 doi:10.20944/preprints202005.0458.v1

Cost-sensitive ensemble feature ranking and automatic threshold selection for chronic kidney disease diagnosis

Syed Imran Ali, Gwang Hoon Park, Sungyoung Lee

Department of Computer Science and Engineering, Kyung Hee University, (Global Campus), 1732, Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Republic of Korea.
imran.ali@khu.ac.kr; ghpark@khu.ac.kr; sylee@oslab.khu.ac.kr

Abstract: Automated medical diagnosis is one of the important machine learning applications in the domain of healthcare. In this regard, most of the approaches primarily focus on optimizing the accuracy of classification models. In this research, we argue that unlike general purpose classification problems, medical applications require special treatment. In case of medical diagnosis, apart from model performance other factors such as cost of data acquisition may also be taken into account. Since, models which are relatively expensive to operate would have diminished applicability in the field. Therefore, both performance and cost factors may be considered for designing the automated diagnosis solutions. In this regard, we have proposed two ensemble based cost-sensitive feature ranking techniques which tend to select an optimal feature subset by evaluating the cost-benefit trade-off on benchmarked chronic kidney disease dataset. We have also addressed the issue of automatic threshold selection which is generally faced by feature ranking approaches; hence enhancing the overall applicability of the solution. In this research, the main focus is on the application of decision-tree based classifiers in a cost-effective manner. As a case study, we have selected chronic kidney disease as the application of interest in which we demonstrate that our proposed approaches select features which are both useful and cost-effective. Furthermore, the proposed approaches are also evaluated against a number of comparative techniques and the results are promising.

Keywords: Cost-sensitive feature selection, ensemble models, decision tree classifiers, chronic kidney disease, random forests, gradient boosted trees

1. Introduction

Chronic kidney disease (CKD) as an ailment which affects working of the kidney. Generally, CKD is divided into multiple stages in which the later stages are denoted as renal failure when kidney is unable to perform its function of blood purification and balancing minerals in the body [1]. Hemodialysis is performed in order to supplant the kidney function which provides a temporary solution to the problem. Hence, it is of paramount importance that the CKD is detected at earlier stages where it can be addressed through medication and lifestyle changes [2]. CKD is a highly prevalent disease, according to an estimate one in nine Korean adults suffer from kidney disease [3]. Likewise, around 2.5-11.2% of the adult population in Europe suffer from it, while around 59% of all American adult population is at a high risk of developing kidney disease at some point [4, 5]. Hence, the high incidence and prevalence of CKD is attribute to its late diagnosis due to overburden healthcare system, especially in developing countries.

In this regard, a number of intelligent clinical decision support systems are designed which tend to automate the CKD diagnosis process [6, 7, 8]. These systems employ machine learning techniques which assist physicians in the diagnosis and treatment of CKD in an efficient manner. Based on a number of important indicators such as blood pressure, albumin levels, blood and urea tests, potassium, and other comorbidities e.g. diabetes, cardiovascular disease, etc., a patient is comprehensively assessed for CKD and its progression. Since, the earlier diagnosis of the disease

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

onset can improve chances of patient to favorably respond to treatment therefore most of the automated systems are optimized for enhancing the overall accuracy of the model [7]. A significant emphasis is devoted to build classification models that are reasonably accurate and easy to interpret by the physicians.

Most of the studies in the CKD domain assume that the cost of data acquisition is symmetric therefore the cost factor associated with each feature is generally ignored [8, 9, 10]. But since data for a number of features are not readily available without conducting medical tests and since medical tests are asymmetric, in terms of cost, hence this important factor may also be given its due importance in building the classification model [6, 7].

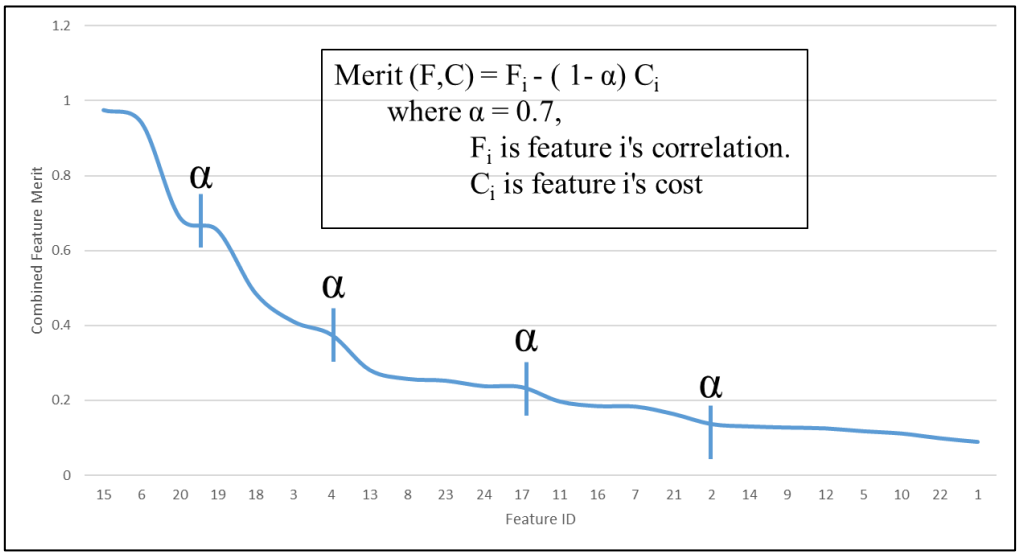

Figure 1. Cost-sensitive feature ranking and threshold selection

In this study we address the problem of cost-sensitive feature selection for building a classification model for the CKD. In this regard, the experimentation performed to assess the efficacy of the proposed ensemble feature ranker is limited to decision tree-based models. We have adopted chronic kidney disease as a case study and restated the feature selection problem in which both performance of the model as well as the accumulated cost of data acquisition are taken into account.

Feature selection has become an essential task in building classification models where the objective is to select a subset of useful features. The notion of usefulness is based on the feature assessment which quantifies the relevance and redundancy of a given feature in the dataset. Ensemble classification approaches have shown promising results where a number of homogenous or heterogeneous models are combined in such a manner as the overall performance of the model is enhanced [11]. Similar approaches for feature ranking are also adopted where a global ranking is obtained based on the individual ranking produced by the scoring functions present in an ensemble. Since, a ranked list doesn't provide with an explicit threshold point for retaining a subset of features therefore the selection of a threshold among a set of candidates a non-trivial problem [12]. As it can be seen in Figure 1 that multiple threshold values can be selected over the feature merit curve.

In this study we have applied the ensemble approach to cost-sensitive feature ranking. To the best of our knowledge it is the first study which has addressed the notion of data acquisition cost within the framework of cost-sensitive feature ranking. We have proposed two ensemble ranking techniques which uses classifiers as heterogeneous scoring functions. Ensemble-1 combines all the scores and afterward automatically selects a threshold value whereas the sensemble-2 applies thresholds to individual ranks and then a resultant ranking is generated. We demonstrate that both the aforementioned techniques generate a slightly different result whereas the ensemble-2 is more conservative in selecting features, hence it selects relatively fewer number of features with a lower overall cost. A schematic diagram for an ensemble feature ranker is shown in Figure 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

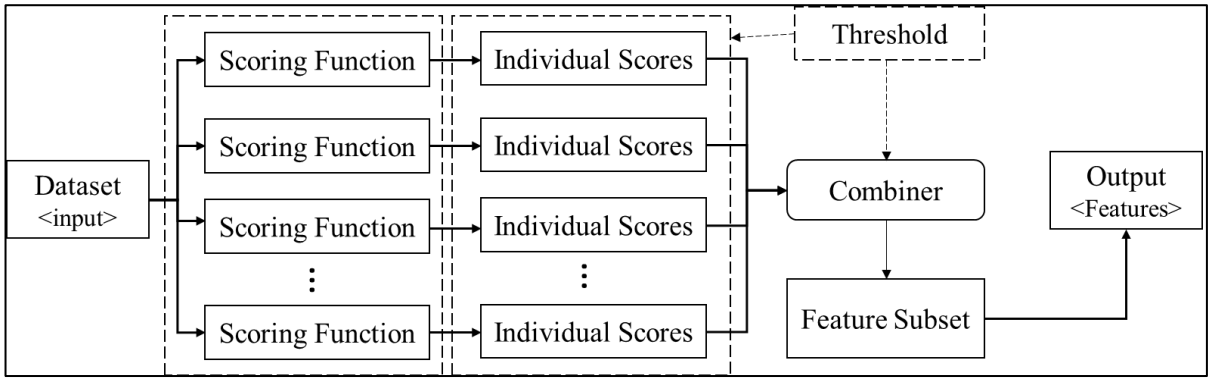

Figure 2. Schematic diagram of an ensemble feature ranker

The major contributions of this study are as follows:

- Framework for cost-sensitive feature ranking
- Reduce ranking variability with a decision-tree based ensemble scoring function
- Automatic threshold value selection based on the proposed framework

This rest of the paper is organized as follows: Section 2 deals the literature review on the subject. Proposed methodology is discussed in section 3 in which both the ensemble techniques are elaborated. Section 4 deals with the experimentation and the case study results in which we provide detailed treatment of both the proposed approaches along with their comparison with other related techniques. Conclusion of the study is provided in section 5 along with a set of future directions for extending the research.

2. Literature Review

A number of studies have shown that feature selection improves the accuracy of the classification task [6-12]. In this regard, a number of feature selection techniques are proposed. Feature selection is similar to dimensionality reduction whereas the objective of the former is to retain the semantics present in the original dataset while the latter transforms the data in such a manner as the overall dimensions of the data are reduced [13]. Feature selection techniques are generally grouped into three broad categories i.e. filter-techniques, wrapper-techniques, and embedded-techniques. Filter techniques score features based on the general characteristics of the dataset [10]. In this regard, most of the filter approaches are based on evaluating correlation between features and the class label. Hence, features having high correlation with the target concept are regarded as useful features. Feature ranking approaches are generally based on filter methods but ranking can also be produced by employing a classification model which in turn evaluates a subset of features [11]. Wrapper approaches involve classification algorithm in the process of evaluating a feature or a subset of features. In this regard, a feature subset generation step is followed by an evaluation step [14]. The main objective is to find a subset of features which are neither irrelevant nor redundant. Filter methods are employed when number of feature are very large since these methods are fast and don't get bogged-down in pairwise comparison of the candidate feature sets. Wrapper methods generally produce results which are relatively more optimized and accurate than that of the filter methods whereas the latter produces result in relatively less time [10]. Embedded methods select a subset of features as an integral part of the process of building a classifier such as a decision tree algorithm which selects most appropriate feature as it grows the tree [15]. Following are some of the representative work in which feature selection techniques are used on the CKD dataset.

Salekin and J. Stankovic [16] proposed a wrapper based feature selection approach which reduces the overfitting of the random forests classification model on CKD dataset. The reported resultant F1-measure of the model on top 5 features is 99.80%. Furthermore, the authors also reported promising results in terms of reduced cost. Z. Chen et al. employed three models i.e. K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Soft Independent Modeling of Class Analogy for decision modeling of CKD patients. The proposed approach achieved accuracy of 93%. It is also

 doi:10.20944/preprints202005.0458.v1

reported that SVM was more robust in dealing with noisy data as compared to other models and hence achieved an accuracy of 99% [17]. A.A. Serpen [18] used C4.5 decision tree model on CKD dataset. The resultant tree model produced 8 production rules of the form IF<condition> THEN<conclusion> and achieved an accuracy of 98.25%. Whereas, Al-Taee et. al. [20] reported lower accuracy on the same data. Furthermore, authors have also identified 5 salient features in their study. In an another study the same framework is used as reported in [20], in which they used three classifiers on a CKD dataset complied from Prince Hamza Hospital, Jordan. The study reported that the decision tree model performed reasonably well on a number of performance metrics [21]. N. Tazin et al. [21] used a number of classification models such as SVM, Naive Bayes, KNN and decision tree on CKD dataset. Subsequently, a feature ranking is generated from which top 10 features were selected. It is reported that decision tree algorithm produced a model having accuracy of 99.75%. Huseyin Polat et al. [10] proposed a feature selection technique for SVM based classification model. The authors used hybrid feature selection by leveraging both filter and wrapper methods. They reported accuracy rate of 98.50% on SVM using 'Best First' search technique using 13 attributes. Likewise, Adeola Ogunleye and Qing-Guo Wang, ref. [7], selected top 13 features for feature selection and performed classification using an optimized random forests classifier for CKD dataset. They reported an accuracy of 100%. In Ref. [22], the authors experimented with SVM and Artificial Neural Network (ANN) on CKD dataset. They reported that ANN produced comparatively higher accuracy model as that of SVM. All the experiments are performed on top 12 features. Jiongming Qin et al. [23] experimented with a number of different data imputation configurations on a set of multiple classifiers. They reported that random forests achieved highest accuracy of 99.75% for the CKD diagnosis, while logistic regression was able to produce accuracy of 98.95%. Afterward, authors proposed an integrated model which employed both the aforementioned classifiers along with perceptron and subsequently produced an accuracy of 99.83%. Alvaro Sobrindo et al. [6] performed a comprehensive study on CKD diagnosis using various machine learning algorithms. The authors reported highest accuracy achieved by decision tree based models in the pool of candidate models which included NB, SVM, ANN, KNN.

We have provided a general overview of the feature selection techniques and classification algorithms applied to CKD diagnosis and it can be observed that decision-tree based models are one of the popular modeling approaches for the CKD diagnosis. Since our proposed approach is based on feature ranking therefore we herein mention a few studies which have addressed the problem of automatically selecting an appropriate threshold value. Most of the studies use a fixed threshold value for retaining a set of top features [24, 25, 26]. But as it is observed that a fixed threshold value may over-select or under-select an appropriate number of features [12, 27, 28, 29]. Authors in [30] used data complexity measures for selecting a threshold value while [31] used a minimum union method to combine multiple rankings and produced promising results on high dimensional datasets [32]. Hence, a number of threshold selection measures are reported in the literature. Our proposed approaches are based on novel idea of feature-cost interaction curve. Please note that it is the first study in which an ensemble feature ranking is applied to the CKD diagnosis problem. In the following section we elaborate the proposed methodology.

3. Proposed Methodology

In this section we elaborate the underpinnings of our proposed approach for cost-sensitive feature ranking. In most of the feature ranking techniques a feature score is produced which in turn is used for the final feature ranking. Afterward a threshold value is used to filter-out undesirable features. The retained features are fed to a data classification model. One of the major challenges in this regard is to find an appropriate threshold value as shown in Figure. 1. Furthermore, it is also important to select the feature scoring function which is not biased towards any particular data characteristics e.g. information gain tend to favor attributes which take on a large number of distinct values [33]. Filter-based feature weighting measures such as Gini index generally doesn't account for feature interaction and hence is not comprehensive enough to capture the complementary features.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

Hence, it is of paramount importance that the scoring function is relatively free from data bias and threshold selection takes place automatically based on the characteristics of the dataset.

Our proposed approach is based on an ensemble of classifiers. We have used three decision tree classifiers; CART, Random Forests (RF) and Gradient Boosting Trees (GBT). Since, these are popular tree based classification models, therefore we have employed the same in our ensemble. It is important to note that the ensemble can be executed in parallel, therefore the running time of the ensemble is proportional to the running time of its slowest classifier. In this regard, using concurrent processes we can execute the classifiers and afterward combine their results. The main objective of the proposed technique is to score features based on their individual importance as well as their interaction with other features in the dataset as well. Once a reliable feature score is obtained, then based on their individual importance values, features are ranked in descending order of their importance. Likewise, each classifier produces both a feature score and a ranked feature list. These disparate lists are combined. In this research we propose two approaches for combining features i.e. score-based combiner and rank-based combiner. The former approach deals with taking the average score for a given feature, i.e., feature-score (f-score), whereas the latter approach is based on frequency of a given feature in multiple ranked lists. Afterward, cost of each feature is accumulated. We have normalized the cost values; hence overall cost of the entire feature list accumulates to 1. An automatic threshold value is selected based on the accumulated cost of features and their worth i.e. f-score. Finally, a subset of features is selected and fed to the subsequent task i.e. classification. Proposed ensemble-1 is depicted in Figure 3, while Figure 4 shows ensemble-2. The input dataset is represented as a data matrix where $D = (X, Y)$. Where ($x \in X$, $y \in Y$). Therefore, using this definition an instance can be represented as $x_i = [x_{i1}, x_{i2}, x_{i3}, \ldots, x_{im}]$ such as $i = 1, 2, 3, \ldots, n\}$ and $y \in \{1, -1\}$. The overall objective is to construct a model using training data x, y = f(x) such as which minimizes the overall error on test data x' where $x \neq x'$.

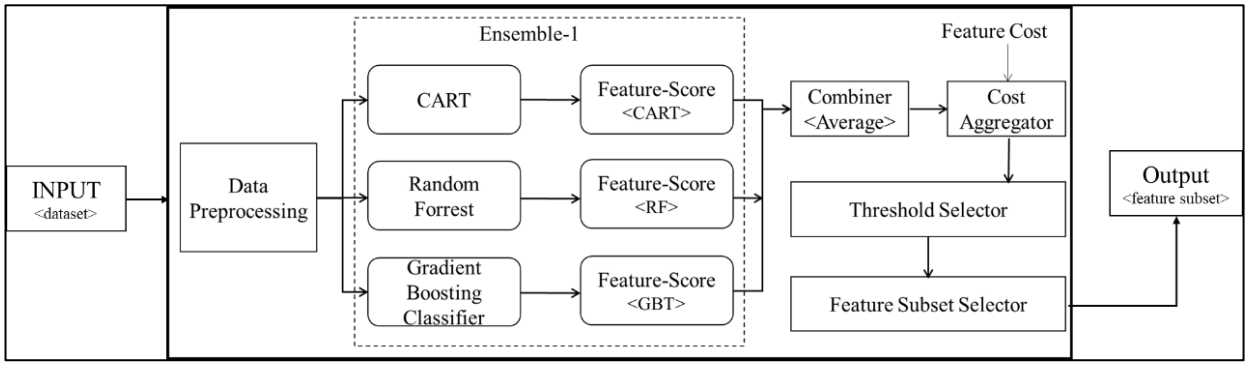

Figure 3. Architecture of the proposed ensemble-1

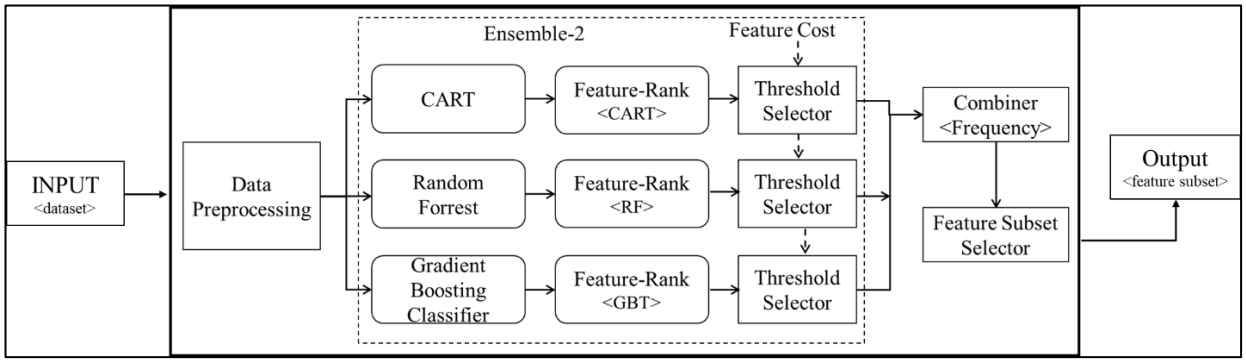

Figure 4. Architecture for the proposed ensemble-2

3.1 Data Preprocessing

It is a pre-requisite task which ensures that the data are of high quality. The major operations performed by this component are missing values imputation and feature id-ness detection and removal. For the purpose of this study we have used k-nearest neighbor to impute missing values, where k=3. This technique selects a set of records from the dataset which are similar to the missing

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

value record and subsequently imputes the value based on the local information of the selected set. Furthermore, we used Euclidean distance for computing the similarity of each record. Numerical values are replaced with taking the mean value of the selected samples subset while mode value is used for nominal features. The chronic kidney disease dataset, discussed in the subsequent section, contains 242 records with missing values while the complete dataset has 400 records. Therefore, around 60% of the records have one or more missing values.

3.2 Classifier-Ensemble

The proposed approach uses decision tree based classifiers. The main impetus for using tree-based classifiers is two-fold. First tree-based models are easy to interpret therefore in domains such as healthcare, it is important to assess the validity of the model through visual inspection, where possible. Along with the statistical evaluation, the interpretability aspect of the model is also important. Hence, where the interpretability plays an important role, one can opt for standalone decision tree models such as CART. Standalone models generally, lack sophistication of more complex models which generally yield higher accuracy at the expanse of interpretability. Hence, where interpretability is of secondary concern one can opt for tree-based models such as Random Forrest and Gradient Boosting Trees. Both these models have shown promising results on small to medium scale structured datasets [7, 6, 8, 23] and hence are suitable for the consideration of the ensemble which is comprised of tree-based models. Following is the brief description of the classification models used in this study:

- *CART*: The CART (Classification and Regression Tree) is one of the popular decision tree induction algorithms. CART employs recursive partitioning to build the model based on decision tree data structure. CART, like other decision tree models, places the most important feature as a root node while the subsequent nodes are internal nodes in a hierarchical structure and the leaf nodes contain the classification label. Each non-leaf node is a test node which branches out in order to accommodate other features. Generally, Gini index is used to calculate the importance of a feature at a specific level in the tree. The main objective of the CART algorithm is to construct the model which can separate the data into homogenous subsets with respect to the class label. The equation (1) is used to define the Gini index:

$$g(n) = \sum_{a \neq b} p(a|n)p(b|n) \quad (1)$$

Where 'a' and 'b' are two classes in the dataset. The value of Gini index is 1 when there is an even distribution of 'a' and 'b' in a given node 'n', and the value is 0 when the distribution is homogenous. Decision regarding splitting 's' over a given node 'n' is defined as following:

$$\theta(s, n) = g(n) - p_L g(n_L) - p_R g(t_R) \quad (2)$$

Where 'p_L' a set of instances at the left child node and 'p_R 'denotes a set of instances at the right child node, respectively. The decision of split is based on the maximum split 's' subject ot the value of $\theta(s, n)$

Model acquired through CART algorithm can be converted into a set of decision lists in which each traversal from the root node to a specific leaf node results in a rule. In this case, each non-leaf node becomes an antecedent condition while the leaf node is treated as a consequent. The resulting rule may have one or more conditions given that there are more than one non-leaf nodes in a particular branch. In this regard, each node is treated as a conjunction and the branch traversal results in a set of conjunctions. While the entire decision tree is translated into a disjunction of conjunctions.

- *Random Forests:* It creates a set of pre-defined tree models based on bootstrapped data for each tree model. It generally performs well for small to medium sized datasets while resulting model is robust against overfitting, feature correlation, feature interaction and spurious patterns [34]. The key approach of this algorithm is to create a set of bootstrapped datasets through which a set of randomly selected features are selected and a tree model is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

created. The standard random forest model is a combination of binary decision trees. The RF algorithm uses bootstrap aggregating technique for learning a set of tree models. In this regard, a training dataset X contains a set of instances $x_1, \dots, x_n$ and their corresponding classification label Y such that $Y = y_1, \dots, y_m$. Using bagging repeatedly for 'K' times and sampling with replacement the following steps are used to build the final model:

- Training: foreach model d in K:
  - Create a bootstrapped dataset with replacement
    - $X, Y: X_d, Y_d$
  - Build a tree model based on bootstrapped dataset
    - T_d **on** X_d, Y_d
- Testing: retrieve test sample x_t and feed it to K models such as:
  - $C(x_t) = \textbf{mode}\{T_1(x_t), T_2(x_t), \dots, T_d(x_t)\}$

The final label is selected based on the predictions produced by individual models through a majority voting scheme. Although the resulting model is not fully interpretable as that of CART but the individual decision trees can be extracted.

- *Gradient Boost Classifier:* It is yet another powerful decision tree model which produces a set of tree models where each individual tree is a weak model. It combines a set of weak models provided each model (φ) is slightly better than a random guess i.e. $\varphi > \frac{1}{m}, m = 2$ for a binary classification problem. This approach is based on three aspects i.e. a differentiable loss function, a pre-defined set of weak learners and additive technique to incorporate weak learners, sequentially. As weak learners are added to the model the error recorded through the loss function minimizes by applying gradient descent approach. The basic approach of gradient boost classifier is as follows:
  - Initialize
    - Initialize model with a constant value
      - $F_0(x) = \min\{w_1, w_2, w_3, \dots, w_d\}$ where $w_1, \dots, w_d$ denotes error of weak learners
    - Loop through each model in the set of pre-defined model set
      - foreach model d in K
        - Compute pseudo-residuals through negative gradient $g_t(x)$
        - Train model to fit pseudo-residuals w(x, Θd)
        - Optimize for step-size in the direction of best gradient descent
        - Update the function estimate
    - Output the model: $F_K(x)$

3.3 Combiner

It plays an important role in the overall proposed architecture. Once individual feature scoring is completed, each classifier produces feature scores. In this regard, we propose two approaches to combine disparate feature scoring lists.

- Approach 1 (henceforth ensemble-1) combines average score of each feature across the score functions 's' using the equation (3):

$$F = \frac{1}{|s|} \sum_{i=1}^{s} \sigma(f_i) \quad (3)$$

Hence, the score of each feature is the average score across three independent scoring functions 'σ'. Please note that the scoring values are scaled between 0 and 1 before applying equation 3.

```
N: number of models in ensemble // i.e. 3
D: dataset
foreach m in N do:
  score[m] = score_function(m, D)
L = average (score [1: N])
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

```
*L = sort(L) // resultant list, *L, is sorted in descending order
cost = assign_cost (*L, cost) //accumulated cost
T = intersection(*L, cost)
S = retained (*L, T) // retained features in *L after applying T
```

Approach 2 (henceforth ensemble-2) deals with the ranks produced by each scoring function 'σ'. The individual ranks can be obtained by the scoring produced by a given classifier. Afterward, frequency of each feature across the scoring functions is computed as follows:

```
L = merge(L(∅1), L(∅2), L(∅3)) //size of L1, L2, L3 may differ
*L = sort(L)     // resultant list, *L, is sorted in ascending order
α = *L // α points to the first element in *L
β = α + 1 // β points to the second element in *L
foreach f in *L do:
IF (α <= β)
      F[α] += 1     // F[.] is a frequency list for the items in *L
L.remove(α)       // removes the first element
α = L
β = α + 1
```

3.4 Feature cost aggregator

Both the aforementioned approaches produce a unified list of features. Features are arranged in the descending order of their importance. Individual cost of a feature is retrieved and accumulated in a top-bottom manner as given in equation 4.

$$Cost(F_i) = Cost(F_i) + Cost(F_{i-1}) \quad (4)$$

Where i = 1, 2, 3, …, m. Therefore, Cost(F_0) = 0

In this study, numeric values pertaining to cost are normalized between 0 and 1.

3.5 Threshold and Feature Subset Selector

The purpose of threshold value is to select a subset of features from the given feature list after incorporating the cost value. Ensemble-1 produces a feature list based on averaged scores. In this regard, we can find a point of intersection between feature score and the corresponding accumulated cost values. The point of intersection between feature score 'f-score' and accumulated cost score 'c-score' can be found where $f_{score} < C_{score}$. In case of Ensemble-2 we have a list of ranked features of the form $< feature: frequency >$. A sample graph based on feature-cost intersection curve is depicted in Figure 5. The F-score depicts the combined worth of a feature which is shown as a blue line in the descending order of their importance. While the orange line show the accumulated cost as defined in equation (4). A threshold value is automatically selected based on the point of intersection e.g. in the following figure it is right above feature no. 6. Hence, all the features starting from feature number 3 leading up to feature number 6 will be retained while rest of the features will be discarded. Since, after feature number 6, the accumulated cost outweighs the importance of features.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

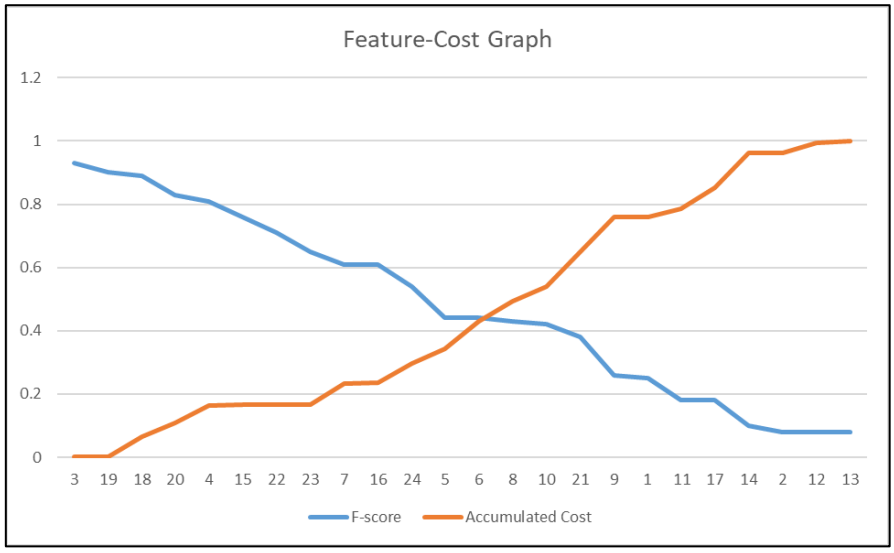

Figure 5. A sample feature-cost intersection graph

In case of Ensemble-2 we combine individual feature lists into a consolidated list by accounting for the occurrence of a feature in multiple lists and taking $^{2}/_{3}$ of the ranked features. For example, we have three features such as α, β, γ, placed at arbitrary positions in three lists produced by different scoring functions, f_1, f_2, and f_3. Then we compute the frequency of these features i.e. $<\alpha{:}3>$, $<\beta{:}2>$ and $<\gamma{:}1>$. According to the aforementioned selection strategy, we select both α, β features based on our heuristic of 2/3 of ranked features while discard γ and all other features which are at the equal rank as that of γ or lower. The intuition between the second approach is that if a feature appears more frequently in different lists then it is less likely due to any spurious pattern or any particular bias of the scoring function. Table 1 shows a sample scenario for the Ensemble-2 approach. In this case we have three different lists. We generate an integrated list by taking into account the frequency of a particular feature regardless of its position in the list. The highest score of a feature is determined by the number of scoring functions in the ensemble. Since, we have three classifiers, therefore, the highest score a feature may get is three. Hence, the select features are denoted with bold letters while the remaining features are discarded. Consequently, the total cost incurred by the selected feature subset would be 0.42 i.e., around 58% cost reduction.

It is important to note that the lists generated by scoring functions may not be of equal size. It is due to the fact that for each scoring function e.g. CART, feature weights are obtained and then subsequently based on the intersection of the F-score and the accumulated c-score, similar to ensemble-1 approach, a subset of features is selected for each function. More details regarding this step are presented in the experimentation section.

Table 1: Ensemble-2 frequency based feature ranking

List 1	List 2	List 3	Feature ID	Frequency	Accumulated Cost
3	19	3	**3**	3	**0.0**
6	18	19	**19**	3	**0.1**
20	3	20	**18**	3	**0.2**
19	15	4	**20**	2	**0.3**
7	5	22	**4**	2	**0.33**
16	8	18	**15**	2	**0.39**
22	14	23	**22**	2	**0.4**
18	2	12	**23**	2	**0.42**
4	1	21	7	1	0.46
23	9	11	16	1	0.60
…	…	…	…	…	…
17	10	24	13	0	1.0

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

In our proposed architecture we can utilize both ensemble combining approaches. The former approach tends to select a relatively larger feature set as compared to the latter approach which selects features in a more conservative manner. We have evaluated both these approaches are their results are discussed in the subsequent sections.

4. **Experimentation**

This section deals with the experimentation details of the study. In this regard, a brief description of the dataset is provided along with a summarized analysis of the quality of the dataset. Afterward, we elaborate the performance metrics used and the interpretation of the results. Furthermore, we carry on two sets of experiments. Experiment 1 deals with evaluating the efficacy of ensmeble-1 approach with that of baseline models, while ensmeble-2 is compared with baseline models in experiment 2. Once, we establish the performance of both the proposed approaches, we then compare them with each other over a number of performance metrics including the predictive accuracy and incurred cost.

In order to demonstrate the efficacy and applicability of our proposed approach, we have used a benchmark dataset from University of California (UCI) online repository. The chronic kidney disease (CKD) is a real world dataset acquired over the period of two months by Apollo Hospitals, Tamilnadu, India [35]. One of the major reasons of incorporating CKD dataset in this study is due to the availability of cost factor (adopted from ref. [16]) along with the recent studies reporting significant scholarly work on developing chronic kidney disease diagnosis and management systems. In this regard, our objective is to demonstrate the importance of considering both statistical evaluation of the prediction model along with the economic considerations of utilizing the model in domains such medical diagnosis.

4.1 Dataset description

The CKD dataset is composed of 400 instances where each instance in comprised of 24 attributes excluding the class attribute. There are 13 categorical attributes while 11 of the attributes have numerical values. The dataset is used to model a dichotomous decision variable i.e. 1 represents a given patient is diagnosed with the disease, while -1 denotes otherwise. The overall dataset has 250 CKD patients while rest are non-CKD patients. The acquired data are preprocessed in order to remove missing values and ID attribute. Table 1 provides a summary of the CKD dataset, along with the economic cost of acquiring data for a particular feature, (ref. [16]).

Table 2. Chronic Kidney Disease dataset description

ID	Attribute	Cost	Description	ID	Attribute	Cost	Description
1	Age <age: numerical>	1	In years	13	Sodium <sod: numerical>	4.2	mEq/L
2	Blood Pressure <bp: numerical>	1	Mm/Hg	14	Potassium <pot: numerical>	50	mEq/L
3	Specific Gravity <sg: numerical>	1	1.005, 1.010, 1.015, 1.020, 1.025	15	Hemoglobin <hemo: numerical>	2.65	Gms
4	Albumin <al: numerical>	26	0 ~ 5	16	Packed Cell Volume <pcv: numerical>	2.62	Integer valued
5	Sugar <su: categorical>	21	0 ~ 5	17	White Blood Cells Count <wc: numerical>	31	cells/cumm
6	Red Blood Cells <rbc: categorical>	40	1:Normal, 0:Abnormal	18	Red Blood Cells Count <rc: numerical>	31	millions/cmm

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

7	Pus Cell <pc: categorical>	31	1:Normal, 0:Abnormal	19	Hypertension <htn: categorical>	1	1:Yes, 0:No
8	Pus Cell Clumps <pcc: categorical>	31	1:Present, 0:Absent	20	Diabetes Mellitus <dm: categorical>	19.4	1:Yes, 0:No
9	Bacteria <ba: categorical>	51	1:Present, 0:Absent	21	Coronary Artery Disease <cad: categorical>	51	1:Yes, 0:No
10	Blood Glucose Random <bgr: numerical>	21	mgs/dl	22	Appetite <appet: categorical>	1	1:Good, 0:Poor
11	Blood Urea <bu: numerical>	12.85	mgs/dl	23	Pedal Edema <pe: categorical>	1	1:Yes, 0:No
12	Serum Creatinine <sc: numerical>	15	mgs/dl	24	Anemia <ane: categorical>	28.64	1:Yes, 0:No

The attribute importance in terms of its correlation with the class variable is shown in Figure 6. As it can be seen that a number of features have high correlation with the target concept. The nature of correlated feature and their treatment are generally domain dependent and therefore, a decision maker is generally involved to arrive at a decision for retaining/removing a highly correlated feature. In the absence of a domain expert, features with high correlation are generally preferred over low correlation features but in case where the availability of such features is not certain at the time of decision making then it is recommended to remove highly correlated features.

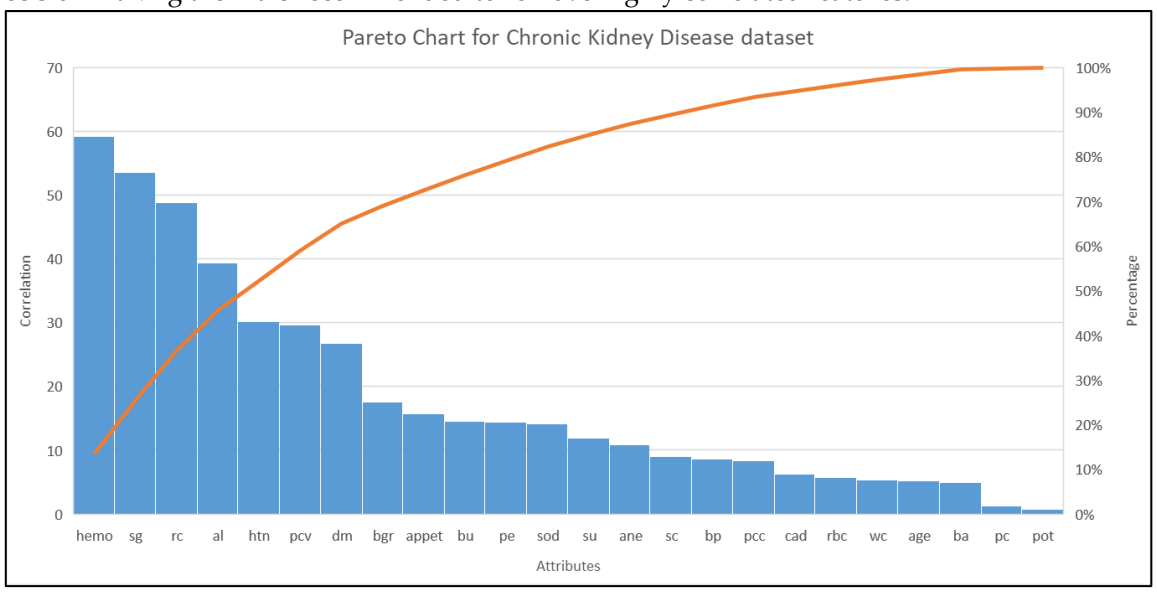

Figure 6. Pareto-chart of features in CKD dataset

4.2 Experimental setup

In this study we have used an ensemble of three decision tree classifiers; CART, Random Forests, and Gradient Boosted Trees. All the experiments are performed on a system having processor AMD Ryzen 3 2200G with 8 GM RAM and 64-bit Windows 10 Enterprise Edition, and RapidMiner Studio 9.6 [36]. The number of trees in both RF and GBT is set to 20, while Gini index is used for CART for the node impurity assessment. The learning rate for GBT is set to 0.10 while maximum depth is set to 2. In case of CART and RF, maximum depth is set at 4 and 7, respectively.

In order to evaluate the efficacy of the proposed approaches we have used a number of classification metrics such as Accuracy, Area under ROC Curve, F1-measure and Sensitivity (recall). The evaluation metrics are computed through the confusion matrix, such as:

True Positive (TP): denotes positive instances predicted as positive

True Negative (TN): denotes negative instances predicted as negative

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

False Positive (FP): denotes negative instances predicted as positive
False Negative (FN): denotes positive instances predicted as negative
Based the aforementioned definitions, the quality metrics of interest are calculated as follows:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (2)$$

$$F1-measure = 2 * \frac{(Recall*Precision)}{(Recall+Precsion)} \quad (3)$$

5. Results

5.1 Baseline Results

In this section we report the results of the baseline models over the full CKD dataset without any feature selection as shown in Table 2. All the results are reported on the 5-fold cross validation.

Table 3. Evaluation results for baseline models

Models	Accuracy	Sensitivity	F-Measure	AUC	Cost Incurred
Decision Tree	92.2 ± 4.8%	90.4 ± 10.5	93.3 ± 4.5	0.0966	475.36
Random Forest	87.7 ± 3.5%	100 ± 0.0%	91.2 ± 2.2%	0.0998	475.36
Gradient Boosted Tree	89.5 ± 5.8%	100 ± 0.0%	92.5 ± 4.0%	1.0	475.36

5.2 Ensemble-1 Results

As mentioned earlier, ensemble-1 is composed of three classifiers ensemble, where each classifier is treated as a scoring function. First we determine the worth of all the attributes across the scoring functions. Afterward, the scores are combined and averaged into a single score. In this regard, we compute individual scores as follows:

- *Decision Tree Score*: Features are scored through decision tree (CART) function. The blue line shows the feature score (F-score) while the orange line denotes accumulated cost (C-score). Both the values are normalized. The point of intersection between F-score and C-score is around feature 5 as shown in Figure 7.

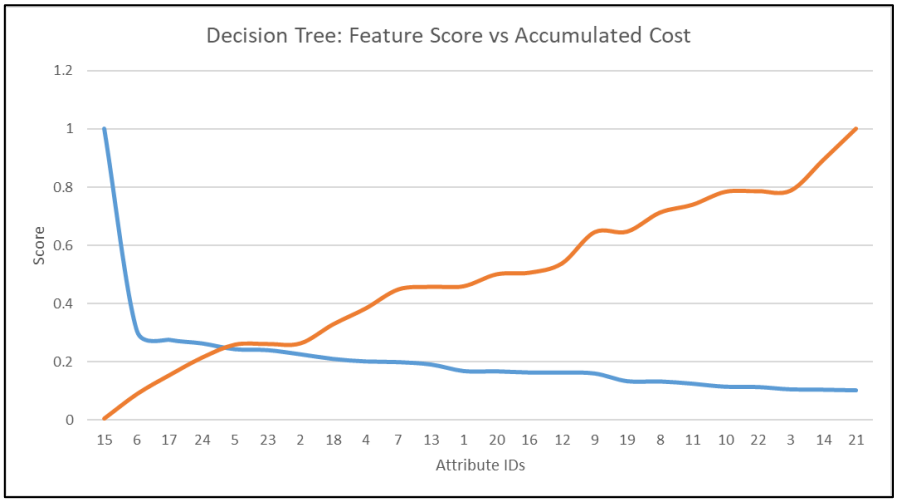

Figure 7. Decision tree based feature scoring

- *Random Forests Score*: The second scoring function is based on random forests. The blue line shows the feature score (F-score) while the orange line denotes accumulated cost (C-score). Both the values are normalized. The point of intersection between F-score and C-score is around feature 9 as shown in Figure 8.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

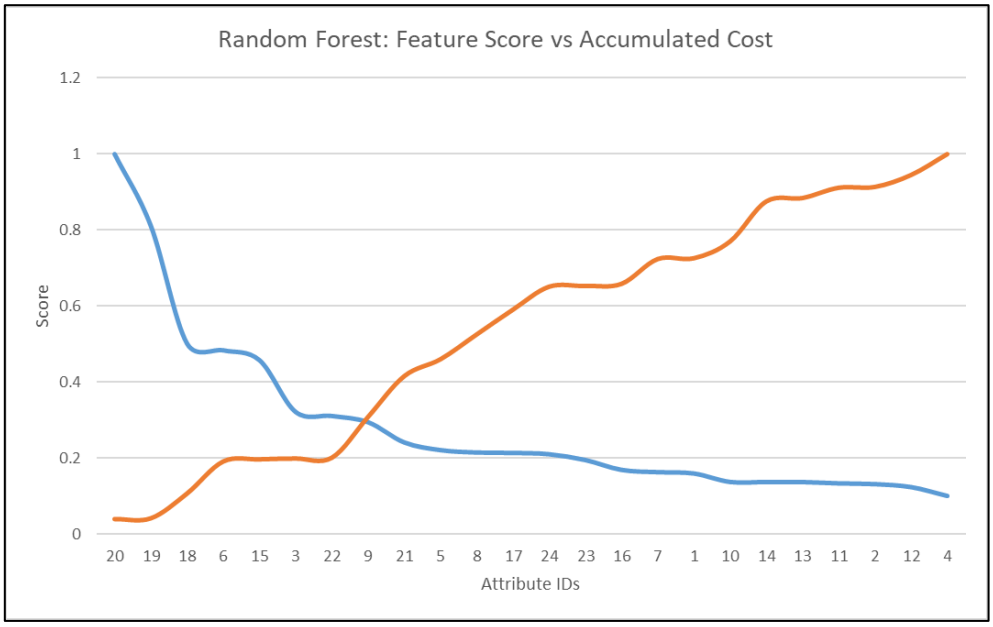

Figure 8. Random forest based feature scoring

- *Gradient Boosted Trees Score*: The second scoring function is based on random forest. The blue line shows the feature score (F-score) while the orange line denotes accumulated cost (C-score). Both the values are normalized. The point of intersection between F-score and C-score is around feature 3 as shown in Figure 9.

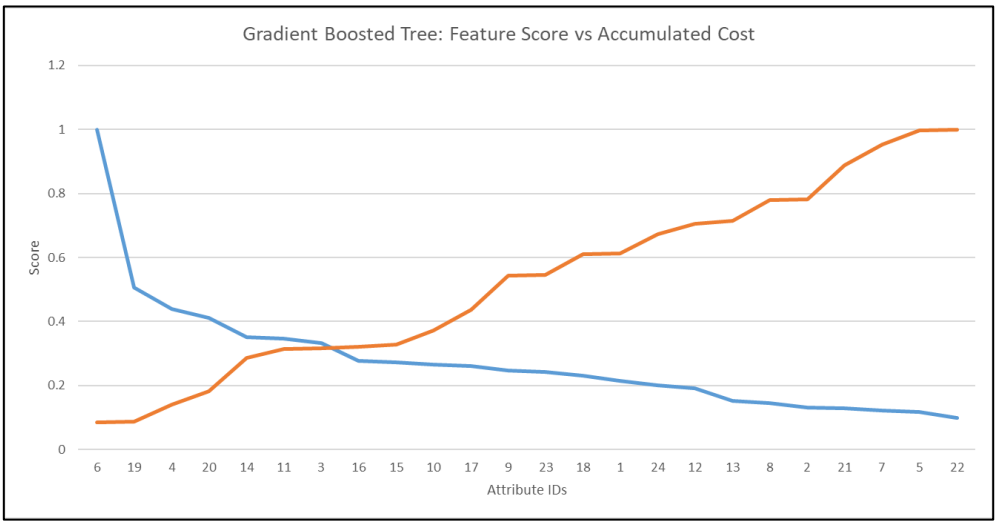

Figure 9. Gradient boosted trees based feature scoring

The resulting selected features collected in list 1, list 2, and list 3 are based on decision tree, random forests and gradient boosted trees, respectively. As it can be seen that there is some variation in the selected features which shows that each scoring function has its own inductive bias while constructing a model as shown in Table 3. Detailed study of the inductive bias of decision tree algorithms is not within the scope of this study.

Table 4. Selected features by individual scoring functions

Scoring function	List	Selected features
Decision Tree	L1	15, 6, 17, 24, 5.
Random Forest	L2	20, 19, 18, 6, 15, 3, 22, 9
Gradient Boosted Trees	L3	6, 19, 4, 20, 14, 11, 3

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

Following are the evaluation results of the aforementioned models based on the selected features. In this regard, decision tree classifier is constructed based on features present in list L1. Likewise, classifier for random forests and gradient boosted trees are built on L2 and L3, respectively, as shown in Table 4.

Table 5. Evaluation results based on selected features

Models	Accuracy	Sensitivity	F-Measure	AUC	Cost Incurred
Decision Tree	79.9 ± 6.4%	98.7 ± 3.0%	86.0 ± 3.9%	0.0958	123.29
Random Forest	96.5 ± 4.8%	100 ± 0.0%	97.4 ± 3.5%	0.099	147.05
Gradient Boosted Tree	93.9 ± 4.9%	100 ± 0.0%	95.4 ± 3.7%	0.0997	150.25

The overall performance of the random forests and gradient boosted trees have increased as indicated by evaluated metrics in Table 2 and Table 4. Although decision tree couldn't improve the accuracy over the full dataset but it has considerably improved the recall. Hence, the automatic threshold mechanism has successfully selected a threshold value which can select a reasonable number of features while also keeping the overall cost of the selected features low. In the following step, we produce a final score of each feature, as discussed in section 3.5.

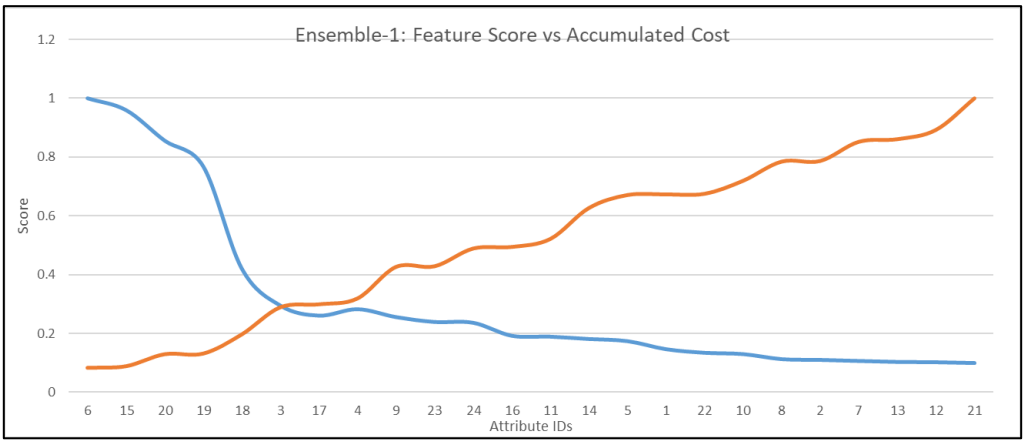

Figure 10. Ensemble-1 based feature scoring

As it can be seen if Figure 10 that the point of intersection is around feature 3. Therefore, all the features starting from 6 leading up to 3 i.e. **6, 15, 20, 19, 18**, and **3**, will be selected. In this regard, Table 5 shows the results of evaluation metrics for feature subset selected by ensemble-1 approach.

Table 6. Ensemble-1 results based on selected features

Models	Accuracy	Sensitivity	F-Measure	AUC	Cost Incurred
Decision Tree	89.4 ± 4.0%	97.1 ± 3.9%	92.1 ± 2.8%	0.0925	
Random Forest	97.4 ± 2.4%	98.6 ± 3.2%	97.9 ± 1.9%	1.0	95.05
Gradient Boosted Tree	95.7 ± 4.3%	100.0 ± 0.0%	96.7 ± 3.2%	1.0	

A cursory glance at Table 4 and Table 5 shows that the proposed ensemble-1 technique has successfully reduced the overall cost while improving the key evaluation metrics.

5.3 Ensemble-2 Results

As mentioned earlier in section 3.5, the ensemble-2 is based on consolidating the ranks acquired from different scoring functions in such a way that 2/3 of the ranked features are retained while rest of the features are discarded.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

Table 7. Feature selected through ensemble-2

Scoring function	List	Selected features	Frequency
Decision Tree	L1	15, 6, 17, 24, 5.	**6**:3, **3**:2, **15**:2, **19**:2, **20**:2, 1:1, 2:1, 4:1, 5:1, … , 24:1
Random Forest	L2	20, 19, 18, 6, 15, 3, 22, 9	
Gradient Boosted Trees	L3	6, 19, 4, 20, 14, 11, 3	
Ensemble-2	*L	6, 15, 20, 19, 3	

In this regard, the Table 7 shows the evaluation metrics applied to models constructed from a selected feature set based on ensemble-2.

Table 8. Ensemble-2 results based on selected features

Models	Accuracy	Sensitivity	F-Measure	AUC	Cost Incurred
Decision Tree	93.9 ± 3.6%	90.4 ± 10.5%	94.7 ± 5.9%	0.0952	
Random Forest	100 ± 0.0%	100 ± 0.0%	100 ± 0.0%	1.0	64.05
Gradient Boosted Tree	100 ± 0.0%	100 ± 0.0%	100 ± 0.0%	1.0	

The proposed ensemble-2 has outperformed ensemble-1 model in terms of number of selected features, accumulated cost incurred and overall performance of the models. Table 8 shows the comparative results of ensemble-1 and ensemble-2 approaches.

Table 9. Comparatively analysis of ensemble-1 and ensemble-2

Models	Accuracy	Sensitivity	F-Measure	AUC	Cost Incurred
Decision Tree	2	1	2	2	
Random Forest	2	2	2	-	2
Gradient Boosted Tree	2	-	2	-	

As it can be seen in Table 8, the ensemble-2 has outperformed ensemble-1 on 9 counts, while ensemble-1 has performed better on 1 count. Although ensemble-2 employs the same individual ranks as that of ensemble-1 but the frequency-based selection, introduced in section 3.5, selects features more conservatively.

5.4 Comparison with other approaches

In this section we compare our results with other models trained on CKD dataset. Please note that the comparative analysis is based on the published literature in which both number of features and their corresponding accuracy are reported.

Table 10. Comparative analysis of proposed technique with others

Study	Accuracy	Sensitivity	F-Measure	AUC	Cost
SVM:BestFit [10]	98.50%	98.50%	98.54%	0.988	-
XGBoost [7]	**100%**	**100%**	**100%**	**1.0**	167.31
SVM-FSS [9]	97.75%	96.40%	98.20%	0.982	-
ANN-FSS [9]	99.75%	99.60%	99.70%	0.999	-
NN:LOG [23]	98.95%	98.44	99.15%	-	141.1
NN:RF [23]	99.75%	99.60%	99.80%	-	141.1
Integrated [23]	99.83%	99.84%	99.86%	-	141.1
Proposed Ensemble-2	**100 ± 0.0%**	**100 ± 0.0%**	**100 ± 0.0%**	**1.0**	**63.05**

A detailed comparison is drawn in Table 9. The bold values denote best performance achieved under a specific criterion. As it can be seen that most of the comparative techniques are primarily tuned to produce models with higher accuracy. Therefore, in terms of accuracy the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 May 2020 doi:10.20944/preprints202005.0458.v1

difference among the models is not much noticeable. But two aspects of the comparison are noticeable i.e. number of selected features and the accumulated cost. In this regard, the proposed ensemble-2 produces a feature subset having both lower number of features and lower accumulated cost, as shown in Table 9 and Figure 11. Hence, the proposed technique successfully optimizes both the criteria while preserving the best overall performance of the model.

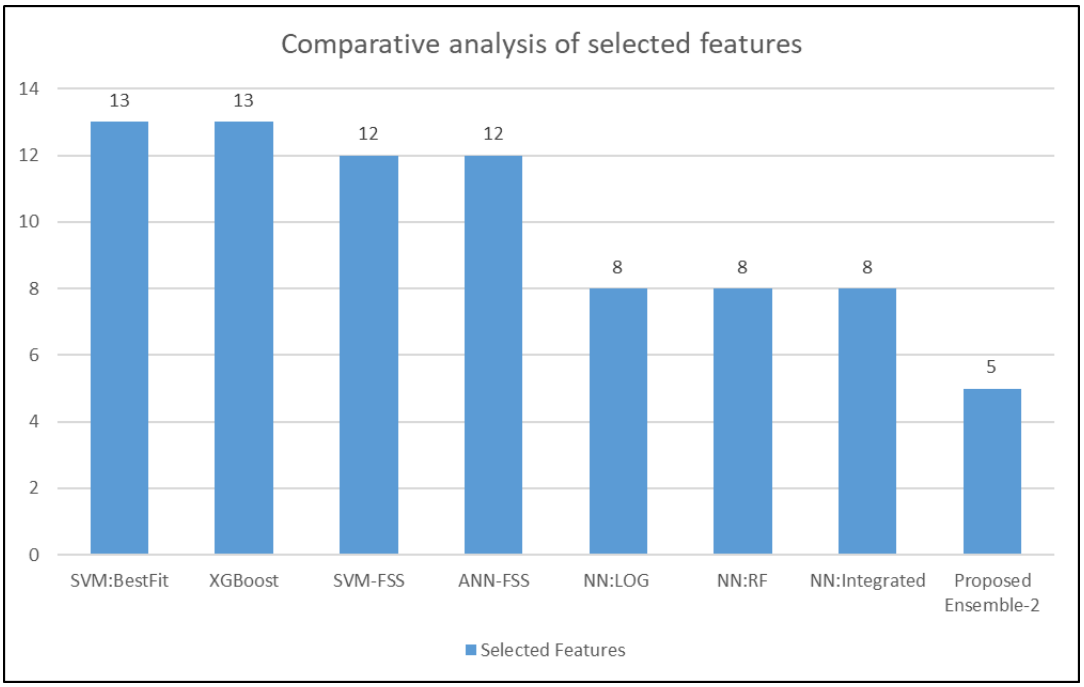

Figure 11. Comparison based on number of features selected

6. Conclusion

Decision-tree based classification models have shown a great promise in the domain of medical diagnosis specially in dealing with structured heterogeneous datasets e.g. electronic medical records for chronic kidney disease patients. This research deals with addressing the applicability concerns of decision tree models when the data acquisition cost is not symmetric i.e. data acquired through different medical tests carry different cost. In this regard, it is shown that the overall cost of an accurate diagnosis can be reduced while preserving and/or enhancing the overall performance of the decision model. We have used chronic kidney disease problem as our case study and investigated model-based feature weighting techniques for ranking. Furthermore, this study has also addressed one of the key challenges in selecting a subset of features from a ranked list. We have introduced two ensemble-approaches which are capable of automatically selecting a threshold value based on the cost-benefit analysis of the dataset. First we have established a baseline result for decision tree based models on the CKD dataset. Afterward, we have ranked features based on individual ranking produced by CART, random forest and gradient boosted trees. The disparate rankings are combined based on a set-theory heuristic. The proposed approach selected a final feature subset by retaining around 1/4 of the original features and preserved the highest F1-score of 100% while the overall cost of data acquisition is also significantly reduced.

This research can be extended in a number of directions such as we have used a classifier-ensemble in order to account for feature interaction. Although this approach has provided promising feature weights but the overall running time of the scoring function can be reduced by employing lightweight filter techniques. Furthermore, cost can be modeled as a multi-objective function along with the error rate and hence a number of candidate solutions can be generated for the decision maker for an informed decision making.

Author Contributions: Syed Imran Ali is principal researcher who conceptualized the idea, designed and implemented the methodology, performed the experiments, and wrote the manuscript. Gwang Hoon Park and

Sungyoung Lee contributed to the funding acquisitions, along with supervising the overall study, provided valuable feedback and reviewed the paper.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-0-01629) supervised by the IITP (Institute for Information & communications Technology Promotion). This work was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00655). This work was also supported by the National Research Foundation (NRF) under the NRF-2016K1A3A7A03951968 and NRF-2019R1A2C2090504.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kidney Disease: Improving Global Outcomes (KDIGO) Transplant Work Group. "KDIGO clinical practice guideline for the care of kidney transplant recipients." American journal of transplantation: official journal of the American Society of Transplantation and the American Society of Transplant Surgeons 9 (2009): S1.
2. Kellum, John A., Norbert Lameire, and KDIGO AKI Guideline Work Group. "Diagnosis, evaluation, and management of acute kidney injury: a KDIGO summary (Part 1)." Critical care 17.1 (2013): 204.
3. Park, Ji In, Hyunjeong Baek, and Hae Hyuk Jung. "Prevalence of chronic kidney disease in korea: The korean national health and nutritional examination survey 2011–2013." Journal of Korean medical science 31.6 (2016): 915-923.
4. Zhang, Qiu-Li, and Dietrich Rothenbacher. "Prevalence of chronic kidney disease in population-based studies: systematic review." BMC public health 8.1 (2008): 117.
5. Moyer, Virginia A. "Screening for chronic kidney disease: US Preventive Services Task Force recommendation statement." Annals of internal medicine 157.8 (2012): 567-570.
6. Sobrinho, Alvaro, et al. "Computer-Aided Diagnosis of Chronic Kidney Disease in Developing Countries: A Comparative Analysis of Machine Learning Techniques." IEEE Access 8 (2020): 25407-25419.
7. Ogunleye, Adeola Azeez, and Wang Qing-Guo. "XGBoost model for chronic kidney disease diagnosis." IEEE/ACM transactions on computational biology and bioinformatics (2019).
8. Khan, Bilal, et al. "An Empirical Evaluation of Machine Learning Techniques for Chronic Kidney Disease Prophecy." IEEE Access 8 (2020): 55012-55022.
9. Almansour, Njoud Abdullah, et al. "Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study." Computers in biology and medicine 109 (2019): 101-111.
10. Polat, Huseyin, Homay Danaei Mehr, and Aydin Cetin. "Diagnosis of chronic kidney disease based on support vector machine by feature selection methods." Journal of medical systems 41.4 (2017): 55.
11. Bolón-Canedo, Verónica, and Amparo Alonso-Betanzos. "Ensembles for feature selection: A review and future trends." Information Fusion 52 (2019): 1-12.
12. Seijo-Pardo, Borja, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. "On developing an automatic threshold applied to feature selection ensembles." Information Fusion 45 (2019): 227-245.
13. Bolón-Canedo, Verónica, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. "An ensemble of filters and classifiers for microarray data classification." Pattern Recognition 45.1 (2012): 531-539.
14. Taradeh, Mohammad, et al. "An evolutionary gravitational search-based feature selection." Information Sciences 497 (2019): 219-239.
15. Lal, Thomas Navin, et al. "Embedded methods." Feature extraction. Springer, Berlin, Heidelberg, 2006. 137-165.
16. Salekin, Asif, and John Stankovic. "Detection of chronic kidney disease and selecting important predictive attributes." 2016 IEEE International Conference on Healthcare Informatics (ICHI). IEEE, 2016.
17. Chen, Zewei, Xin Zhang, and Zhuoyong Zhang. "Clinical risk assessment of patients with chronic kidney disease by using clinical data and multivariate models." International urology and nephrology 48.12 (2016): 2069-2075.
18. Serpen, Alexander Arman. "Diagnosis Rule Extraction from Patient Data for Chronic Kidney Disease Using Machine Learning." International Journal of Biomedical and Clinical Engineering (IJBCE) 5.2 (2016): 64-72.
19. Al-Hyari, Abeer Y., Ahmad M. Al-Taee, and Majid A. Al-Taee. "Clinical decision support system for diagnosis and management of chronic renal failure." 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). IEEE, 2013.
20. Ani, R., et al. "Decision support system for diagnosis and prediction of chronic renal failure using random subspace classification." 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2016.

21. Tazin, Nusrat, Shahed Anzarus Sabab, and Muhammed Tawfiq Chowdhury. "Diagnosis of Chronic Kidney Disease using effective classification and feature selection technique." 2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec). IEEE, 2016.
22. Almansour, Njoud Abdullah, et al. "Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study." Computers in biology and medicine 109 (2019): 101-111.
23. Qin, Jiongming, et al. "A Machine Learning Methodology for Diagnosing Chronic Kidney Disease." IEEE Access 8 (2019): 20991-21002.
24. Ali, Maqbool, et al. "uEFS: An efficient and comprehensive ensemble-based feature selection methodology to select informative features." PloS one 13.8 (2018).
25. Bolón-Canedo, Verónica, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. "A review of feature selection methods on synthetic data." Knowledge and information systems 34.3 (2013): 483-519.
26. Bolón-Canedo, Verónica, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. "Recent advances and emerging challenges of feature selection in the context of big data." Knowledge-Based Systems 86 (2015): 33-45.
27. Seijo-Pardo, Borja, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. "Testing different ensemble configurations for feature selection." Neural Processing Letters 46.3 (2017): 857-880.
28. Khoshgoftaar, Taghi M., Moiz Golawala, and Jason Van Hulse. "An empirical study of learning from imbalanced data using random forest." 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007). Vol. 2. IEEE, 2007.
29. Mejía-Lavalle, Manuel, Enrique Sucar, and Gustavo Arroyo. "Feature selection with a perceptron neural net." Proceedings of the international workshop on feature selection for data mining. 2006.
30. Seijo-Pardo, Borja, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. "Using data complexity measures for thresholding in feature selection rankers." Conference of the Spanish Association for Artificial Intelligence. Springer, Cham, 2016.
31. Willett, Peter. "Combination of similarity rankings using data fusion." Journal of chemical information and modeling 53.1 (2013): 1-10.
32. Seijo-Pardo, Borja, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. "Using a feature selection ensemble on DNA microarray datasets." ESANN. 2016.
33. Chiew, Kang Leng, et al. "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system." Information Sciences 484 (2019): 153-166.
34. Sathe, Saket, and Charu C. Aggarwal. "Nearest neighbor classifiers versus random forests and support vector machines." 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 2019.
35. D. Dua, C. Graff, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2019http://archive.ics.uci.edu/ml.
36. Mierswa, I., and R. Klinkenberg. "RapidMiner Studio (9.2) [Data science, machine learning, predictive analytics]." (2019).