urea(bu), pus cell(pc), pedel edema(pe) are among the greatly correlated features with our class label.

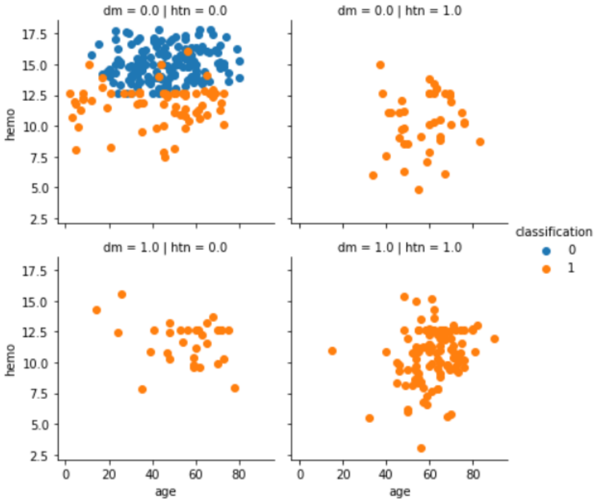


Fig. 3. Facet grid indicating relationship between attributes

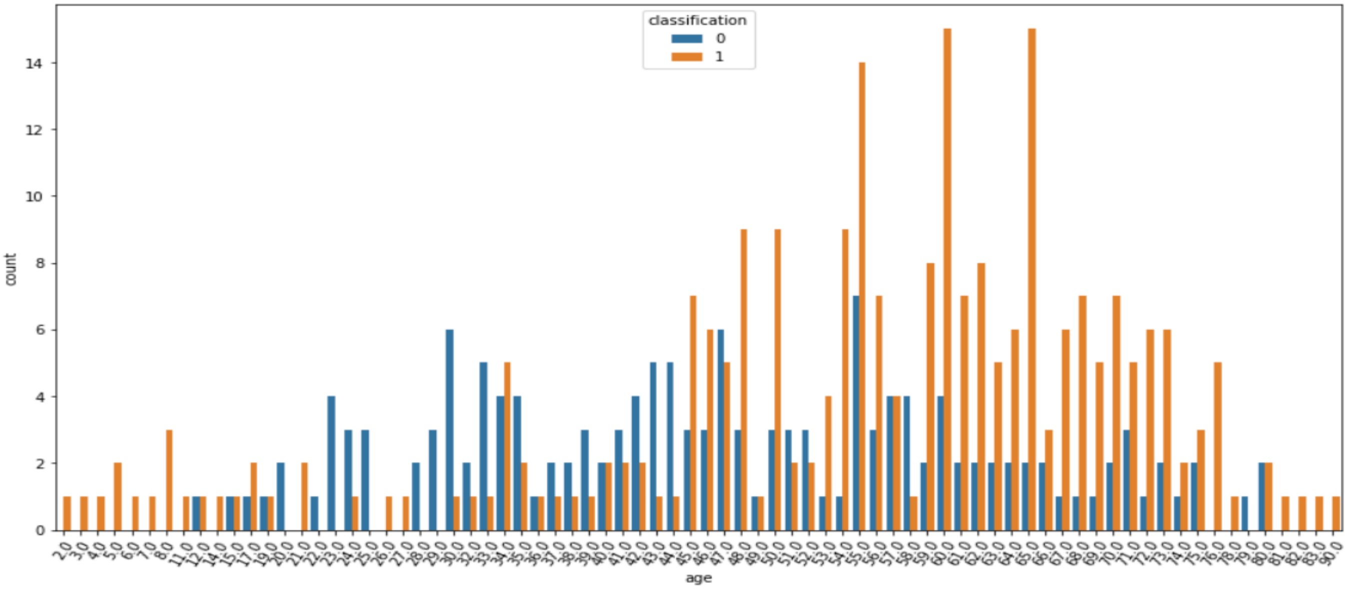


Fig. 4. Distribution of Class Labels based on Age feature

A facet grid was plotted to show the relation of hypertension and diabetes on the probability of having a chronic kidney disease based on the age and hemoglobin factor. This facet grid shows that nonCKD patients has no history of diabetes or hypertension disorders but, people with the same trait with the hemoglobin lower than 12.5 values are potential victims of the CKD. On the other hand, most of the patients with either diabetes or hypertension are having CKD. The patients with both the disorders are mostly over 40+. It was observed that people with high hemoglobin and less hypertension and dm have very low chances of having a kidney disease. However, people with hypertension and dm have the high probability of having the kidney disease. Fig.3 illustrates facet grid plot that demonstrates the relationship of these features.

The analysis on age factor of the data determined that people from 45-60 had high chances of getting a chronic kidney disease. It is observed that there is sudden spike in number of patients after age 45. Fig.4 shows the count of class labels with respect to age factor.

V. Experimental Analysis

In this paper, we have chosen a dataset from UCI repository for CKD which is collected from the hospital nearly 2months of period [6]. Chronic Kidney Disease dataset has over 25 number of attributes and 400 instances. Attribute values are continuous and categorical where string categorical attributes are converted into numerical categorical attributes during preprocessing of the data where classification algorithms are applied to determine the accuracy and the performance of the classifier for disease prediction. There were some null and random values in the features which we have taken care of by using some traditional methods. The dataset is partitioned into 75:25 ratio and Table.1 represents the description of all features of dataset.

TABLE I
Feature Description

Attribute	Description	Data Type
Age	Age of patient	Numerical
Blood Pressure	Blood pressure level	Numerical
Albumin	Albumin level	Nomical
Sugar	Sugar level	Nominal
Red Blood Cells	Red blood cell level	Nominal
Pus Cells	Pus cells level	Nominal
Pus Cell Clumps	Existence of pus cell clumps	Nominal
Bacteria	Existence bacteria in cells	Nominal
Blood Glucose Random	Glucose level in blood	Numerical
Blood Urea	Level of urea nitrogen in blood	Numerical
Serum Creatinine	Creatinine count in blood cells	Numerical
Sodium	Sodium count in cells	Numerical
Potassium	Potassium count in cells	Numerical
Haemoglobin	Haemoglobin count of blood cells	Numerical
Packed Cell Volume	Volume of red blood cells	Numerical
White Blood Cells	Count of white blood cells	Numerical
Red Blood Cells	Count of red blood cells	Numerical
Hypertension	Presence of hypertension	Nominal
Diabetes Mellitus	Presence of diabetes mellitus	Nominal
Coronary Artery Disease	Presence of CAD	Nominal
Appetite	Appetite indication of patient	Nominal
Pedal Edema	Presence of pedal edema in feet	Nominal
Anemia	Presence of Anemia in blood cells	Nominal
Classification	Class label	Nominal

A. Class Imbalance

In the class label of our dataset, the number of ckd labels were much higher compared to the non-ckd labels. This class imbalance issue lead to a very high accuracy on our majority class label, whereas the accuracy of minority class label remained low. To overcome this issue, random sampling of the data was performed. Sampling of minority class labels was done by randomly choosing instances of minority class. Samples were created such that the size of both the class labels in the final dataset remains equal. Thus, the classification results we achieved were on the balanced dataset.

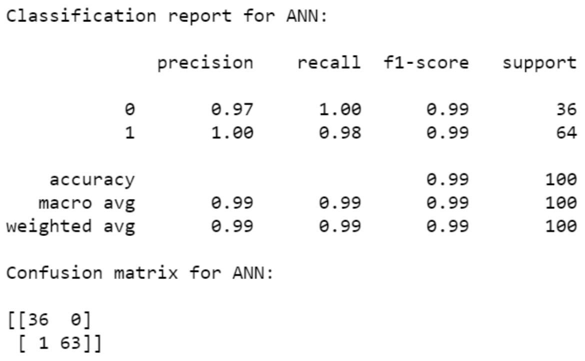
B. *Overfitting*

The factors that can lead to overfitting are bias and variance. On our dataset, the factor that leads to overfitting is high variance. High variance is when the model performs better on testing set and not on training set. The high variance was included to the size of the dataset. The number of instances in our dataset are very less, hence the model is unable to train on a wide range or variety of data. Due to high variance, KNN, random forest, decision tree and SVM are unable to generalize on the training data. Also, due to the class imbalance issue the model predicts the majority class well which leads to high accuracy and overfitting. Even after sampling technique, since only replications are created there is no variety of instances that the model finds to learn. All these factors have been observed during building of the models which might lead to overfitting on the future testing samples.

VI. Results

A. *ANN Model*

A feed-forward artificial neural network is used for classifying the data whether the patient has chronic kidney disease or not. We tested the data by training it on simple feed-forward neural network model by adding dropout layers to avoid overfitting. The final layer of the ANN model has two units with softmax activation function to give the probabilities of both the classes. It was observed that ANN model performed the best among all the models that we used. The sensitivity and specificity for ANN model shows accuracy of 99%. Only one misclassification was observed in the confusion matrix which was a false positive which is quite acceptable in the health domain. Fig.5 demonstrates the classification report and confusion matrix of KNN model.



```
Classification report for ANN:

              precision    recall  f1-score   support

           0       0.97      1.00      0.99        36
           1       1.00      0.98      0.99        64

    accuracy                           0.99       100
   macro avg       0.99      0.99      0.99       100
weighted avg       0.99      0.99      0.99       100

Confusion matrix for ANN:

[[36  0]
 [ 1 63]]
```

Fig. 5. Classification Report and Confusion Matrix for ANN model

Fig.6 shows the graph that indicates the performance for training accuracy vs training loss.

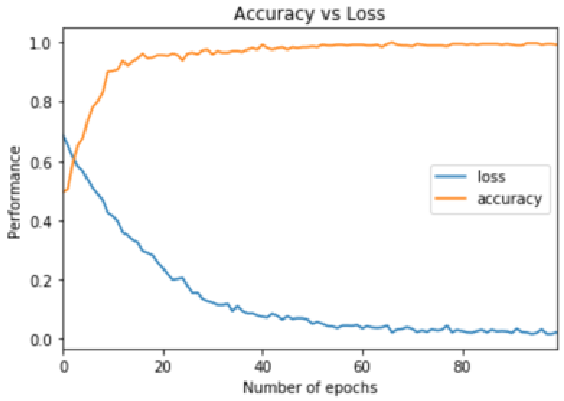
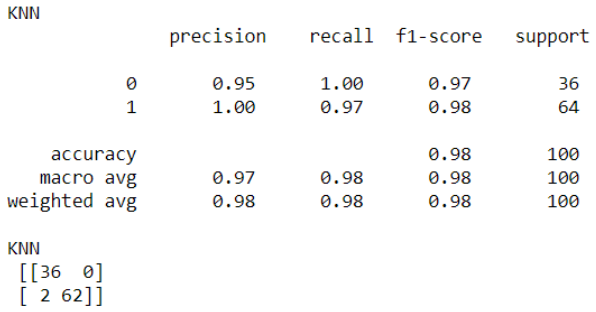


Fig. 6. Performance for training accuracy vs training loss

B. *KNN Model*

The hyperparameter n-neighbors were tested by considering range from 1-10. It was observed that k value between 0-5 showed a very low error rate. Hence, neighbors=5 was selected as the optimal hyperparameter value while training the KNN model. Training of KNN model was done on 75% of the overall dataset. As the overall number of instances of our dataset are very less, the model overfits on the testing data. Even after adding samples of minority class, the size of the data remains small for training the model precisely. Fig.7 illustrates the classification report and confusion matrix of KNN model.



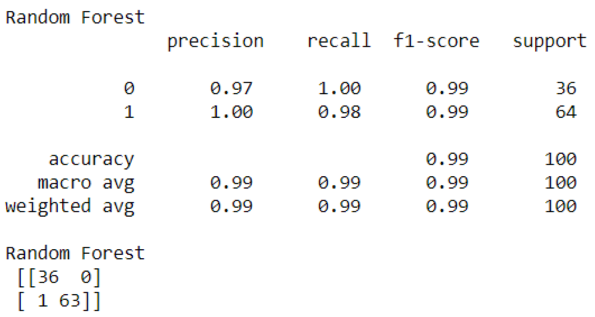
```
KNN
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        36
           1       1.00      0.97      0.98        64

    accuracy                           0.98       100
   macro avg       0.97      0.98      0.98       100
weighted avg       0.98      0.98      0.98       100

KNN
 [[36  0]
 [ 2 62]]
```

Fig. 7. Classification Report and Confusion Matrix for KNN model

C. *Random Forest Model*

The hyperparameter selection was done using random search approach. Hyperparameters that were considered for tuning were n-estimators, maximum depth, and bootstrap. Random search was done using 3-fold cross validation and 10 iterations. Initially n-estimators withing range of 100-200 and maximum depth within the range of 10-100 were considered. The best parameters found were n-estimator as 110 and maximum depth as 40. It was observed that random forest shall overfit the future testing samples mainly due to the size of data that we have trained on. The size of data was very small for random forest to train accurately on training data.

Fig.8 shows the classification report and confusion matrix of random forest model.

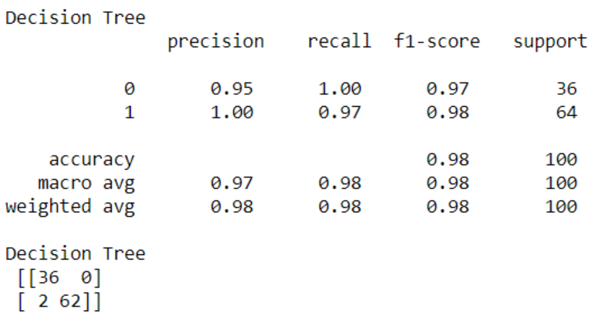


```
Random Forest
              precision    recall  f1-score   support

           0       0.97      1.00      0.99        36
           1       1.00      0.98      0.99        64

    accuracy                           0.99       100
   macro avg       0.99      0.99      0.99       100
weighted avg       0.99      0.99      0.99       100

Random Forest
 [[36  0]
 [ 1 63]]
```

Fig. 8. Classification Report and Confusion Matrix for Random Forest model

D. Decision Tree Model

Decision tree model was training on the training set based on its default parameters. Training for decision tree model was done on 75% of the overall dataset. The training data was trained without any cross-validation method. Decision tree resulted in a slight less accuracy as compared to random forest. It was observed that the split points and final class prediction is greatly influenced by the size of dataset. Hence, greater number of instances helps in finding the optimum split points which can avoid overfitting in the future. Fig.9 represents the classification report and confusion matrix of decision tree model.

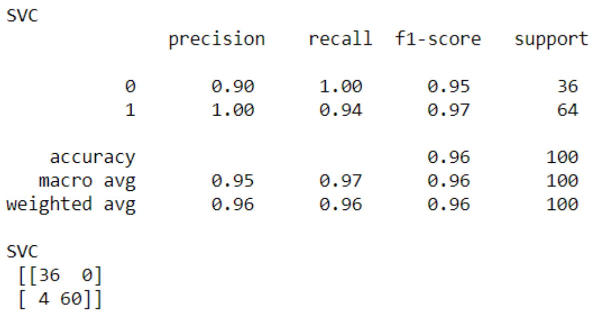


```
Decision Tree
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        36
           1       1.00      0.97      0.98        64

    accuracy                           0.98       100
   macro avg       0.97      0.98      0.98       100
weighted avg       0.98      0.98      0.98       100

Decision Tree
 [[36  0]
 [ 2 62]]
```

Fig. 9. Classification Report and Confusion Matrix for Decision Tree model

E. Support Vector Machine

Support vector machine was trained on its default parameters without any cross-validation of data. Due to smaller number of instances in the training data, the model has generalized the training data accordingly. SVM has the highest false negative rate hence compared to other models it is not performing well. Fig.10 shows the classification report and confusion matrix of SVM model.



```
SVC
              precision    recall  f1-score   support

           0       0.90      1.00      0.95        36
           1       1.00      0.94      0.97        64

    accuracy                           0.96       100
   macro avg       0.95      0.97      0.96       100
weighted avg       0.96      0.96      0.96       100

SVC
 [[36  0]
 [ 4 60]]
```

Fig. 10. Classification Report and Confusion Matrix for Support Vector Machine model

VII. Conclusion

In this study, we predicted the occurrence of Chronic Kidney Disease (CKD) at early stages in advance by studying the different traits of the patient. We used five different models for prediction and compared the results of all the models. The models that we used for predicting the occurrence of disease were a simple feed-forward neural network, KNN classification model, random forest, decision tree and support vector machine. As our analysis is based on the medicine domain, we focused on the sensitivity and specificity of the models rather than just accuracy. Thus, a highly sensitive test rarely overlooks an actual positive which is seen in our trained models. But, this might not be the exact scenario for any future instances.

For our dataset, the ANN model performed the best among all the five models based on its sensitivity and specificity. Random forest and KNN too showed some promising results by giving good percentage of sensitivity and specificity. On the other hand, decision tree and support vector machine did not perform as compared to the other three models. These models showed some misclassification in the false positive and false negatives results which is highly risky in the medicine domain. Apart from these results, the limitation that our study faced was of overfitting. We might encounter overfitting in the future testing samples due to a very small amount of data for our analysis. In our future work, we can try to expand our dataset by gathering more and a variety of instances of patient information. This information will help the prediction process to gain more insights of the data during learning. Thus, in for our future scope we can try to decrease the overfitting issue by collecting more instances which can lead our study to predict the chronic kidney disease occurrence more precisely.

References

[1] Sarada, J. and Lakshmi, Neelam Venugopal Muthu, “Data Analytics on Chronic Kidney Disease Data. IADS International Conference on Computing, Communications Data Engineering”, 2018. RNNs and beyond”, SIGNLL Conference (CoNLL), 2016.

[2] I.A. Pasadana, D. Hartama, M. Zarlis, A.S. Sianipar, A. Munandar, S. Baeha and A.R.M. Alam, ”Chronic Kidney Disease Prediction by Using Different Decision Tree Techniques”, Journal of Physics: Conference Series. Volume 1255, 2019.

[3] Basma Boukenze, Hajar Mousannif and Abdelkrim Haqiq, ”Performance of Data Mining techniques to predict in healthcare case study: Chronic Kidney Failure Disease”, International Journal of Database Management Systems (IJDMS) Volume 8, Number 3, 2016.

[4] Abhishek, Gour Sundar Mitra Thakur, Dolly Gupta, ”Proposing Efficient Neural Network Training Model for Kidney Stone Diagnosis”, Interna-

tional Journal of Computer Science and Information Technologies, Vol. 3 (3), pp 3900-3904, 2012..

[5] Parul Sinha and Poonam Sinha, “Comparative Study of Chronic Kidney Disease Prediction by Using KNN and SVM”, International Journal of Engineering Research and Technology (IJERT), Volume 4, Issue No. 12, 2015.

[6] P.Soundarapandian and L.Jerlin Rubini, “Chronic_Kidney_Disease Data Set”, UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease