Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 1031–1040

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Thyroid Disease Treatment prediction with machine learning approaches

Lerina Aversano[a,*], Mario Luca Bernardi[a], Marta Cimitile[b], Martina Iammarino[a], Paolo Emidio Macchia[c], Immacolata Cristina Nettore[c], Chiara Verdone[a]

[a] *University of Sannio, Dept. of Engineering, Via Traiano 1, Benevento 82100, Italy*
[b] *UnitelmaSapienza University, Viale Regina Elena 295,Rome 00161, Italy*
[c] *University of Naples Federico II, Dept. of Clinical Medicine and Surgery, Naples, Italy*

Abstract

The thyroid is an endocrine gland located in the anterior region of the neck: its main task is to produce thyroid hormones, which are functional to our entire body. Its possible dysfunction can lead to the production of an insufficient or excessive amount of thyroid hormone. Therefore, the thyroid can become inflamed or swollen due to one or more swellings forming inside it. Some of these nodules can be the site of malignant tumors. One of the most used treatments is sodium levothyroxine, also known as LT4, a synthetic thyroid hormone used in the treatment of thyroid disorders and diseases. Predictions about the treatment can be important for supporting endocrinologists' activities and improve the quality of the patients' life. To date, there are numerous studies in the literature that focus on the prediction of thyroid diseases on the trend of the hormonal parameters of people. This work, differently, aims to predict the LT4 treatment trend for patients suffering from hypothyroidism. To this end, a dedicated dataset was built that includes medical information related to patients being treated in the "AOU Federico II" hospital of Naples. For each patient, the clinical history is available over time, and therefore on the basis of the trend of the hormonal parameters and other attributes considered it was possible to predict the course of each patient's treatment in order to understand if this should be increased or decreased. To conduct this study, we used different machine learning algorithms. In particular, we compared the results of 10 different classifiers. The performances of the different algorithms show good results, especially in the case of the Extra-Tree Classifier, where the accuracy reaches 84%.

Keywords: Thyroid diseases ; Thyroid treatment ; Prediction ; Classifiers; K-Cross Validation ; machine learning

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: aversano@unisannio.it

1877-0509
10.1016/j.procs.2021.08.106

1. Introduction

The thyroid is an endocrine gland found in the neck whose function is to produce hormones (FT3 and FT4) which it releases into the bloodstream.

Thyroid hormones regulate, among other things, heart rate, body temperature, and above all metabolism, that is the way the body uses and consumes nutrients [26]. The thyroid gland may function more than normal (hyperthyroidism with increased hormones) or less than normal (hypothyroidism with low hormones) and in both cases, major disorders [24] can occur. Furthermore, the thyroid gland can undergo inflammation (thyroiditis) or enlarge due to one or more swellings that are formed inside it (nodules, multinodular goiter). Some of these nodules can be the site of malignant tumors. For this reason, the treatment of thyroid diseases is a very critical issue.

The most common cure for thyroid diseases consists to use levothyroxine (LT4), indicated for the treatment of hypothyroidism states [12]. In detail, this active ingredient can be used in the following cases:

- goiter;
- prophylaxis of relapses after total or partial removal of the goiter;
- hormone replacement therapy in case of thyroid hypofunction;
- inflammation of the thyroid gland;
- therapy with antithyroid drugs;
- malignant thyroid tumors, especially after surgery to suppress new tumor growth and to compensate for the lack of thyroid hormone.

Dosing the levothyroxine is not an easy task since the treatments can vary greatly and strongly depend on the amount of residual thyroid function of the patient, the body weight, and thyroid-stimulating hormone levels [12].

For this reason, the dose of levothyroxine should be administered over the patients lifetime and adjusted based on the physiological changes (a.e., weight or hormonal changes) throughout life and concomitant medical conditions (a.e., pregnant women). This requires continuous monitoring of the patients status based on clinical and laboratory assessment and appropriate adjustment of their levothyroxine therapy. Therefore, the prediction of treatment trends could represent an useful support to the endocrinologist and can improve the quality of life of the patient.

The use of machine learning techniques can effectively support endocrinologists while monitoring patients. Recent studies have been successfully applied to classify and predict and, for this reason, have been widely used in the diagnosis of many different problems such as heart disease [17], diabetes [14] and Parkinson's disease [3, 4], reducing the time and costs required for the treatment of a patient.

This study proposes an approach based on machine learning techniques exploiting hormonal parameters related to the thyroid and other clinical data concerning the patient, to predict if the patient's treatment needs to be increased, decreased, or remain unchanged.

Differently from other studies, that mainly aim to detect the disease, the proposed approach focuses on the clinical history of patients suffering from hypothyroidism and treated with thyroid hormone to predict the treatment trend. The proposed prediction approach consisted of a set of features identified on the basis of the endocrinologist experience. These features are used to train different machine learning and a neural network classifier to explore their capabilities to predict patient's LT4 treatment trends. The performance of the classifiers is evaluated on a real dataset obtained by integrating data extracted by the medical system used at the "AOU Federico II" hospital of Naples. Specifically, another contribute of the paper is this dataset, that is obtained preprocessing the information of each patient about the hystory of the clinical picture. The time varies from patient to patient, in fact, in some cases, the information is acquired thanks to visits carried out over a year, in other cases, there is information relating to decades. The paper is organized as follows. Section 2 provides a brief discussion of the related literature while Section 3 describes the study definition and gives an accurate description of the dataset built to perform the validation. The results are presented and discussed in Section 4, while threats to the validity of the study are discussed in Section 5. Finally, Section 6 concludes the article and defines some directions for the future.

2. Related Works

In literature, several studies deal with the identification of thyroid diseases thanks to the use of hormonal parameters and personal data of the patient, such as age and sex. Notably, some studies use machine learning classification and prediction models, while other approaches use deep neural network models.

In the first group, we find the work of Izdihar and Bozkus [2] who used a dataset from the UCI repository to classify thyroid disease using the decision tree algorithm. In particular, they have developed a machine learning tool for the diagnosis of thyroid diseases, called MLTDD capable of making an intelligent forecast of thyroid gland diseases. This study shows an overall accuracy of 98.7% and 99.8% for testing.

Authors in [21, 25], also focus on machine learning techniques such as Support Vector Machine (SVM), Multiple Linear Regression, Nave Bayes, Decision Trees, to perform a comparative diagnosis of thyroid disease. Their results (precision is equal to 99.23%) show that decision trees have the best performance and can be used successfully as an aid in the detection of thyroid disease.

In [16] and [5], the authors conducted a study, the goal of which was to predict thyroid disease using different data mining techniques and find the correlation between TSH, T3, T4 characteristics with hyperthyroidism or hypothyroidism. In particular, the authors considered KNN, Nave Bayes, Support vector machine, ID3 as data mining algorithms, applying them to the public dataset of the UCI data archive.

On the other hand, in the study [20], the authors used neural networks (MLP, PNN, GRNN, FTDNN, CFNN) to diagnose types of thyroid disease. More specifically they conducted a study on 244 subjects suffering from different pathologies to investigate the state of their thyroid, taking into account some hormonal parameters and the patient's age. The results of this research show that the neural network offers very precise answers, classifying correct thyroid pathologies based on hormonal parameters.

Authors in [8] conducted a study aimed at diagnosing hyperthyroidism and hypothyroidism, the two most common thyroid disorders. The classification was carried out using two techniques, multinomial logistic regression models and neural networks. The research was conducted on 310 patients, and even in this case, the models took as input demographics and hormonal parameters. The results showed better performance of the neural network model (with an average accuracy of 96.3%) than multinomial logistic regression (with a mean accuracy of 91.4%) in all cases.

This paper differs from the above discussed approaches because it is the first (to the best of our knowledge) to focus on the thyroid disease treatment prediction. Therefore, the main objective of this work is to use all the data collected overtime on a patient to predict whether the LT4-based treatment needs to be increased or decreased. According to this goal, in this study, we use and compare different machine learning techniques to predict the course of care of patients suffering from thyroid disease. Moreover, this study proposes a new set of features identified on the basis of the endocrinologist's experience in the patient's treatment.

Finally, another contribution of our work is represented by the use of a dataset obtained by extracting real data. This dataset is built by integrating two sub-datasets including information about the patient's medical history and the patient current state.

3. Approach

The proposed approach aims to predict the treatment trend of the tyroid disease on the base of patient's historical and current data. In this section, we first describe the construction of the dataset, then we describe the proposed features model and finally we focus on the machine learning algorithms used to conduct the study and their validation.

3.1. Data Collection

To conduct this study we built a dataset from patients with thyroid disease being treated at the "AOU Federico II" Naples hospital. This dataset is obtained as the integration of two data sources containing information related to 800 patients.

In particular, the first data source collects for each patient the personal information (such as age, date of birth, sex, pathology, profession, education, sex, marital status), the family history, the physical characteristics (such as height, weight, body mass index, information relating to menstruation or possible pregnancies for women and others concerning appetite, alvus, diuresis) and some clinical information (such as skin, neck, heart, thorax, abdomen, extremities and eyes).

The second data source, on the other hand, is the diary of the doctor's visits. It contains, for each patient, all the information about the clinical tests and visits made by the doctor in the time.

The data from the two data souces are then merged into one large data set, using the patient identifier as a key. Successively, a cleaning activity is performed. In particular, all the missing values and uncorrected data are managed. Moreover, looking at the clinical history of each patient, all the patients having a single visit are discarded from the dataset as they are not adequate to study the evolution of the disease. Furthermore, among the patients, in this study we select only those suffering from hypothyroidism because they underwent the analysed treatment for the prediction. The obtained dataset contains three macro-groups of pathologies: Congenital hypothyroidism (i), Hashimoto's thyroiditis with hypothyroidism (ii) and hypothyroidism (iii). The final dataset collects data from 247 patients (196 women and 51 men), with a mean age of 46 years, each of whom has hypothyroidism and has more than one hospital visit over several years, or in some cases multiple visits in the same year. More specifically, the entire collection of data sets 2784 instances, referring to a specific patient and a specific visit, during which the necessary physical and clinical values are stored.

Starting from this dataset (D), in this study three additional datasets are obtained by applying different pre-processing techniques. The first dataset (D1), is obtained by performing an interpolation and a balancing on the dataset D and consists of 6556 instances. The interpolation consists of eliminating empty cells. In particular, we use spline interpolation [1]. The balancing is obtained using the Synthetic Minority Oversampling TEchnique (SMOTE) oversampling method [10]. SMOTE tries to mitigate the imbalance problem by oversampling the examples in the minority class, by randomly increasing minority class using replication. The second dataset (D2) is obtained from the dataset D afther a discretization and a balancing step. It contains 6556 instances. The discretization consists of dividing the values assumed by the continuous attributes into multiple intervals of equal number and then replacing them with a certain label relating to the interval in which they fall. Finally, the last dataset (D3) is obtained by performing a normalization and balancing of the dataset D. It contains 6556 instances. The goal of normalization is to change the values of the numeric attributes in the dataset to use a common scale, without affecting the differences between the ranges of values or the loss of information. In this work, we use mean normalization.

3.2. The proposed feature model

The set of features used in this study is extracted by a initial group of 135 available attributes describing the patients from different point of views. These attributes refer to the patient's personal data, personal and family medical history, physical conditions, hormonal and thyroid parameters, parameters relating to blood tests. Of the initial feature set, we select 27 attributes that are related to patient information and thyroid parameters. This selection is perfomed by an expert on the base of the criteria usually used to evaluate the patient's treatment. Several features are also discarded because they are present only in some patients and therefore missing for more than 50% of the entire dataset. The Table 1 includes for each row an attribute considered in the dataset, reporting the name in the first column, a brief description in the second column, and the type in the last column.

Last row shows the predicted feature: LT4 treatment trend. In particular, this can take four values, *increased* when the dosage needs to be increased, *decreased* on the contrary when the patient needs a lower dosage, *stable* when the treatment must remain unchanged and others instead is a mixture of different situations, such as the one in which the patient must suspend the treatment.

3.3. Classifiers

To predict the course of treatment to which a patient with a thyroid problem is subjected, we used more than one machine learning classifier. The classifier suggests to the endocrinologist if, based on the historical and current patient's state, the LT4 dosage should be increased, decreased, or maintained. We compared several algorithms with different characteristics to understand which was the one that best labeled each instance of our dataset.

A subset of the chosen algorithms belongs to the boosting algorithms. In particular, we used AdaBoost, Gradient Boosting , XGBC, and CatBoost. These approaches use decision trees as weak learners, are not parametric, and do not assume or require data to follow a particular distribution.

Ada-boost Classifier [13] (ABC) assigns weight to each training item, after training a grader at any level, an incorrectly graded item assigns more weight so that it appears in the next grader's training subset more likely. Its main drawback is the noisy data because the efficiency of AdaBoost is strongly influenced by the outliers as the algorithm tries to adapt perfectly to each point.

Table 1. The proposed features model

Feature	Description (*t*)	Type (*t*)
Height	patient's height (cm)	int
Body mass index	patient's BMI	float
Age at the visit	patient's age on the date of medical check-up	int
Pathology	patient's thyroid disease (in some cases more than one)	string
Severity of the pathology	severity of the thyroid disease	string
Cause of the pathology	cause of the thyroid disease	string
Weight	patient's weight (kg)	int
Gender	patient's gender (Male - Female)	string
Familiar anamnesis: Thyroid Diseases	it indicates if there were or are Thyroid diseases in the patient's family (yes - no)	boolean
Familiar anamnesis: Diabetes	it indicates if there were or are diabetics in the patient's family (yes - no)	boolean
Menarche	age at which a patient had the menarch (female patients only)	int
Menstruation	woman's period (female patients only) (regular - irregular)	string
pregnancies	number of pregnancies, if any (female patients only)	int
Pregnancy interruptions	number of pregnancy interrunptions, if any (female patients only)	int
Menopausal age	age at which a patient entered menopause (female patients only)	int
Appetite	degree of the patient's appetite (poor - good - regular - excessive - great - variable)	string
Bowel function	degree of the patient's bowel function (regular - irregular - constipation - frequent - variable)	string
Diuresis	degree of the patient's diuresis (regular - irregular - frequent - poor - variable)	string
TSH	TSH (Thyroid Stimulating Hormone) is a pituitary hormone that stimulates the thyroid gland to produce thyroxine (T4), and then triiodothyronine (T3) which stimulates the metabolism of almost every tissue in the body	float
FT3	FT3 Triiodothyronine is a thyroid hormone, partially composed of iodine.	float
FT4	FT4 Thyroxine is a thyroid hormone, partially composed of iodine.	float
Thyroglobulin	Thyroid hormone used as a tumor marker to evaluate the efficacy of thyroid cancer therapy.	float
Ab<>	Thyroid antibodies are components of the immune system that are mistakenly directed against the thyroid gland or against certain factors that are fundamental for its normal function	float
AbTg	anti body Thyroglobulin	float
AbTPO	Anti-thyreoperoxidase antibodies	float
LT4	Levothyroxine, also known as L-thyroxine, is a manufactured form of the thyroid hormone thyroxine (T4). It is used to treat thyroid hormone deficiency, including the severe form known as myxedema coma. It may also be used to treat and prevent certain types of thyroid tumors. (doses per week)	int
LT4 treatment trend	trend of LT4 treatment (increased - decreased - stable - others)	string

Gradient Boosting Classifier (GBC) [7] means Gradient Descent + Boosting. It is similar to the previous one, but the difference lies in what it does with the under-fitted values of its predecessor. Unlike AdaBoost, which changes the instance weights with each interaction, this method tries to adapt the new predictor to the residual errors made by the previous predictor. Gradient augmentation redefines augmentation as a numerical optimization problem where the goal is to minimize the model's loss function by adding weak students using a procedure similar to gradient descent.

XGB Classifier (XGBC) Extreme Gradient Boosting [19] is an advanced implementation of Gradient Boosting. More specifically, XGBoost, short for eXtreme Gradient Boosting, is a decision tree-based supervised machine learning algorithm that uses a gradient enhancement framework. Characterized by optimal performance and speed, it is based on three essential characteristics: Sparse Aware implementation with automatic handling of missing data values. Block structure to support parallelization of tree construction. Continuous training to further improve an already adapted model on new data. Xgboost no longer runs trees in parallel as you noticed, you need predictions after each tree to update the gradients. Parallelization occurs during the construction of each tree, at a very low level. Each independent branch of the tree is trained separately.

CatBoost Classifier (CBC) [11] during the training build a series of decision trees consecutively. Each subsequent tree is built with a reduced loss compared to previous trees. It is designed for categorical data, and precisely the internal identification of some categorical data significantly slows down its training time compared to XGBoost, but by taking longer it has better learning resources.

The second group of classifiers chosen is based on decision trees [23]. In particular, we used the Decision Tree, the ExtraTree, and the Random Forest.

With *Decision Tree Classifier* (DCC) [23] the classification functions are learned in the form of a tree where each internal node represents a variable, an arc to a child node represents a possible value for that property, and a leaf the expected value for the class starting from the values of the other properties, which in the tree is represented by the path from the root node to the leaf node. *Extra-Tree Classifier* (EXTC) [15] and *Random Forest Classifier* (RFC) [9] are based on decision trees. These classifiers differ for three main reasons: the input data, the division of knots, and the speed. The random forest subsamples the input data with the replacement, while the additional trees use the entire original sample. Another difference is the selection of the cutting points to divide the nodes. Random Forest chooses the optimal division while Extra Trees chooses it randomly. However, once the split points are selected, the

two algorithms choose the best of all subsets of characteristics. Finally, in the execution time, Extra Trees is much faster.

To conclude our study we decided to also compare the results of two of the most used approaches, namely the classifier based on the Bayesian algorithm and the K-Nearest Neighbors, and one of the neural networks, MLP.

Naive Bayes Classifier (NBC) [27] is a supervised learning algorithm based on Bayes' theorem, which describes the probability of an event, based on prior knowledge of the conditions that could be related to the event. It is based on the fact that all characteristics are unrelated l 'one to the other and essentially its operation consists of three steps: Calculation of the probability of class (i), Calculation of conditional probability (ii), and decision (iii).

The *K-Nearest Neighbors Classifier* (KNNC) [18] is non-parametric, meaning it makes no assumptions about the distribution of the data it analyzes. Its operation is based on the similarity of characteristics: the closer an instance is to a data point, the more the knn will consider them similar. Usually, the similarity is calculated using the Euclidean distance and the algorithm envisages setting a parameter k, chosen arbitrarily, which identifies the number of closest data points. The algorithm evaluates the k minimum distances thus obtained. The class that obtains the greatest number of these distances is chosen as the prediction. This classifier, in particular, was chosen because it is excellent in cases where there is no previous knowledge on the distribution of data, as in our case.

Finally, we used the *Multilayer Perceptron Classifier* (MLPC), a deep artificial neural network [6]. It is composed of more than one perceptron, with an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and, between these two, an arbitrary number of hidden layers that are the real computational engine of the MLP. This approach was selected to understand the best classifier, also having results with a first deep learning approach. All models were generated in Python, thanks to the use of the scikit-learn library, which is currently one of the most popular open-source libraries of machine learning algorithms. For each classifier the parameters used to train the model are reported in the following:.

- DecisionTreeClassifier(class_weight='balanced', max_depth=5) : the class_weight is the weights associated with classes in the form "class_label: weight". The balanced mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as "n_samples / (n_classes * np.bincount(y))", instead the other parameter indicates the maximum depth of the tree, in our case 5.

- KNeighborsClassifier(3) : where 3 is the number of neighbors to use.

- RandomForestClassifier(class_weight='balanced', random_state=1, max_depth=5, n_estimators=10, max_features=1): as in the decision tree, here too we have chosen the weight associated with the balanced classes, and as the maximum depth 5. The random state parameter causes the same result to always be obtained when the train_test_split function divides arrays or matrices into random trains and test subsets. n_estimators indicates the number of trees in the forest and max_features the number of features to consider when looking for the best subdivision. In this case, choosing an integer, we have chosen to consider the 'max_features' characteristics in each subdivision.

- ExtraTreesClassifier(class_weight='balanced', random_state=1): in this case we have chosen to balance the weights of the classes and to adopt 1 as random state which controls 3 sources of randomness: - the bootstrapping of the samples used when building trees (if "bootstrap=True") - the sampling of the features to consider when looking for the best split at each node (if "max_features n_features") - the draw of the splits for each of the 'max_features'

- MLPClassifier(hidden_layer_sizes=(256, 128, 64, 32), activation="relu", random_state=1): hidden_layer_sizes represents the number of neurons in the i-th hidden layer, activation is the activation function for the hidden layer. and we chose the "relu", the rectified linear unit function which returns f (x) = max (0, x), random_state which determines the generation of random numbers for the weights and the initialization of the bias, the division train- test if early shutdown and batch sampling is used when solver = "sgd" or "adam". Pass an int for reproducible results across multiple function calls.

- XGBClassifier(learning_rate=0.01, use_label_encoder=False) : learning_rate to increase learning speed, and use_label_encoder set to false to not use scikit-learn's tag encoder to encode tags.

- CatBoostClassifier(max_depth=4, verbose=False): in this case we have set the maximum depth to be 4, and the verbose to False, so logging_level is set to Silent.

- GradientBoostingClassifier(n_estimators=20, learning_rate=0.01, max_features=2, max_depth=2, random_state=0): n_estimators indicates the number of boost phases to perform. The gradient boost is robust enough for over-fitting, so a large number usually results in better performance. The learning rate corresponds to the speed with which the error is corrected from each tree to the next and is a simple multiplier, max_features corresponds to the number of features to consider when looking for the best subdivision: max_depth indicates the maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree. random_state: Checks the random seed given to each Tree estimator at each boost iteration.

3.4. Validation

To define the degree of accuracy or effectiveness of any machine learning model, it is necessary to perform a thorough assessment of the performance obtained when the approach is used to produce forecasts on real data. Therefore, to obtain a more reliable estimate of performance metrics we use cross-validation [22], a statistical method that is particularly suitable for machine learning models assessment.

In particular, the k-fold cross-validation consists in the subdivision of the total data set into k parts of equal size and, at each step, the k-th part of the data set becomes the validation part, while the remaining part always become the training set. Hence, the model is trained for each of the k parts, thus avoiding problems of overfitting, but also of asymmetrical sampling (and therefore affected by distortion) of the observed sample, typical of the subdivision of the data into only two parts (i.e. training / validation).

Specifically, we used k-fold cross-validation which involves randomly dividing the data set into k groups, of approximately equal size. The first fold is kept for testing, while the model is trained on the k-1 folds. The process is repeated K times and each time different bends or a different set of data points are used for validation. This means that each data point must be in a test set exactly once and must be in a training set k-1 times. In our case, we have chosen to use a k equal to 10. The disadvantage of this method is related to the effort. The training algorithm must be executed k times, one for each part.

The metrics we used to validate the model are accuracy, precision, recall, and F-score.

In detail, accuracy indicates the accuracy of the model, i.e. the fraction of the test dataset on which the model provides a correct prediction. By defining true positives (TP) and true negatives (TN) as the instances that are correctly classified, and the false positives (FP) and false negatives (FN) as the instances that are misclassified, accuracy is defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Precision and recall, on the other hand, want to quantify the rate of True Positive (TP) and True Negative (TN) respectively. More specifically, precision is the ability of a classifier not to label a positive instance that is actually negative and is defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Instead, recall measures the sensitivity of the model. It is the ratio between the correct predictions for a class on the total of cases in which it actually occurs and is defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

Table 2. Results on the dataset D.

Classifier	Acc	Pre	Rec	F1
XGBC	0.59	0.32	0.31	0.29
CBC	0.71	0.37	0.35	0.34

Table 3. Results on the interpolated and balanced dataset (D1).

Classifier	Acc	Pre	Rec	F1
DTC	0.44	0.45	0.44	0.44
NBC	0.34	0.32	0.34	0.30
KNNC	0.50	0.52	0.50	0.48
RFC	0.44	0.44	0.44	0.41
EXTC	**0.59**	**0.60**	**0.59**	**0.58**
MLPC	0.44	0.44	0.45	0.41
XGBC	0.51	0.52	0.51	0.47
CBC	**0.57**	**0.57**	**0.57**	**0.55**
ABC	0.42	0.40	0.42	0.39
GBC	0.36	0.37	0.36	0.33

Table 4. Results on the discretized and balanced dataset (D2).

Classifier	Acc	Pre	Rec	F1
DCT	0.60	0.62	0.60	0.58
NBC	0.49	0.50	0.49	0.47
KNNC	**0.81**	**0.81**	**0.81**	**0.80**
RFC	0.57	0.56	0.56	0.54
EXTC	**0.84**	**0.85**	**0.84**	**0.84**
MLPC	0.73	0.74	0.72	0.72
XGBC	0.71	0.72	0.71	0.72
CBC	**0.80**	**0.81**	**0.81**	**0.80**
ABC	0.57	0.57	0.57	0.53
GBC	0.58	0.58	0.58	0.56

Table 5. Results on the normalized and balanced dataset (D3).

Classifier	Acc	Pre	Rec	F1
DTC	0.61	0.63	0.61	0.59
NBC	0.53	0.56	0.53	0.52
KNNC	**0.81**	**0.81**	**0.81**	**0.80**
RFC	0.58	0.58	0.58	0.57
EXTC	**0.82**	**0.84**	**0.82**	**0.82**
MLPC	0.69	0.70	0.70	0.68
XGBC	0.72	0.73	0.72	0.71
CBC	**0.82**	**0.83**	**0.82**	**0.82**
ABC	0.58	0.60	0.58	0.57
GBC	0.58	0.60	0.60	0.59

Finally, the F1-score used to compare classifier models and not global precision. Represents the weighted harmonic average of the Precision and Recall metrics, which gives more weight to small values. This causes a classifier to get a high F1-score only when accuracy and recovery are both high. It is defined as follows:

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

4. Results and Discussion

In this section, we report and discuss the results obtained with our approach.

To predict the course of care of a patient undergoing treatment with levothyroxine and to understand if its dosage should remain stable, or be increased or decreased, we compared the results of the aforementioned classifiers desciobed in the previous subsection. As already remarked, for the validation of the proposed approach, we use the k-cross validation, setting k = 10. In particular, the performance of the classifiers are evaluated on all the datasets described in Section 3. Therefore we report the results of all the four conducted experiments (one for each considered dataset), respectively in the Table 2, Table 3, Table 6 and Table 5. Each table shows in the first column the considered classifier and then the value relating to the accuracy, precision, recall, and F-Score.

On the original dataset, only two of the chosen classifiers could be applied: XGBC and CBC. These are the only classifiers, among those selected, capable of working giving the reduced number of available instances. The table 2 reports the results obtained. As you can see, the two classifiers have higher values for precision than the others, respectively 59% and 71%, but neither metric chosen for validation takes on unsatisfactory values. For example, the F-score obtained is 29% and 34% respectively. For this reason, we have carried out the aforementioned pre-processing.

Table 3 shows the results obtained with the dataset D1 subjected to interpolation and balancing, reporting in bold the best results found. As you can see, the models that have the best performances are EXTC and CBC. But their F1-score is between 55 and 60%, so they are still not satisfactory.

Table 6. Results comparison for the adopted classifiers (bold text highlights the best results).

Using discretization and balancing on the dataset, the results on D2 confirm the classifiers just mentioned as the best 6. In particular, KNNC improves concerning all metrics, reaching the F-score of 80 %, CBC improves in the same way, reaching the same f-score score. Finally, EXTC reaches the highest score, with an F-score of 84 %.

Table 5 shows the results obtained with the normalization and balancing of the dataset. Also in this case the three aforementioned are confirmed as the best classifiers. With these preprocessing techniques, the KNN model achieves 80% F-score, EXTC, and CBC 0.82%.

Finally, in summary, the best results are obtained by using discretization and balancing as data pre-processing and ExtraTreesClassifier as a machine learning model. With these parameters, we have obtained accuracy, precision, recall, and F-Score, respectively: 84%, 85%, 84%, and 84%.

5. Threats to validity

There are three types of threats to validity in the proposed study: constructive, internal, and external.

With reference to the construct validity, there may be some inaccuracies and omissions due to the construction phase of the dataset used. In fact, for this study, real data is used, belonging to patients being treated at a hospital. This risk has been mitigated by adopting a manual construction process articulated on multiple steps. Therefore, in order to avoid errors, the authors discussed among themselves to identify guidelines, so that each attribute of the dataset could be categorized. Following this, a careful manual cleaning operation was carried out.

Furthermore, internal validity could be undermined by classification errors if the adopted datasets were not labeled correctly. To avoid this limitation, several manual checks have been carried out.

Finally, the threats to external validity concern the generalization of the results discussed. With reference to the constructive ones, there may be some inaccuracies and omissions due to the construction phase of the dataset used. For this study, in fact, real data is used, belonging to patients being treated at a hospital. This risk has been significantly mitigated by adopting a manual construction process based on multiple steps. Therefore, in order to avoid errors, the authors argued with each other to identify the guidelines, so that each attribute of the dataset could be classified. A thorough manual cleaning was then performed.

6. Conclusion and future work

The thyroid is defined as the ”powerhouse” of our body: if something in this gland fails, the whole body suffers as well. Therefore, the early diagnosis of a possible malfunction plays a fundamental role, as well as the prediction of the course of treatment of a patient with hypothyroidism, which can be of great help for doctors who have patients under treatment. In this study we proposed an approach to predict the tyroid disease treatment.

This approach is intended to be a machine learning-based decision support system for endocrinologists treating patients with thyroid disease. In medicine, these methods are gathering growing interest and our work can be of great help thanks also to the high diagnostic accuracy we have achieved in the specific clinical context. In particular, the proposed model is able to predict the progress of the patient’s treatment on the basis of other parameters related to the person being treated, therefore the doctor is facilitated in choosing the dosage of the drug to prescribe.

The main unexpected consequences related to the use of ML systems concern the risk of an excessive tendency to rely on these systems by their users, with a consequent disqualification and desensitization towards the clinical context. The proposed feature model is evaluated using 10 different machine learning classifiers, belonging to different classes of algorithms. In fact, in particular, to conduct our experiments we have chosen models belonging to the class of enhancement algorithms, models based on decision trees, models heavily used in literature, and a model based on a neural network to test even a small Deep Learning approach.

The models are tested on an overall dataset containing 2211 instances referring to a total of 247 patients. A contribution of this work consists in the use of a dataset that collected real information belonging to subjects under treatment at the Naples hospital.

Another fundamental step involves the pre-processing of the data to be included in the classifiers, in particular, that relating to the use of the SMOTE method, a minority oversampling technique. This is necessary since the original

dataset is unbalanced, as the three classes are not equally represented. This phase mitigates this problem by oversampling the examples in the minority class, randomly.

The results obtained and shown in Section 4 demonstrate a good performance of the EXTC model, compared to the other approaches used, with an F-score equal to 84%.

On the other hand, the main limitation of this study concerns the quality of the dataset, as, as already mentioned, this was constructed starting from real data belonging to patients being treated at a hospital.

In the future, to better generalize our findings it is necessary to further expand the set of data and attributes considered. With more data the training process is likely to produce more effective classifiers also allowing a more reliable estimate of the exhibited performance. Finally, another aspect that could be investigated concerns the presence of any secondary thyroid disease linked to the patient, to understand if there is a particular additional thyroid disease that can affect hypothyroidism. In fact, it often happens that patients are suffering from more than one thyroid disease at the same time.

References

[1] *Spline Interpolation*, pages 141–173. Springer New York, New York, NY, 2006.

[2] Izdihar Al-muwaffaq and Zeki Bozkus. Mltdd : Use of machine learning techniques for diagnosis of thyroid gland disorder. 2016.

[3] L. Aversano, M. L. Bernardi, M. Cimitile, and R. Pecori. Early detection of parkinson disease using deep neural networks on gait dynamics. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[4] Lerina Aversano, Mario Luca Bernardi, Marta Cimitile, and Riccardo Pecori. Fuzzy neural networks to detect parkinson disease. In *29th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2020, Glasgow, UK, July 19-24, 2020*, pages 1–8. IEEE, 2020.

[5] A. Begum and A. Parkavi. Prediction of thyroid disease using data mining techniques. In *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*, pages 342–345, 2019.

[6] Mario Luca Bernardi, Marta Cimitile, Fabio Martinelli, and Francesco Mercaldo. Keystroke analysis for user identification using deep neural networks. In *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, pages 1–8. IEEE, 2019.

[7] G. Biau, B. Cadre, and L. Rouvière. Accelerated gradient boosting. *Machine Learning*, 108(6):971–992, 2019.

[8] Shiva Borzouei, Hossein Mahjub, NegarAsaad Sajadi, and Maryam Farhadian. Diagnosing thyroid disorders: Comparison of logistic regression and neural network models. *Journal of Family Medicine and Primary Care*, 9:1470, 06 2020.

[9] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[10] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321357, June 2002.

[11] Essam Al Daoud. Comparison between xgboost, lightgbm and catboost using a home credit dataset. *International Journal of Computer and Information Engineering*, 13(1):6 – 10, 2019.

[12] Leonidas H. Duntas and Jacqueline Jonklaas. Levothyroxine dose adjustment to optimise therapy throughout a patient's lifetime. *Advances in Therapy*, 36(2):30–46, 2019.

[13] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[14] Siva Shankar G. and Manikandan K. Diagnosis of diabetes diseases using optimized fuzzy rule set by grey wolf optimization. *Pattern Recognition Letters*, 125:432 – 438, 2019.

[15] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[16] Irina Ionit and Liviu Ionit. Prediction of thyroid disease using data mining techniques. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 7(3), 2016.

[17] T. Karaylan and . Kl. Prediction of heart disease using neural network. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 719–723, 2017.

[18] Huafeng Liu, Chao Li, Shuheng Zhang, Huan Zhang, Lifang Pang, Kinman Lam, Chun Hui, and Su Zhang. Using the k-nearest neighbor algorithm for the classification of lymph node metastasis in gastric cancer. *Computational and Mathematical Methods in Medicine*, 2012:876545, 2012.

[19] Saad Ijaz Majid, Syed Waqar Shah, and Safdar Nawaz Khan Marwat. Applications of extreme gradient boosting for intelligent handovers from 4g to 5g (mm waves) technology with partial radio contact. *Electronics*, 9(4), 2020.

[20] Mahdiyar O. Obeidavi M. R, Rafiee A. Diagnosing thyroid disease by neural networks. 2017.

[21] S. Razia, P. SwathiPrathyusha, N. Krishna, and N. Sumana. A comparative study of machine learning algorithms on thyroid disease prediction. *International journal of engineering and technology*, 7:315, 2018.

[22] Payam Refaeilzadeh, Lei Tang, and Huan Liu. *Cross-Validation*, pages 532–538. Springer US, Boston, MA, 2009.

[23] Lior Rokach and Oded Maimon. *Data Mining with Decision Trees*. WORLD SCIENTIFIC, 2nd edition, 2014.

[24] Jo£o Hamilton Romaldini, Jos Augusto Sgarbi, and Chady Satt Farah. [subclinical thyroid disease: subclinical hypothyroidism and hyperthyroidism]. *Arquivos brasileiros de endocrinologia e metabologia*, 48(1):147158, February 2004.

[25] Ankita Tyagi and Ritika Mehra. Interactive thyroid disease prediction system using machine learning technique. 03 2019.

[26] Saffron Whitehead. *Endocrinology: An integrated approach*. 01 2001.

[27] Harry Zhang and Jiang Su. Naive bayesian classifiers for ranking. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Machine Learning: ECML 2004*, pages 501–512, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.