Hypothyroid Classification using Machine Learning Approaches and Comparative Analysis

Vishakha Mistry

Gujarat, India

Abstract - *Diseases increase day by day as a result of lifestyle. In particular, Thyroid Disease (TD) is one of the most advanced endocrine disorders in humans today. Thyroid hormone regulates many metabolic processes throughout the body. Machine learning (ML) has shown productive results in decision-making and prediction from large data generated by the healthcare industry. It has been found that classification is widely used in all fields. Classification is a supervised learning method that uses a predefined data set to make precise decisions. In this work, I used Naïve bayes, KNN and the Decision tree to identify a type of thyroid disease using ANACONDA as software and a python programming language to use these algorithms. I collected a thyroid patient database from the UCI Machine Learning Repository. I have compared the results of the different classification techniques mentioned above, and their accuracy has been compared to the confusion matrix. Gradient boosted trees have become the go-to algorithms when it comes to training with table data. Over the past few years, we have been fortunate to have not only a single implementation of advanced trees, but a few advanced algorithms - each with its own unique features. In this work, I used Gradient Boosting, Adaboost, XGBoost, LightGBM and CatBoost. Comparisons are made about the accuracy and speed of training.*

Key Words: **Hypothyroid, Gradient Boosting, Machine Learning, KNN, CatBoost, LightGBM, Naïve bayes.**

1. INTRODUCTION

Thyroid diseases are common worldwide. Disease diagnosis is a difficult step in the medical field because numerous diseases occur per annum. Machine learning is employed in various areas like education and healthcare. With the advancement of technology, the higher computing power and availability of datasets on open-source repositories have further increased the utilization of machine learning. Classification techniques play a very important role in analyzing diseases with reduced cost to the patients.

2. OVERVIEW OF THYROID DISEASE

The thyroid gland absorbs iodine from food we eat and convert it into two major hormones: Triiodothyronine (T3) and Thyroxine (T4). The thyroid gland is controlled by the pituitary gland which is located in the center of the skull, below your brain. When the thyroid hormones level is too low or high, the pituitary gland produces Thyroid Stimulating Hormone (TSH) which will tell the Thyroid gland to produce more or fewer hormones. Thyroid diseases are classified mainly into two types. If your body makes less thyroid hormone, this is called hypothyroidism. If your body makes too much thyroid hormone, this is called hyperthyroidism [15].

3. LITERATURE SURVEY

Authors in [1] have implemented machine learning techniques such as Nave Bayes, Multiple Linear Regression (MLP), Support Vector Machine (SVM), Decision Trees, to perform a comparative diagnosis of thyroid disease. They show that decision trees are the best performer. Authors in [2] have used support vector machine (SVM), Decision Tree for classification, where data set was chopped for training and testing purpose. Both the methods are compared based on accuracy. The highest accuracy was achieved by SVM with 99.63% accuracy. Authors in [3] used classification techniques like K-nearest neighbour and Naive Bayes. The accuracy of KNN is better than Naive Bayes classifier. The parameters used to classify thyroid disorder are TSH, T4U, and goiter. The KNN achieves an accuracy of 93.44% whereas Naïve Bayes achieves 22.56% accuracy.

S.B.Patel and Yadav in [4] worked to predict the diagnosis of heart disease using classification techniques. Three classification function techniques are compared for predicting heart disease with a reduced number of attributes. In another research Hetal Patel [5] came to the conclusion that the multiclass classifier algorithm achieved the highest accuracy of 99.5%. Yadav and Pal [6] generated ensemble methods for thyroid prediction after comparing bagging, boosting, and stacking methods. In his work, he used Random tree, J48, and, Hoeffding on the thyroid dataset and identifies a more accurate model of a decision tree on all possible experiments. In this analysis, the ensemble classification technique improved to evaluate the accuracy and test the thyroid dataset. K. Rajam [7] research work is based on supervised ML techniques Naïve bayes, decision tree, backpropagation, SVM identifies thyroid disease. Outcomes were evaluated based on parameters speed, accuracy, performance, and cost and found effective for the treatment of the patient.

4. DATASET DESCRIPTION

The data set used for the experimental purpose can be downloaded from UCI machine learning Repository. It has 3772 instances and 30 attributes. Out of them, 24 are categorical and 6 of them are real attributes. We leverage scikit-learn python package for our analysis. A train-test split of 80:30 is used here. My first work is to identify attributes under 3 categories nominal, numeric, and class. The nominal Attributes of the dataset are shown in Table 1.

Table -1: Nominal Attributes

Sr. No.	Name of Attribute	Label	Count
1.	Sex	F	2396
		M	1134
2.	on thyroxine	f	3214
		t	464
3.	query on thyroxine	f	3628
		t	50
4.	on antithyroid medication	f	3637
		t	41
5.	sick	f	3531
		t	147
6.	pregnant	f	3625
		t	53
7.	thyroid surgery	f	3625
		t	53
8.	I131 treatment	f	3619
		t	59
9.	query hypothyroid	f	3444
		t	234
10.	query hyperthyroid	f	3446
		t	232
11.	lithium	f	3660
		t	18
12.	goitre	f	3644
		t	34
13.	tumor	f	3582
		t	96
14.	hypopituitary	f	3677
		t	1
15.	psych	f	3494
		t	184
16.	TSH measured	f	3401
		t	277
17.	T3 measured	f	3001
		t	677
18.	TT4 measured	f	3539
		t	139
19.	T4U measured	f	3383
		t	295
20.	FTI measured	f	3385
		t	293
21.	TBG measured	f	3678
		t	0
22.	referral source	other	2107
		SVI	1034
		SVHC	386
		STMW	112
		SVHD	39
23.	Target class(Hypothyroid)	P	3387
		N	291

The numeric Attributes of the dataset are shown in Table 2.

Table -2: Numeric Attributes

Sr. No.	Attribute Name	Statistic	Value
1	age	Min.	1
		Max.	455
		mean	51.9151
		Std.	20.1451
2	TSH	Min.	0.005
		Max.	530
		mean	5.0888
		Std.	24.5285
3	T3	Min.	0.05
		Max.	10.6000
		mean	2.0135
		Std.	0.8277
4	TT4	Min.	2
		Max.	430
		mean	108.337
		Std.	35.6060
5	T4U	Min.	0.25
		Max.	2.3200
		mean	0.9949
		Std.	0.1955
6	FTI	Min.	0.25
		Max.	2.3200
		mean	0.9949
		Std.	0.1955

5. CLASSIFICATION ALGORITHMS

In machine learning, classification refers to a predictive modelling problem where a class label is predicted for a given input data. Two types of Classification are Supervised classification and unsupervised classification. In supervised classification, labelled datasets is used to train algorithms to classify data. In Unsupervised classification, models are not supervised using training dataset but models themselves find the hidden patterns and insights from the given data. It can be compared to learning which takes place within the human brain while learning new things. Some examples of popular data mining classification algorithms include Support Vector Machine, Decision tree, Naïve Bayes, K-Nearest Neighbour, and ANN. I compared several algorithms with different characteristics to understand which is the best algorithm for a given thyroid dataset. This work also focuses on Extreme Gradient Boosting(XGBoost), CatBoost, and LightGBM for better speed and performance.

Decision Trees are the most popular and widely used in data mining [8]. These architectures use a divide-and-conquer

strategy so as to partition the instance space into decision regions. At first, a root node is designated by employing a test. Then, the value of the related test attribute splits the data set and, the process is repeated until the determined stopping criterion is satisfied. At the end of the tree, each node is known as a leaf node. Each leaf node indicates the class. Also, each branch denotes a path defined as a decision rule.

Ada-boost Classifier [9] (ABC) is termed adaptive because it uses multiple iterations to come up with one a single composite strong learner. Strong learner is created in Adaboost by iteratively adding weak learners. In each phase of training, a new weak learner is added to the ensemble. Then a weighting vector is adjusted to concentrate on examples that were not classified in previous rounds. The result classifier has higher accuracy than the weak learners' classifiers.

Extreme Gradient Boosting (XGBoost) classifier is a scalable highly accurate implementation of gradient boosting ensemble technique. This method has been consistently placing among the top contenders in Kaggle competitions [10]. When compared with other gradient boosting algorithms, XGBoost uses more accurate approximations to find the best model to control the overfitting of data, which gives it better performance.

CatBoost [11] uses a complex ensemble learning technique based on the gradient descent framework. During model training, each Decision Tree learns from the previous tree and influences the upcoming tree to boost the performance of model, thus constructs a strong learner. CatBoost can handle categorical features automatically, thus saving considerable computational time and resources.

LightGBM [12] is a gradient boosting framework based on decision trees to increase the efficiency of the model and lower memory usage. It splits the tree leaf wise with the best fit while other boosting algorithms split the tree depth wise instead of leaf wise. So, when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss as compared to depth-wise algorithm and hence achieves much better accuracy which might rarely be achieved by any of the present algorithms.

To conclude my study, I decided to compare the results of the two most popular approaches, the Bayesian algorithm and the K-Nearest Neighbours.

Naïve Bayes is a supervised learning algorithm. It works on the Bayes theorem and is used for solving classification problems. The theorem has an assumption that each feature makes an independent and equal contributor. The theorem predicts the probability of occurrence of the class of unknown data sets [3].

KNN [13] is one of the simplest Supervised algorithms. KNN assumes that there is a similarity between the new data point and available data points and puts the new data point into the category which is very similar to the available categories. Here k is a positive integer and decides how many neighbors influence the classification. "Euclidean Distance" or "Manhattan Distance" is the distance metric that defines the "Closeness".

6. SYSTEM ARCHITECTURE AND IMPLEMENTATION

6.1 System Architecture

The hypothyroid prediction block diagram is shown in Fig. 1. Before developing a predictive model, we need to Pre-Process the data. I filled the NaN values with the spline interpolation. interpolate() function is basically used to fill missing values within the dataframe. It uses various interpolation techniques to fill the missing values instead of hard-coding the value. Sometimes '?' has been used instead of 'nan', so replace it with NaN. Here, we can see that the feature column "TBG" contains an extremely high number of null values. So, I will not be using this column for my model. For the classification is important that the dataset only has numerical attributes, so I have to encode the categorical values into numerical values. All 't' is encoded with 1, all 'f' is encoded with 0. 'F' in the sex column is replaced with 1 and 'M' with 0. Four categorical features of referral source have been one hot encoded.

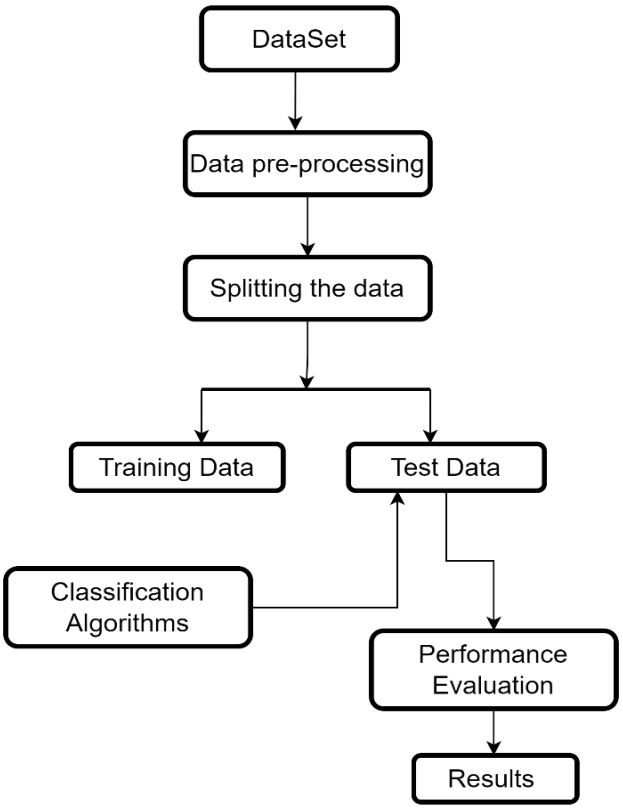

Fig -1: System Architecture

6.2 Implementation

All the models are generated in Python with the use of the scikit-learn library which is one of the most popular open-source library in ML. Table 3 shows python function descriptions and parameters chosen for implementation.

Table -3: Model implementation parameters

Python function	Parameters	Explanation
XGBClassifier()	learning rate=0.01	increasing the learning rate will increase learning speed [17]
	use_label_encoder=False	not use scikit-learn's label encoder [17]
CatBoostClassifier()	max_depth=4	The maximum depth to be used for the Decision Tree algorithm [17]. The optimum value for max_depth ranges from 4 to 10. Its default value is 6.
AdaBoostClassifier()	n_estimators= 100	A maximum number of estimators to perform learning [17].
LGBMClassifier()		It is implemented with all default parameters
KneighborsClassifier()	n_neighbors=3	3 is the number of neighbors to use
DecisionTreeClassifier()	class_weight="balanced"	It shows the weights associated with classes. Weights are automatically adjusted with this setting [17]
	max_depth=5	It represents the maximum depth of the tree [17]
GaussianNB()		It is implemented with all default parameters

7. EVALUATION CRITERIA AND VALIDATION

The goal of this work is to classify thyroid disease by the use of different machine learning approaches. The results of different models are analyzed and compared on the following evaluation criteria.

7.1 Classification Accuracy (ACC)

Accuracy is defined in terms of positives and negatives by the following equation [16]. It ranges from 0 to 100(%)

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

Here, TP =True Positive, TN = True negative, FP = False Positive, FN = False Negative [16].

7.2 Precision

It shows the model's positive prediction quality which is defined by the following equation [16].

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

7.3 Recall

Recall gives the information of missed positive prediction numbers. It is defined by the following equation [16].

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

7.4 F1 Score

Many real-life classification problems have imbalanced class distribution. Accuracy is used when TP and TN are more important but F1-score is used when the FN and FP are critical. Thus F1-score could be a better metric to judge our model on. The F1-Score is the harmonic mean of precision and recall [16].

$$\text{F1 score} = \frac{2(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})} \quad (4)$$

7.5 Validation

Model validation is necessary for machine learning. It will help us to evaluate how well our machine learning model goes to react to new data. So, we use cross-validation to obtain a more reliable estimate of performance metrics [14]. In k-fold cross-validation, we split the input data is divided into k subsets. Then train ML model on all but (k-1) subsets, and then evaluate the model on the subset that was not used for the training process. Repeat the process for k times, with a different subset reserved for evaluation each time. k-fold cross-validation procedure can be very effective in general but sometimes gives misleading results and fail when used on classification problems with a severe imbalance class distribution. As I am dealing with thyroid datasets which are highly imbalanced positive and negative class. The

techniques must be modified to stratify the sampling by the class label, called stratified k-fold cross-validation. It is the same as K fold cross-validation, just a slight difference is there. It maintains the same class ratio throughout the K folds as the ratio in the original dataset as shown in Fig. 2.

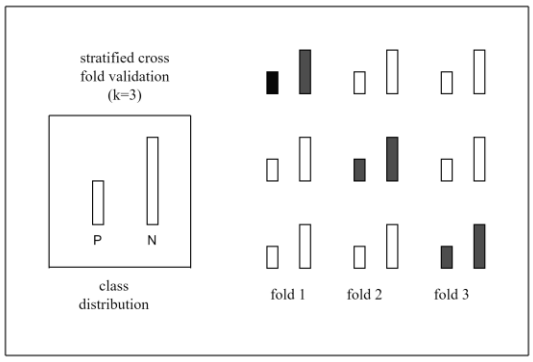

Fig -2: Stratified k-fold cross-validation

I have used RepeatedStratifiedKFold (n_splits=5,n_repeats=2,random_state=0) function. 5 splits and repeating 2 times is perfectly enough to perform validation.

8. RESULTS AND DISCUSSION

The experiment was implemented in ANACONDA as software and python programming language on Windows 64-bit operating system with AMD Ryzen 5 4500U 2.38 GHz and 8 GB of RAM. The confusion matrix with different performance measures of deployed classifiers is shown in Table 4.

Table -4: Confusion Matrix Comparison

Classifier		**Actual**	
NB	**Predicted**	75	2(FP)
		825(FN)	202
KNN	Predicted	34	43
		18	1009
Decision tree	Predicted	77	0
		3	1024
Catboost	Predicted	77	0
		3	1024
LightGBM	Predicted	76	1
		2	1025
XGBoost	Predicted	77	0
		1	1026
Adaboost	Predicted	77	0
		37	990

Three TD(Thyroid) patients were misclassified as healthy controls in the decision tree and CatBoost. Only one person is misclassified as a hyperthyroidism patient however he/she belonged to healthy control in LightGBM. CatBoost, Decision Tree, Adaboost, and XGBoost have not misclassified any healthy person as a hypothyroidism patient. After performing data preprocessing, there were 3678 samples. They are divided into 2574 training and 1104 testing samples with 80:30 train test split ratio. In NB out of 1104, 825 patients were wrongly classified healthy which was a very poor classifier for TD prediction. For the problem used in this paper FP and FN numbers have to be less. Results of Table 4 shows that Decision Tree, CatBoost, LightGBM, and XGBoost has low numbers of FP and FN.

The comparison of accuracy obtained on the deployed classifiers is shown in Table 5. In this XGBoost obtained the highest accuracy of 99.91% followed by LightGBM, Decision tree, Adaboost, KNN, and NB which obtained the accuracy of 99.73%, 99.64%, 96.65%, 94.47, and 25.09% respectively. Accuracy is used when TP and TN are more important but F1-score is used when the FN and FP are critical. In my problem, imbalanced class distribution exists and thus F1-score is a better metric to evaluate the model. An F1 score is considered perfect when it's 1. XGBoost has the highest F1 score which is 0.9965 amongst all classifiers. Thus XGBoost outperformed the other models in terms of accuracy, precision, recall, and F1 score. This shows that the XGBoost classifier is significantly more effective in TD prediction.

Table -5: Comparison of all Classifiers

Classifier	**Accuracy**	**Recall**	**Precision**	**F1 score**
NB	0.2509	0.5854	0.5368	0.2409
KNN	0.9447	0.7120	0.8065	0.7489
Decision tree	0.9964	0.9920	0.9805	0.9862
Adaboost	0.9665	0.982	0.8377	0.8942
XGBoost	**0.9991**	**0.9995**	**0.9936**	**0.9965**
LightGBM	0.9973	0.9925	0.9867	0.9896
Catboost	0.9973	0.9985	0.9812	0.9897

Sometimes you want to optimize your model over speed in real data science project. The running times of the algorithms are compared in Table 6.

Table -6: Comparison of all Classifiers on the basis of Speed

Classifier	**Speed(fit) in s**
NB	0.0923
KNN	0.5453
LightGBM	0.7329
DT	1.074
XGBoost	1.119
AdaBoost	4.1251
CatBoost	13.3974

NB and KNN take lesser time to train but their accuracy is very poor as seen from Tabel 5. In our case, LightGBM took 0.7029s which is faster than the most accurate XGBoost. Catboost took 13.3974s which is the slowest type of implementation. In my paper, I prefer between XGBoost and LightGBM. LightGBM is faster and gives accuracy nearer to XGBoost.

9. CONCLUSION AND FUTURE WORK

Thyroid Detection using Machine Learning is a smart and precise way to predict thyroid disease. The first work is collecting the data from the UCI repository, then analyzing it with exploratory analysis where I found insights from the data, then the data was cleaned and transformed for prediction. NB, KNN, LightGBM, DT, XGBoost, AdaBoost, and CatBoost have been implemented and precision, recall, accuracy, F1 score were used to evaluate the implemented models' performance. XGBoost classifier did pretty well achieving the highest accuracy. Other evaluation metrics also support the performance of this algorithm. I thereby recommend the XGBoost classifier for the predictive model. The running times of the algorithms are compared. CatBoost took the highest amount of time. LightGBM took less time than XGBoost. The optimal model that should be used for this dataset is LightGBM for fast results and XGBoost for a higher accurate model.

To predict thyroid disorder, all classifiers produce good results except NB. In the future, these algorithms can be implemented for the prediction of thyroid disease with more real data related to thyroid and with multiple classes.

REFERENCES

[1] 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(2.8):315, March 2018.

[2] Tyagi Ankita, Mehra Ritika, "Interactive Thyroid Disease Prediction System Using Machine Learning Technique," 5th IEEE International Conference on Parallel, Distributed and Grid Computing, pp. 689- 693, 2018

[3] K. Rajam, R. Jemina Priyadarsini, "A Survey on Diagnosis of Thyroid Disease Using Data Mining Techniques", International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 5, Issue. 5, May 2016, pg.354 – 358.

[4] S.B Patel, P. K Yadav, Dr. D. P.Shukla," Predict the Diagnosis of Heart Disease Patients Using Classification Mining Techniques", (IOSR-JAVS), e-ISSN: 2319- 2380, p-ISSN: 2319- 2372. Volume 4, Issue 2 (Jul. - Aug. 2013), Pg.no 61-64.

[5] Patel Hetal. An Experimental Study of Applying Machine Learning in Prediction of Thyroid Disease. International Journal of Computer Sciences and Engineering, Vol. 7, Issue 1, pp. 130-133, 2019.

[6] Yadav Dhyan, Pal Saurabh, "To Generate an Ensemble Model for Women Thyroid Prediction Using Data Mining Techniques," Asian Pacific journal of cancer prevention, Vol. 20, Issue 4, pp.1275-1281, 2019.

[7] K. Rajam, R. Jemina Priyadarsini, "A Survey on Diagnosis of Thyroid Disease Using Data Mining Techniques", International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 5, Issue. 5, May 2016, pg.354 – 358.

[8] D. T. Larose, Discovering knowledge in data: An introduction to data mining, John Wiley & Sons, (2005) 385

[9] Yadav Dhyan, Pal Saurabh, "To Generate an Ensemble Model for Women Thyroid Prediction Using Data Mining Techniques," in Asian Pacific journal of cancer prevention, Vol. 20, Issue 4, pp.1275-1281, 2019.

[10] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system." in Proceedings of the 22Nd ACM SIGKDD International Conferenceon Knowledge Discovery and Data Mining, KDD '16, pages 785–794, NewYork, NY, USA, 2016. ACM

[11] John T. hancock, Taghi M. Khoshgoftaar, "CatBoost for Big Data : An interdisciplinary review" in Journal of Big Data, (2020) 7:94 https://doi.org/10.1186/s40537-020-00369-8

[12] Guolin Ke, Qi Meng, Thomas Finley, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree", in 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA

[13] Pratiksha Chalekar, Shanu Shroff, Siddhi Pise, SujaPanicker, "Use of k-Nearest Neighbor in Thyroid disease classification", in International Journal of Current Engineering and Scientific Research

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

[15] Ozyilmaz, Lale, and Tulay Yildirim, "Diagnosis of thyroid disease using artificial neural network methods" Neural Information Processing,. Proceedings of the 9th International Conference on. Vol. 4. IEEE, 2002.

[16] Yasir, Sonu Mittal, "Thyroid disease prediction using hybrid ML technique", in International Journal of scientific and technology research, Vol. 9, Issue 2, February 2020.

[17] Python Software Foundation, Python Language Reference, Documentation, version 3, available at http://www.python.org

BIOGRAPHY

Vishakha earned her Masters degree in Communication Systems from the National Institute of Technology (NIT), Surat. She has been associated with various educational institutes in the capacity of Assistant Professor and research.

During her professional stints, she has contributed to R. V. College of Engineering in Bangalore as well as National Institute of Technology in Surat. Vishakha earned her bachelors in Electronics and Communication.

She is passionate about Machine Learning, Artificial Intelligence, speech and audio signal processing, and Digital Signal Processing.