Journal of Computer Science

Original Research Paper

Evaluation of Classification Models for Predicting Mortality Rate Using Thyroid Cancer Data

Norah Saleh Alghamdi

Department of Computer Science, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

Article history
Received: 11-10-2018
Revised: 21-11-2018
Accepted: 16-01-2019

Corresponding Author:
Norah Alghamdi
Department of Computer Science, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia
Email: nosalghamdi@pnu.edu.sa

Abstract: Machine Learning (ML) can potentially enhance predictions in real-life domains. This study presents an evaluation and comparison of different ML methods which can be applied on thyroid cancer dataset, called Prostate, Lung, Colorectal and Ovarian (PLCO), of approximately 155,000 participants with thyroid cancer occurrence and mortality incidence. The ML models are explored for predicting mortality rates of patients with thyroid cancer. These models include the Logistic Regression model (LR), K-Neighbors model (KN), Support Vector Classifier (SVC), Gaussian Naïve Bayes (GNB), decision tree classifier (DT), Random Forest classifier (RF), ada boost classifier (AdaB) and Gradient Boosting classifier (GB). The results reveal that AdaB and GB classifiers have the best performance among the models. The results also show that different predictive models can significantly differ with others in terms of their performance evaluated by various metrics. This study shows that the chosen parameters for classifiers will affect their performance; therefore, it is important to explore and evaluate them before final implementation.

Keywords: Machine Learning, Classification, Thyroid Cancer, Data Mining, Predictive Model, Unsupervised Learning Algorithm, Supervised Algorithm

Introduction

Utilizing data to support decision makers and to create predication models is not a novel method. However, complexities come along with this method due to challenges of managing and analyzing large volumes of data. According to Alpaydin (2009) and Marsland (2015), ML is an analysis technique with a distinctive ability to learn from experience without explicit programming by humans. The two types of ML algorithms are supervised and unsupervised classification. The main objective of the supervised algorithm is to infer a function from labelled training dataset. By adjusting to the training dataset, the most optimal model can be found to predict unknown labels on a test set. On the other hand, unsupervised learning algorithms use training datasets with unlabelled data. They will cluster them based on observed similarities between the data.

Unlike ML algorithms, statistical methods are user-driven. The user should determine variables, functions and the interaction approach. This involvement could affect the results. Due to an automatic ML technique in scanning and analyzing variables, it has been considered a useful tool. It has led to dramatic changes in research and practice in all fields of science (Cukier and Mayer-Schoenberger, 2013). In addition, because of the variation of data from different fields, there is no single model that achieves the highest accuracy of all solutions for all problem types.

Utilizing ML approaches in the prediction of medical cases (i.e., disease, death) has attracted many researchers in the medical domain. In this study, the authors evaluate classification models using PLCO data (https://biometry.nci.nih.gov/cdas/learn/plco/instructions/?type=data) for thyroid cancer occurrence and mortality incidence. This dataset consists of a record for approximately 155,000 participants and 143 features. This large number of participants and features highlights the strengths and uniqueness of the study. To the best of the author's knowledge, this is the first study to evaluate the performance of ML classification models for predicting mortality rates of patients with thyroid cancer using a PLCO dataset. These models are LR, KN, SVC, GNB, DT, RF, AdaB and GB.

Background

The thyroid gland is a critical organ in the human body. This organ, which is centrally located on the human neck, is shaped like a butterfly. The thyroid creates hormones which are sent to the bloodstream to control the human body's functionalities. There are

Science Publications

© 2019 Norah Saleh Alghamdi. This open access article is distributed under a Creative Commons Attribution (CC-BY) 3.0 license.

different diseases related to the thyroid gland. Hypothyroidism occurs when the bloodstream has small amounts of thyroid hormones. In this case, patients normally lose weight. Hyperthyroidism is a case in which high amounts of thyroid hormones exist in the bloodstream. Patients normally suffer from a high heart rate. Thyroid cancer exists when malignant cells are found in the tissues of the thyroid gland. This is most likely caused when patients have been exposed to large amounts of radiation. If symptom occur, including a lump or swelling in the neck, an investigation must be performed through blood tests and scans. The doctor may require a biopsy requiring the removal of a small amount of the gland's tissue. After inserting a fine needle, the removed tissue is checked under a microscope to identify the cancer. Four types of thyroid cancer can be identified: (1) papillary; (2) follicular; (3) medullary; and (4) anaplastic. The prognosis depends on the cancer's speed of growth, stage of the cancer's spread to other parts of the body and the age and health of the patient. Cancer recovery options include four types of treatment based on the cancer: (1) radiation; (2) hormone therapy; (3) chemotherapy; and (4) surgery to remove the cancer through a lobectomy, near-total thyroidectomy, total thyroidectomy and/or lymph node dissection.

Related Works

Applying machine learning techniques in predicting various medical targets using medical datasets attract researchers in the medical domain. The evaluation study in this paper is different from the previous evaluation studies. It evaluates the performance of ML classification models for predicting mortality rates of patients with thyroid cancer using different dataset called PLCO. It is based on supervised ML methods using different classification models including, LR, KN, SVC, GNB, DT, RF, AdaB and GB on a dataset called PLCO. which is highly unbalanced. The designed models predict the mortality incidence for patient with thyroid cancer. In order to evaluate the classification models, different performance metrics is accomplished for the ML models trained with and without sampling.

Some of the previous works focus on designing predictive models for thyroid dysfunctions (i.e. hypothyroidism or hyperthyroidism) using either supervised or unsupervised methods. Saastamoinen and Ketola (2006) studied results established through logical similarity measures in classification, namely C4.5, LDA, MLP and DIMLP. Medical data is retrieved from the UCI ML Repository. Logical comparison measures are used to compare the approach used in this article with the approach proposed by Bologna (2000).

Prerana and Taneja (2015) predicted the analysis of thyroid disease using the back propagation algorithm, a learning algorithm of a neural network. The chosen algorithm consists of a propagation stage and a weight update. The former stage experiences forward and backward propagations to generate input activations and deltas for output and hidden neurons, respectively. The latter stage updates weight by multiplying output delta with input activation to form the gradient, as well as subtracting a ratio of gradient from the weight. The two stages are repeated until the network is satisfactorily performed. Gradient descent and Levenberg algorithms present a difference between Mean Squared Error (MSE) and numbers of epochs, as well as plots of variant gradient errors during training. As a result of the experiment, the Levenberg algorithm achieved a better performance in 59 epochs as compared to a gradient decent approach. Chandel *et al*. (2016) classified thyroid disease using K-NN and Naïve Bayes techniques based on parameters like TSH, T4U and goiter. The researchers aimed to understand and analyze thyroid disease in association with the chosen parameters. The study conducted was on KEEL dataset using RapidMiner software (http://sci2s.ugr.es/keel/dataset/data/classification/thyroid-names.txt). To measure criteria to evaluate the performance of the chosen algorithms, it used Kappa, accuracy and classification errors. The experimental results indicated that the obtained accuracy of K-NN was comparatively higher than Naïve Bayes. Ioniţă and Ioniţă (2016) performed an analysis and comparison between ML models for classification with the following: (1) decision tree; (2) Naïve Bayes; (3) radial basic function network (RBFN); and (4) multilayer perception. The authors targeted hypothyroidism and hyperthyroidism. UCI dataset and Romanian data were used to construct the classifiers and perform a data analysis (http://tiroida.ro/). Two data mining platforms, namely Weka and KNIME, were used to build and test the classification models. A significant accuracy was recorded for the classification models. However, the decision tree beat the other with a 97% accuracy. Ahmed and Soomrani (2016) proposed a Thyroid Diseases Types Diagnostics (TDTD) framework to support physicians. They preprocessed the data using medical data cleaning (MDC) based on the algorithm of Bayesian isotonic regression to fill missing values and eliminate incomplete observations. They trained the data using multi- and binary support vector machine (SVM) algorithms to decide the occurrences of thyroid disease. Shaheed Muhtarma Benazir Bhutto Medical University (SMBBMU) medical and UCI datasets provided information for training and testing (the former), as well as for constructing a decision model to fill the omitted values (the latter). Precision and recall measures were calculated for the performance evaluation. The overall performance of the TDTD system was measured at 95.7%. Chandio *et al.* (2016) proposed an intelligent system for Thyroid Disease Visualization (TVD) to support policymakers by providing them with a significant observation of thyroid diseases between 2013 and 2022. The authors presented thyroid disease incidences from the last 10 years (2002-2012). The system was built on three phases: (1) preprocessing of data; (2) construction of the decision model with Time Series Regression (TSR); and (3) visualization of the outcomes using Q-GIS. The system showed that thyroid disease is more likely to be incremented by 15% for the next 10 years, especially in females between 21 and 30 years of age. The findings focused on the analysis of the data rather than the measurement of the system's performance. Prasad *et al*. (2016) proposed a hybrid system to identify disease progression of the thyroid gland. The authors' system consists of a rough set theory and ML algorithms. A String-Matching System (SMS) was developed to predict the thyroid disease. When SMS fails to achieve the desired goal, Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) optimization are implemented to accomplish high levels of accuracy. The rough set theory calculates missing values of attributes. Therefore, closely correlated diseases of the thyroid can be identified. The expected system outputs may be information, description about the disease, diagnosis, or health advice. The system performance was measured based on the accuracy level when predicting disease levels. The accuracy consistently reached between 99% and 100% optimistic levels. González *et al*. (2017) developed a local laboratory testing kit to classify indeterminate nodules of thyroid glance using an *in vitro* diagnostics (IVD) set. The first phase was a determination of expression data to apply supervised ML methods for training classifiers. The second phase was a prototype assay called the multiplexed-qPCR IVD, which was developed after selecting the best classifier. An optimal performance was achieved by the best classifier when testing 10 genes to register 81% specificity and 93% sensitivity.

Others focus on designing the predictive models for thyroid cancer using either supervised or semi-supervised methods. Upadhayay *et al*. (2013) implemented the C4.5 and C5.0 data mining classification algorithms. The authors observed the large size of the C4.5 tree in comparison to the C5.0 tree. The C5.0 algorithm was superior to the C4.5 in the following ways: (i) generated more accurate rules after the pruning process, (ii) consumed less running time and (iii) generated six rule sets with a 95% confidence level. Kongburan *et al*. (2016) used text mining technology to extract hidden significance from vast unstructured text of biomedical information published in PubMed in reasonable time. They utilized a text mining technique called, Named Entity Recognition (NER) to discover information from a custom-made corpus called a thyroid cancer intervention corpus. Steps were conducted to construct the corpus by retrieving all abstract files containing INTERVENTION and DISEASE from PubMed, tokenizing the data, manually annotating each token based on three classes (i.e., INTERVENTION, DISEASE and O) and training the NER model. The experiments showed that NER was not able to classify INTERVENTION from O. It could, however, classify DISEASE from O. An excellent classification was achieved by NER when classifying INTERVENTION and DISEASE classes. Using a multiclassifier technique, Jothi and Rajam (2017) proposed a Computer Aid Diagnosis (CAD) platform to distinguish between histopathology images for a normal thyroid and papillary thyroid cancer. The platform consisted of three phases. First, the extraction of dark blue nuclei and orphan annie-eye nuclei are taken from binary images. These images segment the particle swarm optimization-based Otsu's multilevel thresholding (PSO-O) into multiple partitioned binary images. Then, texture features and morphology are extracted from the partitioned images. During the last stage, the classification distinguishes between affected thyroid by papillary cancer and unaffected by utilizing individual or multiple classifiers. The maximum accuracy of 99.54 is achieved by combining SVM-L, SVM-Q, SVM-RBF and CMR, as well as by combining SVM-L, SVM-Q and CMR. Table 1 shows comparisons of related work.

Table 1: Comparison of existing work

Research - Dataset	Disease	Algorithms	Performance test	Results
Saastamoinen and Ketola, 2006, UCI ML Rep	Hypo/ Hyper-thyroidism	Classification, C4.5, LDA, MLP, DIMLP	Combined (Lukasiewicz and Shweizer & Sklar), Kleene-Dienes, and Reichenbach	Maximum, optimal and variant performance depends on dependent on used data set
Upadhayay and *et al*., 2013, UCI ML Rep	Thyroid Cancer	Decision Tree, C4.5, C5.0	Accuracy, speed, memory, smaller DT, boosting, weighting	C5.0 Tree generated more accurate rule set with smaller running time
Prerana and Taneja, 2015, UCI ML Rep	Hypo/ Hyper-thyroidism	Prediction- ANN (gradient descent, Levenberg)	Accuracy-Number of epochs-MSE-Gradient values	Levenberg has shown a better training compared to gradient decent
Chandel *et al*., 2016, KEEL, Text	Hypo/ Hyper-thyroidism	KNN, SVM and Naive Bayes	Accuracy, Kappa and Classification error	The accuracy of KNN is better than NB
Ioniță and Ioniță, 2016, UCI ML Rep.	Hypo/ Hyper-thyroidism	ANN (MLP- RBF), Naïve Bays, DT (ID3, CART, C4.5)	Accuracy (Precision, Recall, Sensitivity, Specify, F-measures)	The best model being Decision Tree with 96.5% accuracy
Kongburan *et al*., 2016, Intervention Corpus	Thyroid Cancer	Text-mining (NER)	Accuracy (precision, recall, F1-score)	Reasonable performance in constructing of the corpus
Ahmed, 2016, SMBBMU dataset and UCI dataset	Hypo/ Hyper-thyroidism	Bayesian isotonic regression- multi and binary SVM	Confusion matrix, precision and recall measures	Overall classification of the system was measured as 95.7%
Chandio *et al*., 2016, Ten year real-world datasets	Hypo/ Hyper-thyroidism	Time Series Regression	None	Focusing on the findings of the plot and graph analysis
Prasad *et al*., 2016, UCI data	Hypo/ Hyper-thyroidism	Un-supervised (SMS, ABC, PSO)	Accuracy	Accuracy reached at most 99% to 100 %.
González *et al*. 2017, Gene data	Thyroid cytology	a multiplexed-qPCR IVD prototype	Sensitivity, specificity	sensitivity 93% and specificity 81%
Jothi and Rajam, 2017, annie-eye nuclei and the dark blue nuclei	Normal thyroid and papillary thyroid cancer	SVM- KB- KNN- CMR	Sensitivity, specificity, accuracy	99.54 is achieved by combining SVM-L, SVM-Q, SVM-RBF and CMR

Method

In this study, the authors use a comprehensive thyroid dataset consisting of the PLCO data for thyroid cancer occurrence and mortality incidence. The dataset contains the data of 155,000 participants.

There are 19 different classes of 143 variables containing data about the diagnosis of patients. Classes include trial entry (patient information), exit, cancer diagnoses (target variables), cancer characteristics, mortality status, cause of death and cause of death. The authors analyze the trial entry class because it contains data on patients and mortality status. After exploring the trial entry, the target variable for the prediction of thyroid cancer is "thyd_cancer." The number of patients with thyroid cancer is 248; 154,649 patients are without cancer. Figure 1 shows the phases of our experiments.

Data Preprocessing

Out-of-range values and/or missing values may be collected during the data gathering phase. Excessive amounts of noisy or missing data makes it difficult to discover knowledge during the training phase.

Data preprocessing is important. However, processing, cleaning, normalization, transformation, feature extraction of data consume a considerable amount of time. If the data is incorrectly preprocessed, the analysis and produced prediction results may be misleading. The following preprocessing step was conducted in this study. Manipulating missing values, for example, "thyd_fh_age" refers to the age of the youngest relative with thyroid cancer and "thyd_type" refers to thyroid cancer type, has been performed using a missing data imputation model for variables containing 80% to 90% of missing values. The model is based on computing mean values for the missing values.

Feature Selection

In practice, incorporating irrelevant variables in ML predictive models can cause unnecessary complexity in the generated model. To develop a better model, an "extra tree classifier" is utilized in this study to select feature importance (Geurts *et al.*, 2006).

The Gini algorithm is used to calculate feature importance. In this algorithm, across all trees containing the feature, the sum is calculated over the total number of splits averaged by the total number of splitting samples. It assists the feature importance of each variable. These variables are then ranked according to importance. Figure 2 shows the top 10 variables selected based on feature importance for the models. These include "thyd_is_first_dx" (if thyroid cancer was the first diagnosed cancer) and "mortality_exitstat" (patient exit stage status for the case of mortality to "dead" or "alive").

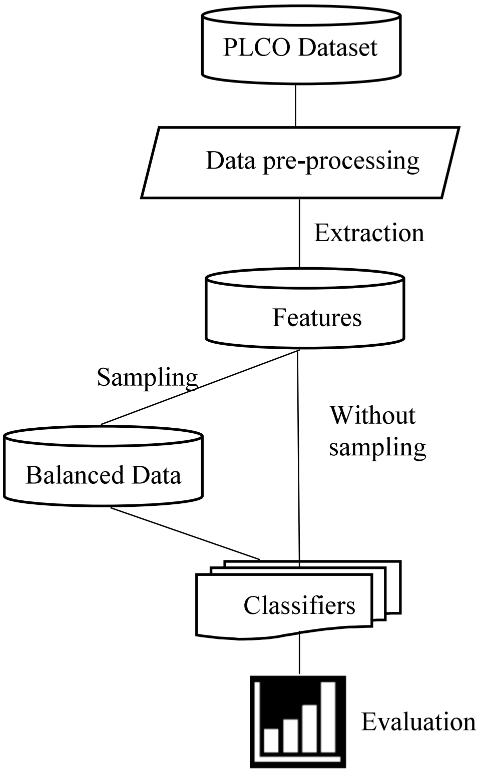


Fig. 1: Experimental phases

Unbalancing Data

A main challenge of the PLCO dataset is imbalanced data because 154,649 patients are not diagnosed with cancer (99.83%) and 248 patients are diagnosed with cancer (0.16%). Typically, this indicates that the dataset classes are not equally represented. In turn, misleading prediction accuracy is more likely to occur (Pazzani *et al.*, 1994). In other words, if the number of instances in each class is minimally different, it will not affect the analysis results. According to the authors' experiments, Fig. 3 shows the confusion matrix and Receiver Operating Characteristic (ROC) curve for the Naïve Bayes classification model. A100% accuracy is achieved due to the imbalanced data because 99.83% of the dataset is a "no cancer" class. The model will always predict the class "no cancer" due to its majority. Therefore, it achieves high accuracy.

There are different approaches to treating an imbalanced dataset. One approach is to perform data sampling through over-sampling or under-sampling. The over-sampling approach duplicates variables to the minority class to balance the dataset. The under-sampling approach drops the variables of the majority class to balance the dataset. According to Drummond and Holte (2003) and the authors' experiments with both approaches, under-sampling is more effective because it generates a reasonable change in class distribution and performance.

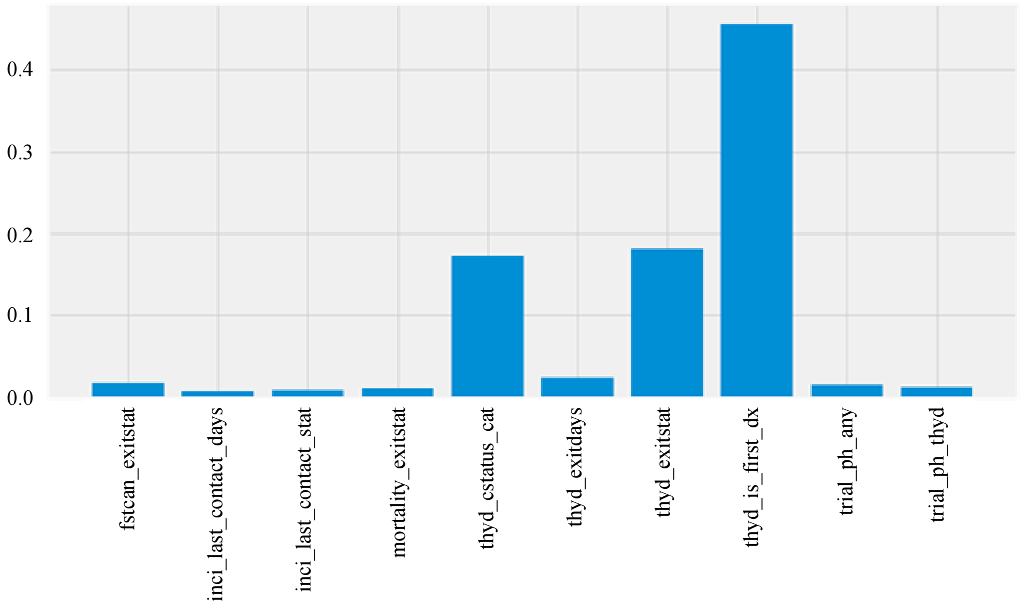


Fig. 2: Variables selected based on feature importance in building classification models

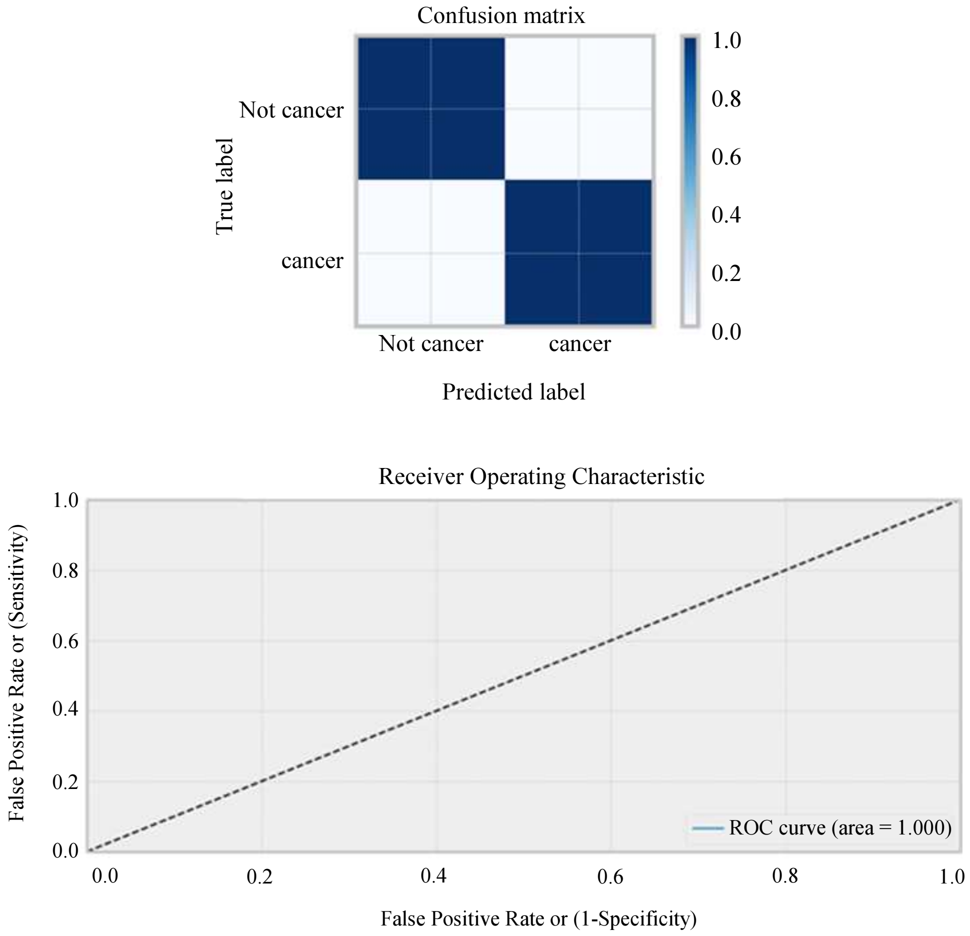


Fig. 3: Overfitting data due to data imbalance

However, in the case of over-sampling, little or no change is generated in class distribution and performance. The authors use under-sampling, particularly the NearMiss algorithm, to handle imbalanced classes. This algorithm chooses samples from the majority class based on computing the average distance where the nearest neighbors are the smallest.

Classification Models

The authors' experiments are performed on the following ML classification techniques:

LR: This predictive model employs to two levels of the dependent variable with two possible values labelled "1" and "0." In fact, the model can cover a binary dependant variable like categorical variables with more than binary values (Cox, 1958). The authors apply a prepacked Python library for the implementation of the LR model. They vary the solver and penalty parameters. They use "newton-cg" and "lbfgs" solvers, which only support L2 penalty. The "liblinear" solver can support either L1 or L2 penalty.

KN: This memorizes the training set to predict the label based on other labels of nearest neighbors in the training set. Usually, the k closest training variables in the feature space are provided to the model as inputs to generate a class membership as output. The classification of an object is based on a majority vote of its neighbours (Altman, 1992). The authors apply a prepacked Python ML "sklearn" library for the implementation of the KN classifier. They fix the number of neighbors to be used by the model queries to two. They use BallTree, KDTree and brute-force search algorithms to calculate the closest neighbors.

SVC: This creates a hyperplane or a set of it in infinite-dimensional space to be used in the classification. The hyperplane with the largest distance to the closest point of training data in any class achieve better separation. The authors test the model using the Python implementation based on "libsvm" (Cortes and Vapnik, 1995). They specify a different kernel to be used in the algorithm, including "linear," "rbf," and "poly." The penalty, gamma, epsilon and degree parameters are varied by different given values in the testing model.

GNB: This classification model consists of a group of Naïve "probabilistic classifiers" using Bayes' theorem. This theorem is based on Naïve assumptions between the features. These classifiers are highly scalable. They require several parameters that are linear with the number of features when creating a learning model. The authors apply the implemented ML Python library for "GaussianNB" classification.

DT: This model uses a tree-like graph as a predictive model to infer from observations about data conclusions about the data's target variable. If the target variable takes a discrete set of values, the tree model is used for classification. In this tree-like graph model, leaf nodes represent class labels. Conjunctions of features leading to these labels are represented by tree branches (Apté and Weiss, 1997). In the authors' testing model implementing Python, different values of the following parameters are used: "criterion," "max_depth," and "max_features." The criterion measures function for the quality of a split. The authors use "entropy" (information gain) and "gini" (Gini impurity) as two supported criteria. The authors use "max_depth" to identify the maximum depth of the tree (3, 5 and 7). To gain the best split, the authors determine the number of features to be considered during the split.

RF: This model forms many decision trees during the training or testing phases, as well as outputting a class that is the mode of the classes (classification). The model's learning process is based on decision rules from the data features. This model overcomes problems of the decision trees, including over-fitting issues (Ho, 1995). The authors test this model using Python. The authors use 200 trees in the forest. They vary the number of features to use when searching for the best split (0, 3 and 5).

AdaB: This adaptive model can be adapted with other types of ML models to boost performance. A weighted sum of the boosted classifier's output is combined with the output of the weak algorithm. AdaBoost is sensitive to noisy data (Freund and Schapire, 1996). The authors' test using Python varies the parameters "n_estimators" (100 and 250) and "learning_rate" (0.1, 1 and 10). The former parameter represents the maximum number of estimators when terminating the boosting process. The latter parameter shrinks the influence of each classifier by the provided learning rate.

GB: This creates an additional model in a forward stage-wise fashion by optimizing arbitrary differentiable loss functions (Breiman, 1997). In each phase, decision trees adjust to the negative gradient differentiable loss function. By implementing Python for GB, the authors test the model performance when changing the values of "n_estimators" parameters (50, 100 and 150) and "learning_rate" parameters (0.01, 0.1 and 1). The "n_estimators" parameter represents the number of boosting stages to be performed. Better performance can be achieved when a large number of gradient boosting is performed.

Evaluation Metrics

To evaluate ML models, the authors apply the k-fold cross-validation evaluation approach in which the data is equality partitioned into k-subsets. Next, k-time test operations are performed. This method reduces the variance in the model performance over k-partitions. Partitioning steps are computed using the following metrics.

Accuracy is the fraction of correct predictions of the authors' model. It is computed considering the positives and negatives of the prediction:

$$Accuracy = TP + TN / TP + TN + FP + FN$$

Precision denotes the ratio of the classifier's ability to label tuples as positive that is not negative:

$$\text{Precision} = TP / TP + FP$$

Recall is the ratio of the classifier's ability to find all the positive tuples:

$$\text{Recall} = TP / TP + FN$$

F-beta score represents the weighted harmonic mean of precision. Recall True Positive (TP) refers to the number of patients without thyroid cancer who are classified as without thyroid cancer. False Positive (FP) refers to the number of patients without thyroid cancer who are classified as with thyroid cancer. On the other hand, False Negative (FN) refers to the number of patients with thyroid cancer who are classified as patients without thyroid cancer. True Negative (TN) refers to the number of patients with thyroid cancer who are classified as patients with thyroid cancer.

ROC curve is a graph plotting the model performance at all classification thresholds. It plots the following parameters:

$$\text{True positive rate} = TP / TP + FN$$
$$\text{False positive rate} = FP / FP + TN$$

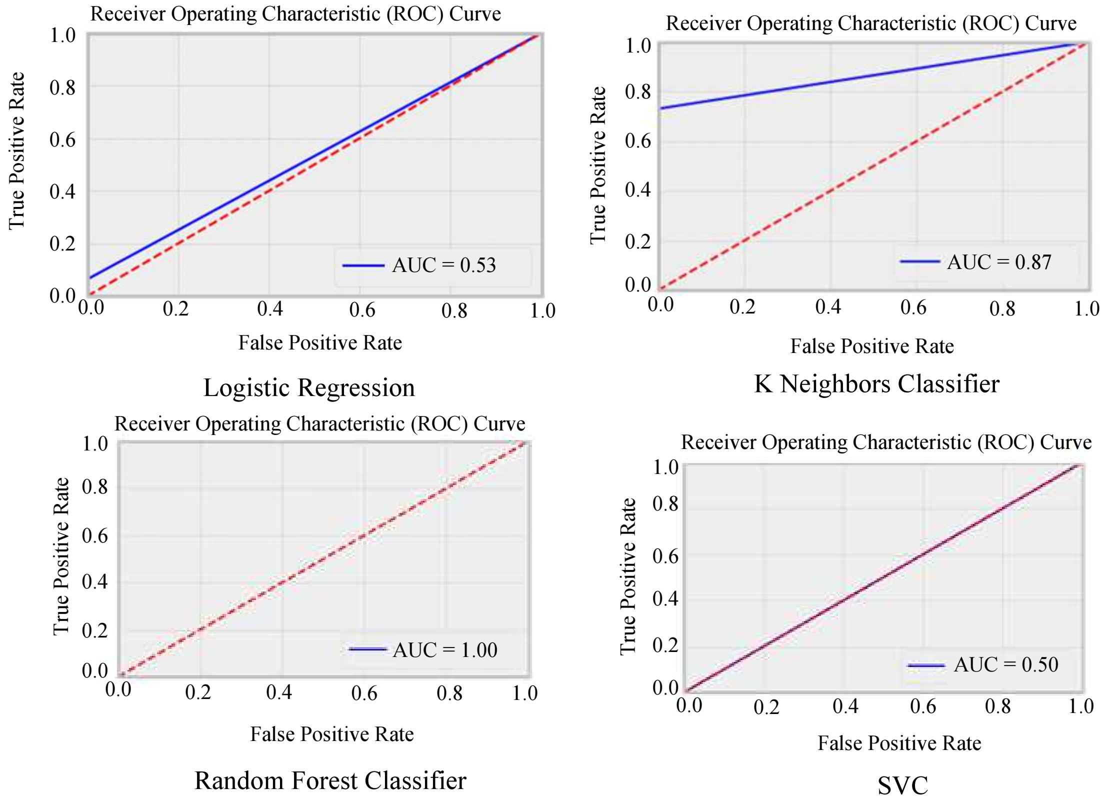
To calculate the points in the ROC curve, the area under the ROC curve (AUC) measurement is utilized. This is a sorting-based algorithm that aggregates the performance measurements of all possible classification thresholds.

Results and Discussion

This part of the study presents performance results of the models showing the discussion of the outcomes. The authors first compare the performance of the classification models with and without the sampling technique. Next, they present a comparison of the classification model performance using different parameters. Finally, the authors compute the evaluation metrics using cross-validation score, accuracy, precision, recall and F-score for the best classifiers.

As mentioned, the PLCO dataset has high imbalanced data because 99.83% of its data is the "No Cancer" class. This leads to inaccurate prediction training. In fact, the training process will rely on the majority of the class. To overcome this issue, the authors applied the under-sampling technique. They evaluate prediction models with and without applying the sampling approach.

Figure 4 shows the performance of the classification models, including ROC and AUC, without data sampling. Figure 5 shows the performance of the classification models, including ROC and AUC, with data sampling. In general, DT, AdaB and GB achieve the best performance (100%) with and without sampling.

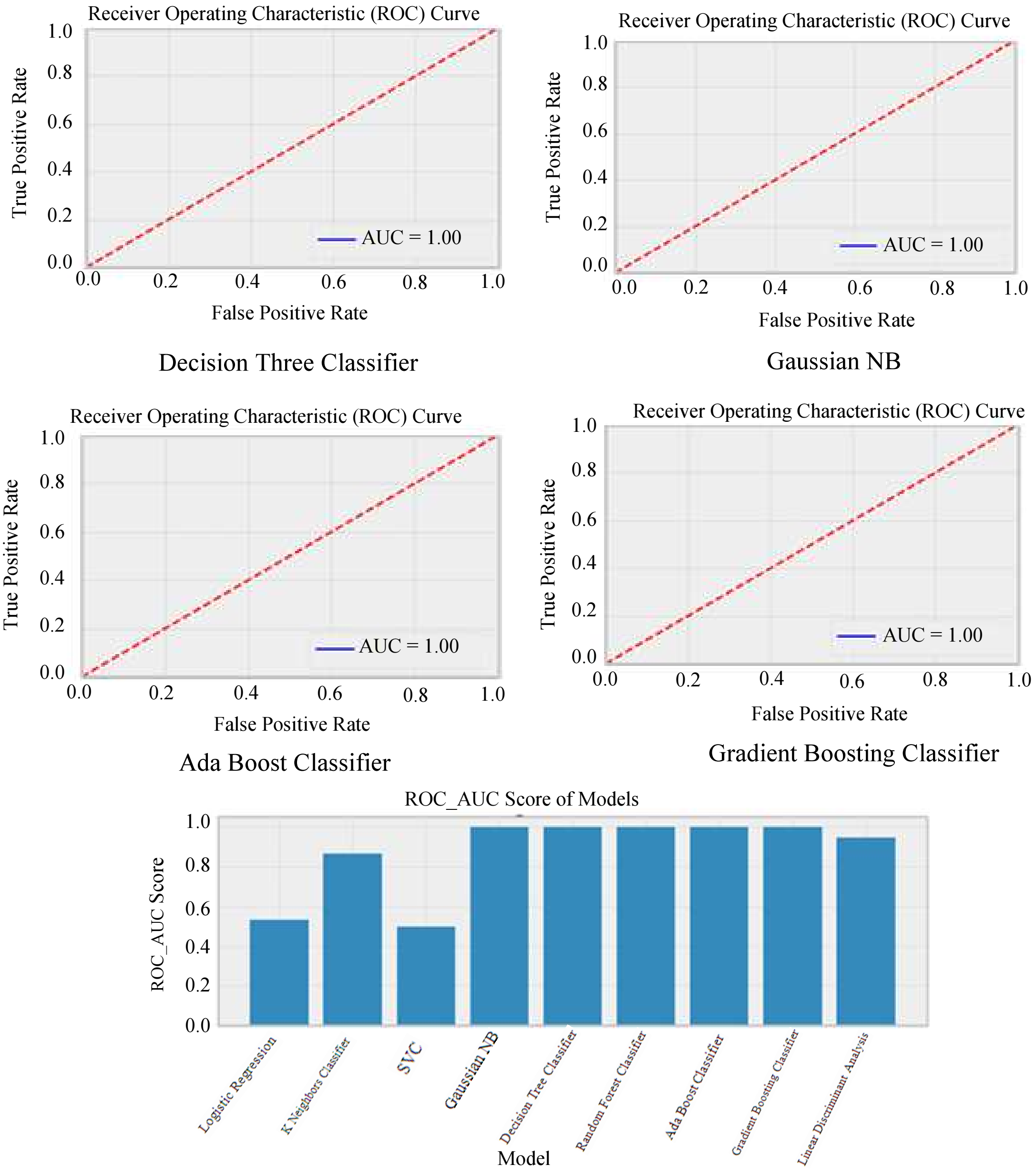


Logistic Regression

K Neighbors Classifier

Random Forest Classifier

SVC



Decision Three Classifier

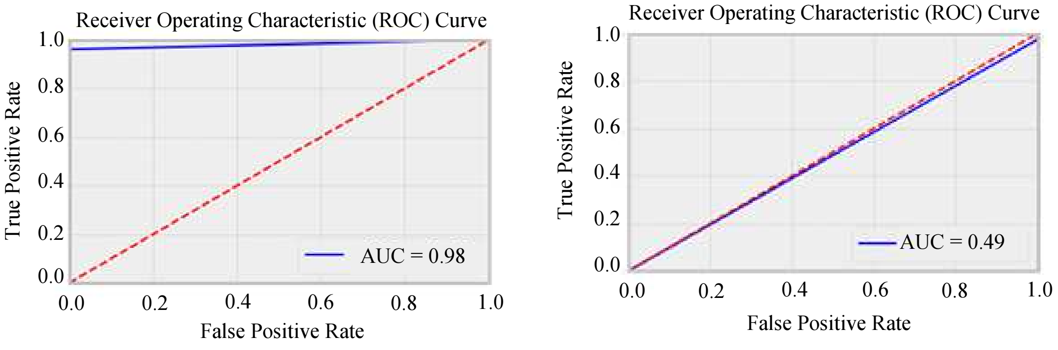
Gaussian NB

Ada Boost Classifier

Gradient Boosting Classifier

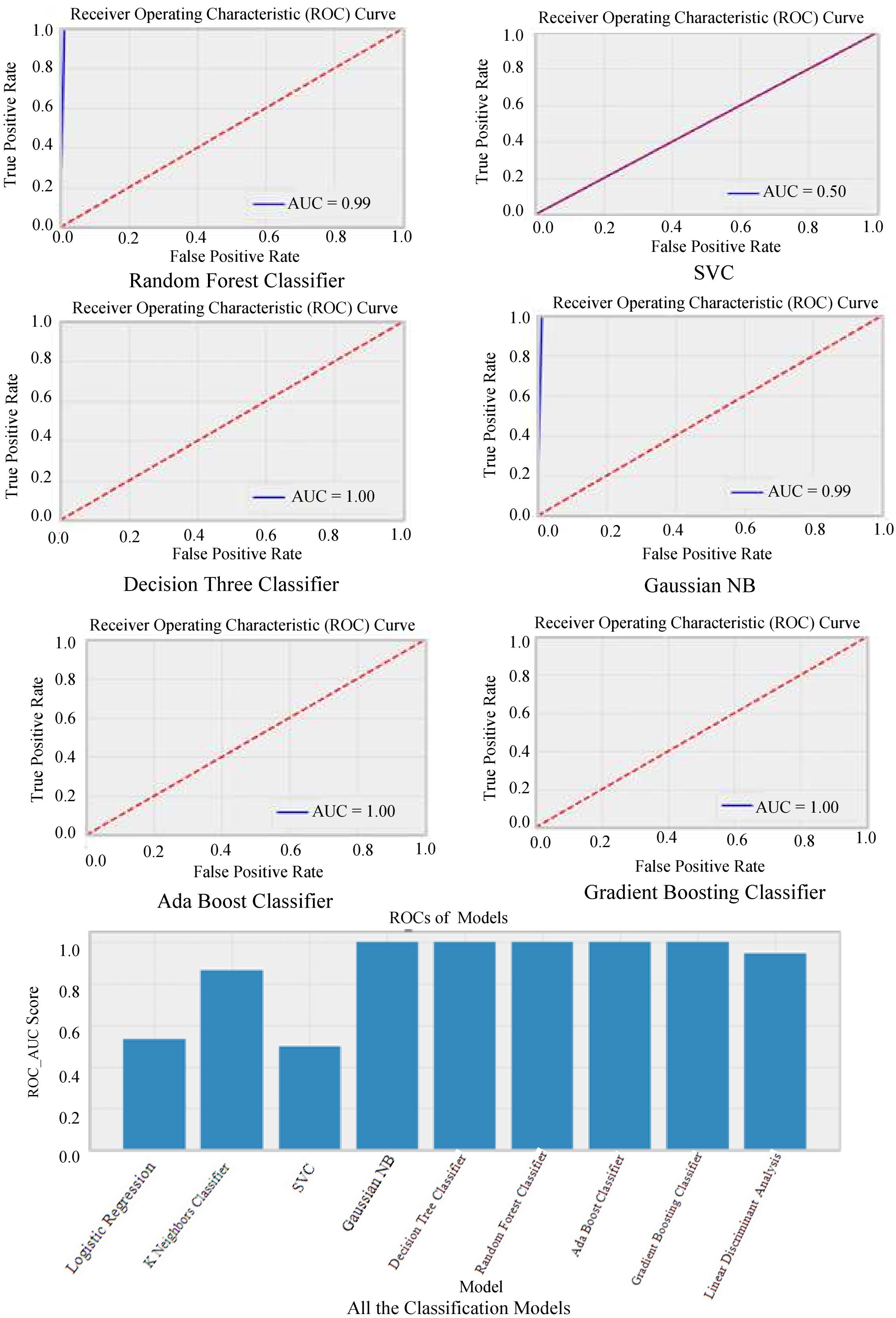
All the Classification Models

Fig. 4: Performance of the classification models without applying the sampling approach



Logistic Regression

K Neighbors Classifier



Random Forest Classifier

SVC

Decision Three Classifier

Gaussian NB

Ada Boost Classifier

Gradient Boosting Classifier

All the Classification Models

Fig. 5: Performance of the classification models with applying the sampling approach

Table 2: Comparing multiple models with different parameters

Models	Parameters	Performance:5_fold Cross-validation
DT	criterion = entropy, max_depth = 3, max_features = 'auto'	0.985
	criterion = gini, max_depth = 3, max_features = 'sqrt'	0.994
	criterion = 'entropy,' max_depth = 5, max_features = 'auto'	0.994
	criterion = 'gini,' max_depth = 5, max_features = 'sqrt'	0.997
	criterion = 'entropy,' max_depth = 7, max_features = 'auto'	0.988
	criterion = 'gini,' max_depth = 7, max_features = 'sqrt'	0.982
SVC	kernel = linear, C = 100	0.994
	kernel = linear, C = 10	0.994
	kernel = linear, C = 1	0.994
	kernel = rbf, gamma = 0.1, C = 10, epsilon = 0.4	0.532
	kernel = rbf, gamma = 0.01, C = 1, epsilon = 0.3	0.744
	kernel = rbf, gamma = 0.001, C = .1, epsilon = 0.2	0.949
	kernel = poly, gamma = 0.1, C = 10, epsilon = 0.4	0.994
	kernel = poly, gamma = 0.01, C = 1, epsilon = 0.4	0.994
	kernel = poly, gamma = 0.01, C = 1, epsilon = 0.4	0.994
RF	max_depth = None, random_state = 0, n_estimators = 200'	1.000
	max_depth = 3, random_state = 0, n_estimators = 200'	1.000
	max_depth = 5, random_state = 0, n_estimators = 200'	1.000
AdaB	n_estimators = 100, learning_rate = 10'	1.000
	n_estimators = 100, learning_rate = 1'	0.997
	n_estimators = 250, learning_rate = 10'	1.000
	n_estimators = 250, learning_rate = 1'	0.997
GB	n_estimators = 50, learning_rate = 1'	0.997
	n_estimators = 50, learning_rate = 0.1'	1.000
	n_estimators = 50, learning_rate = 0.01'	1.000
	n_estimators = 100, learning_rate = 1'	1.000
	n_estimators = 100, learning_rate = 0.1'	1.000
	n_estimators = 100, learning_rate = 0.01'	1.000
	n_estimators = 150, learning_rate = 1'	1.000
	n_estimators = 150, learning_rate = 0.1'	1.000
	n_estimators = 150, learning_rate = 0.01'	1.000
LR	penalty = 'l1,' solver = 'liblinear'	0.994
	penalty = 'l2,' solver = 'newton-cg'	0.991
	penalty = 'l2,' solver = 'lbfgs'	0.991
	penalty = 'l2', solver = 'liblinear'	0.991
KN	n_neighbors = 2, algorithm = 'auto'	0.985
	n_neighbors = 2, algorithm = 'ball_tree'	0.985
	n_neighbors = 2, algorithm = 'kd_tree'	0.985
	n_neighbors = 2, algorithm = 'brute'	0.985

The worst performance is gained by LR without sampling (53%). Yet a significant improvement is noticed with the model after data sampling (98%). The sampling technique improves the performance of all classification models except KN, RF and GNB. Their performance is reduced after the sampling process. In fact, the reduction of the performance is not that significant. The authors implement the classification models using Python. They change the models' parameters and compare performance based on the cross-validation scores (Table 2).

For DT model, they used two metrics for splitting a tree "gini" and "entropy." They found the former algorithm outperforms the latter except when the maximum depth is increased (for example, the case of "max_depth" is seven).

The results of SVC are based on different kernels, including "linear," "poly," "rbf," and other complexity parameters. The first two kernels achieve the best performance ("0.994"). It can be noticed that the changing of coefficient parameter C when using these two kernels does not influence performance. On the other hand, the empirical results of implementing SVC with the kernel "rbf" reveal that the optimal choice of epsilon with the changing of coefficient parameter C has a clear improvement on the performance of the model.

In the model RF, the accuracy performance is high if the authors fix the number of trees with variant depth.

AdaB performs well in all cases when changing the number of estimators of the boosting process, as well as when changing the learning rate. However, there is a slight performance improvement when the learning rate is increased.

Table 3: Best classifiers after comparing multiple models with different parameters

Method	Cross Validation Score	Accuracy	Precision	Recall	F_Score
DT	0.994	0.8462	1.00000	0.6923	0.81820
SVC (linear)	0.994	0.9936	0.98730	1.0000	0.99360
SVC (rbf)	0.532	0.5000	0.00000	0.0000	0.00000
SVC (poly)	0.994	0.9936	0.98734	1.0000	0.99363
GNB	0.997	0.9936	1.00000	0.9872	0.99350
RF	1.000	0.6474	1.00000	0.2949	0.45540
AdaB	1.000	1.0000	1.00000	1.0000	1.00000
GB	1.000	1.0000	1.00000	1.0000	1.00000
LR (liblinear)	0.994	1.0000	1.00000	1.0000	1.00000
LR (newton-cg)	0.991	0.9870	0.97400	1.0000	0.98700
KN	0.985	0.4870	0.00000	0.0000	0.00000

The model GB accomplishes the best performance for all cases of changing the number of gradient boosting process and learning rate.

The authors noted that LR performs worst without sampling. It records a significant improvement with sampling. Based on this observation, they apply the LR model with balanced data and test different optimizers including "blinear," "newton-cg," and "lbfgs." Implementing "blinear" optimizer achieves the highest performance with penalty L1. The variation in performances among the applied optimizers is not big. The implementations of LR perform well even with the changing of the solvers and the penalties.

Regarding the model KN, the authors use two neighbors and different algorithms to compute the nearest neighbors. The authors found that the model performs the same with other searching algorithms because the number of neighbors is fixed.

After evaluating the classifiers using different parameters, the authors select the best classifiers and compute the following evaluation metrics: cross-validation score, accuracy, precision, recall and F-score for the best classifiers (see Table 3). High accuracy indicates the high ratio of correct predictions of the classification models. RF, AdaB and GB achieve the best performance regarding accuracy (accuracy = 1.0). On the other hand, SVC using the kernel "rbf" suffers from generating wrong predictions. Thus, the accuracy is affected negatively. By using other kernels, such as "linear" and "poly," the accuracy performance is significantly improved for SVC. The RF and DT classifiers perform well by identifying TP cases correctly. This achieves high precision (see precision = 1.0 for both in Table 3). However, they miss several positive cases leading to low recall (see RF-recall = 0.295 and DT-recall = 0.692 in Table 3). KN and SVC (rbf) classifiers have zero precision, which indicates a large number of FP. They also have zero recall, which indicates many FN. AdaB and GB record the best performance among other classifiers by achieving high precision and recall (equal to one).

Conclusion

In different domains (i.e., economy, education, healthcare and medicine), ML methods approve their ability of prediction. In this article, the authors evaluate the performance of ML classification models for predicting mortality rates of patients with thyroid cancer using PLCO data. The result revealed that:

- The performance of the classification models varies when evaluating different metrics.
- The chosen parameters for classifiers will affect their performance. It is important to explore and evaluate them before final implementation.
- When designing an efficient predictive model, the domain problem and its dataset must be carefully investigated, evaluated and modelled.

The authors conclude that the result of this study meets with the nature of the ML process, which requires variant experiments and exploration to identify the best design of the classification model.

Acknowledgement

This work was supported by the Research Center of the Faculty of Computer and Information Sciences, Princess Nourah bint Abdulrahman University. The authors offer their sincere thanks to the deanship of scientific research at Princess Nourah bint Abdulrahman University for their support. The authors are grateful for the support in conducting this successful research.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and there are no ethical issues involved.

References

Ahmed, J. and M.A.R. Soomrani, 2016. TDTD: Thyroid disease type diagnostics. Proceedings of the International Conference on Intelligent Systems Engineering (ICISE), New York, IEEE, pp: 44-50. DOI: 10.1109/INTELSE.2016.7475160

Alpaydin, E., 2009. Introduction to ML. Cambridge, MA: MIT Press.

Altman, N.S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Ame. Statistician, 46: 175-185. DOI: 10.1080/00031305.1992.10475879

Apté, C. and S. Weiss, 1997. Data mining with decision trees and decision rules. Future Generation Computer Systems, 13: 197-210. DOI: 10.1016/S0167-739X(97)00021-6

Bologna, G., 2000. A study on rule extraction from neural networks applied to medical databases. Proceedings of the 4th European Principles and Practice of Knowledge Discovery in Databases, Layon, France.

Breiman, L., 1997. Arcing the edge [Technical report 486]. Statistics Department, University of California at Berkeley.

Chandel, K., V. Kunwar, S. Sabitha, T. Choudhury and S. Mukherjee, 2016. A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques. CSI Transactions ICT, 4: 313-319. DOI: 10.1007/s40012-016-0100-5

Chandio, J.A., A. Sahito, M.A.R. Soomrani and S.A. Abbasi, 2016. TDV: Intelligent system for thyroid disease visualization. Proceedings of the International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), New York, NY: IEEE, pp: 106-112. DOI: 10.1109/ICECUBE.2016.7495206

Cortes, C. and V. Vapnik, 1995. Support-vector networks. ML, 20: 273-297.

Cox, D.R., 1958. The regression analysis of binary sequences. J. Royal Statistical Society. Series B (Methodological).

Cukier, K. and V. Mayer-Schoenberger, 2013. The rise of big data: How it's changing the way we think about the world. Foreign Affairs, 92: 1-28.

Drummond, C. and R.C. Holte, 2003. C4. 5, class imbalance and cost sensitivity: Why under-sampling beats over-sampling. Proceedings of the Workshop on Learning from Imbalanced Datasets II. Washington, DC: CiteSeer, pp: 1-8.

Freund, Y. and R.E. Schapire, 1996. Experiments with a new boosting algorithm. Proceedings of the International Conference on Machine Learning. Bari, Italy: Morgan Kaufmann Publishers Inc., pp: 148-156.

Geurts, P., D. Ernst and L. Wehenkel, 2006. Extremely randomized trees. ML, 63: 3-42.

González, H.E., J.R. Martínez, S. Vargas-Salas, A. Solar and L. Veliz *et al*., 2017. A 10-Gene classifier for indeterminate thyroid nodules: Development and multicenter accuracy study. Thyroid, 27: 1058-1067. DOI: 10.1089/thy.2017.0067

Ho, T.K., 1995. Random decision forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, New York, IEEE, pp: 278-282.

Ioniţă, I. and L. Ioniţă, 2016. Prediction of thyroid disease using data mining techniques. BRAIN. Broad Research in Artificial Intelligence Neurosci., 7: 115-124.

Jothi, A.A.J. and M.A. Rajam, 2017. Automatic classification of thyroid histopathology images using multi-classifier system. Multimedia Tools Applications, 76: 18711-18730.

Kongburan, W., P. Padungweang, W. Krathu and J.H. Chan, 2016. Semi-automatic construction of thyroid cancer intervention corpus from biomedical abstracts. Proceedings of the 8th International Conference on Advanced Computational Intelligence (ICACI), New York, IEEE, pp: 150-157. DOI: 10.1109/ICACI.2016.7449819

Marsland, S., 2015. Machine Learning: An algorithmic perspective. Boca Raton, FL: CRC Press.

Pazzani, M., C. Merz, P. Murphy, K. Ali and T. Hume *et al*., 1994. Reducing misclassification costs. Proceedings of the Machine Learning. New Brunswick, NJ, USA: Morgan Kaufmann Publishers Inc., pp: 217-225. DOI: 10.1016/B978-1-55860-335-6.50034-9

Prasad, V., T.S. Rao and M.S.P. Babu, 2016. Thyroid disease diagnosis via hybrid architecture composing rough data sets theory and ML algorithms. Soft Computing, 20: 1179-1189. DOI: 10.1007/s00500-014-1581-5

Prerana, P.S. and K. Taneja, 2015. Predictive data mining for diagnosis of thyroid disease using neural network. Int. J. Res. Management, Science Technol., 3: 75-80.

Saastamoinen, K. and J. Ketola, 2006. Medical data classification using logical similarity based measures. Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems. New York, NY: IEEE, pp: 1-5. DOI: 10.1109/ICCIS.2006.252362

Upadhayay, A., S. Shukla and S. Kumar, 2013. Empirical comparison by data mining classification algorithms (C 4.5 & C 5.0) for thyroid cancer data set. Int. J. Computer Science Communication Networks, 3: 64.