Heartbeat Signal Classification via Ensemble Learning

Feihong Hong

School of Economics and Management, Beijing Jiaotong University, No.3, Shangyuan Village, Haidian District, Beijing, China

* Corresponding Author Email: 19711010@bjtu.edu.cn

Abstract. As the enhancement of life conditions, the incidence of heart-related diseases is increasing now. And ECG data is a valuable and easily accessible heart data that can predict heart-related disease occurrence. Hence, it is of great importance to classify heartbeat signal data to prevent the negative effect of heart diseases. This paper proposed a multi-classification model via ensemble learning, integrated learning, and SMOTE sampling. By testing and validating the model on the ECG dataset, our model achieves lower metric scores and better prediction performance, accurately classifying heartbeat signals on a dataset with 250 thousand records.

Keywords: Integrated learning, Model Ensemble, SMOTE sampling.

1. Introduction

With the rapid development of the social economy and enhancement of residents' life quality, a significant change has been made in every aspect of urban residents' lives, especially clothing, diet, housing, and commuting. It is suitable for the enrichment of physical and emotional life. However, this change also brings about a new problem. According to the "China Cardiovascular Disease Report" [1] released by the National Center for Cardiovascular Diseases, deaths from cardiovascular diseases remain the leading cause of death among Chinese residents. The high mortality rate of cardiovascular diseases has made it one of the hot spots in the medical community today. Cardiovascular disease results from the joint action of multiple risk factors such as genetic genes and behavioral habits, and the complex interaction of multiple risk factors often multiplies the risk of cardiovascular disease. In cardiovascular diseases that should not be underestimated, arrhythmias are the most common and essential group of heart diseases, which are an essential factor inducing heart disease, so it is of great significance to detect patients' arrhythmias as early as possible and accurately for the prevention and diagnosis of palm soldiers.

With the development of intelligent information technology, the auxiliary analysis of medical diagnosis through computers has become a reality and has been widely used. Electrogram is a bioelectric signal reflecting the electrical activity of the human heart, with low frequency, low amplitude, non-invasive and other characteristics, and has been widely used in the clinical diagnosis of cardiovascular diseases. The employment of computationally assisted analysis techniques to automatically classify ECG signals. Computer-aided diagnostics in the field of ECG are of great help in diagnosing cardiovascular diseases more effectively and improving accuracy. Determining that ECG signals belong to that type provides strong technical support for the prevention, timely diagnosis, and treatment of cardiovascular diseases, dramatically reduces the workload of doctors, reduces misdiagnosis and missed diagnosis and additional medical expenses, and is of great significance for the diagnosis of diseases in clinical medicine.

This paper puts forward a method based on integrated learning and model ensemble to complete the heartbeat signal classification task. From the outset, preprocess the original dataset to reduce memory usage. Then, conduct EDA to check on the sample and feature. Third, solve the problem found in the EDA process. Fourth, make feature engineering including feature processing and encoding. Fifth, build and evaluate three integrated learning models based on three classical boosting algorithms. Last, do model ensemble and output the predicted result. The flow chart below shows the solution mentioned above:

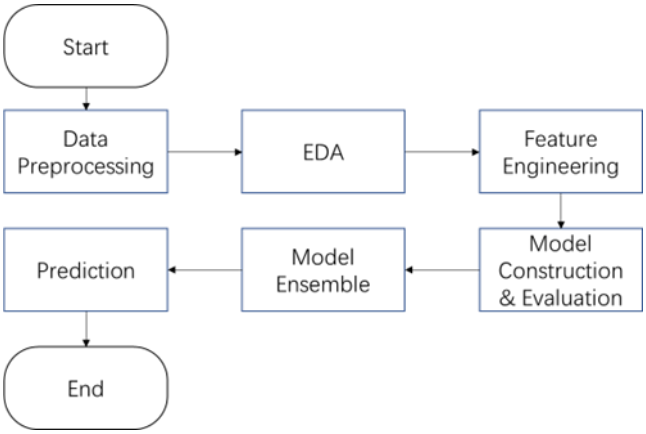


Figure 1. The plan of the paper

2. Dataset

The dataset used in this article is from the ECG data record of a platform, mainly a column of heartbeat signal sequence data, in which the signal sequence of each sample is sampled at the same frequency and of equal length. There are 100,000 data records in the training set and 20,000 in the test set. Since this paper focus on the classification method, exact names of heartbeat signals would be replaced by number label from 0 to 3.

By knowing the character of each column, we can improve our understanding of the dataset and conduct a better analysis later. The table below illustrates the characters of columns in this dataset:

Table 1. Field Description for the Heartbeat Signal Dataset

Field	Description
id	The unique identifier assigned for each heartbeat record
heartbeat_signals	The heartbeat signal serial
label	The number representing the type of the signal

The id column is the unique identifier assigned for each record in the dataset. No benefit can be obtained from analyzing the id column. The column named heartbeat_signals stores information about the numerical heartbeat record in each interval. This column can be transferred to a time-series data sequence for further analysis. The visualization of some records is illustrated in the following figure:

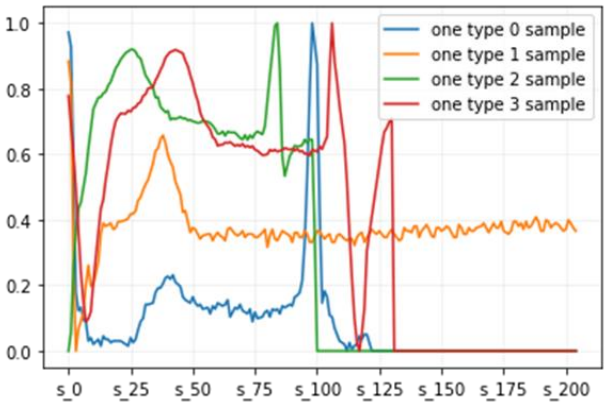


Figure 2. Visualization of some heartbeat signal series

The figure above shows the difference deviation among different types of signals as the time goes by from 0 to 1. Therefore, it would push the following data analysis via transferring the heartbeat signal column.

The column named label is the classification name of the ECG signal, which belongs to the category variable. There are 4 ECG signal categories in this data set, with 0,1,2,3 as a label. The label column can be converted to a numeric type for subsequent analysis and calculation.

3. Method

3.1. Integrated Iearning

Integrated learning, a way of machine learning, complete the learning task via building and combining a host of learners. It aims to achieve higher prediction performance than that of a single learner.

A general process of achieving integrated learning is to build a group of base learners and then combine them. The performance of the integrated learning model depends on two key aspects. One aspect is training each base learner, and the other combines these trained base learners. According to diverse responses to these two aspects, two prevailing frameworks exist: boosting and bagging.

Bagging is a way of improving a group of weak learners into a solid learners. Its mechanism is: to train a base learner from the original training set and then make modifications to the distribution of samples by the performance of the base learner. This modification can let these training samples receive more concern in the next round. Then train the next learner according to the modified sample distribution and conduct another modification of sample distribution until the number of base learners achieves some given number or satisfies the condition to end. Then combine these learners via weights. XGBoost, LightGBM, and CatBoost, employed by this paper, are all classical algorithms based on boosting framework.

3.2. XGBoost

XGBoost is an open-sourced Gradient Boosting framework created by Tianqi Chen [2]. It has strong performance in the efficiency of parallel computing, missing value disposal, and prediction.

a. Base learner

XGBoost is based on the CART decision tree. CART [3] assume that the decision tree is binary with a yes-no value in the internal node, where the left branch is valued by yes, and the right one is valued by no. CART is equivalent to recursively doing dichotomy to each feature to divide the feature zone into finite units and determine the probability distribution of these units. CART algorithm contains two steps: decision tree generation and decision tree pruning.

The pseudocode for building the CART decision tree

```
Def dichotomy(node N):
    For each feature f in F:
        initialize best gain g_best, best feature f_best, best feature value x_best;
        for each feature value x_k in {x_1, x_2, ..., x_k}:
            divide the set D into D_1 and D_2;
            compute the division gain g_k;
            if g_k > g_best :
            then g_best = g_k, f_best = f_k;
        return f_best, x_best;
def BinaryDevision(node N):
    call dichotomy(N);
    Compute Gini index in f = x_best;
    D_1(f, x_best) = {x|x^(f) = x_best};
    D_2(f, x_best) = {x|x^(f) ≠ x_best};
    return D_1, D_2;
While the condition to exit is not satisfied:
    For the father node N:
        BinaryDevision(N);
    For two son node N_1 and N_2:
        BinaryDevision(N_1);
        BinaryDevision(N_2);
```

The pseudocode for pruning the CART decision tree

initialize $k = 0,\ T = T_0$;
def disposal:
  let $\alpha = +\infty$;
  for each inner node t:
    compute prediction error $C(T_t)$, *the leave node of tree* T_t:$|T_t|$;
    compute $g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1},\ \alpha = \min(\alpha, g(t))$;
    prun t satisfying $g(t) = a$ *and identify the type of t via majority voting*
  $k = k + 1,\ \alpha_k = \alpha, T_k = T$;
if T_k *NOT the tree by the root node/two leaves*
  call disposal;
else $T_k = T_n$
cloose best tree T_α *in* $\{T_0, T_1, \cdots, T_N\}$ *by cross validation*

b. Model Expression

The idea of this algorithm is to keep adding trees and growing them via feature separation. Each time a tree is added, a new function is learned to fit the residual of the last prediction. K trees are obtained after completing the training. For each data sample, a score is predicted by the k trees. The mathematical expression for this model is:

$$\hat{y} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \tag{1}$$

$$where\ F = \{f(x) = w_{q(x)}\}(q: R^m \to T, w \in \mathrm{R}^T,)$$

$w_{q(x)}$ is the score of the leave node q and $f(x)$ is the mathematical expression of one classification tree.

c. Objective Function

$$\text{Obj} = \sum_{i=1}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

The formula measures the difference between the predicted and real scores on the left-hand side and the regularization term. The regularization term also includes two parts. T refers to the number of leave nodes, w refers to the score of leave nodes, γ controls the number of leave nodes, and λ can prevent the score of leave nodes from becoming relatively larger to avoid overfitting.

According to the additive model, the prediction score can be expressed as below after generating t trees:

$$\widehat{y_i^{(t)}} = \widehat{y_i^{(t-1)}} + f_t(x_i) \tag{3}$$

Use Taylor's second-order expansion to approximate the objective function in $f_t(x) = 0$. Denote the first-order derivative and second-order derivative of $l(y_i, \hat{y_i})$ to be g_i and h_i respectively.

$$g_i = \partial_{\widehat{y^{(t-1)}}} l\left(y_i, \widehat{y^{(t-1)}}\right), h_i = \partial^2_{\widehat{y^{(t-1)}}} l\left(y_i, \widehat{y^{(t-1)}}\right) \tag{4}$$

Abandon the constant $l\left(y_i, \widehat{y^{(t-1)}}\right)$, the object function is reduced to:

$$\widetilde{\mathcal{L}^{(t)}} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{5}$$

d. Optimization

The optimization methods of XGBoost mainly include the exact greedy algorithm and the approximate algorithm. Under the set learning conditions, the exact greedy algorithm calculates the

gain of all possible node divisions at each split of the decision tree and selects the node division with the maximum gain. The exact greedy algorithm is very accurate but time-consuming. The approximation algorithm compresses the number of features and candidate segmentation points for each feature through column sampling and weighted quantization and uses minor precision loss to obtain a better index score.

3.3. LightGBM

a. Model expression:

The LightGBM has the same mathematical expression as XGBoost.

b. Optimization

LightGBM optimizes the XGBoost algorithm through the number of split points, the number of samples, and the number of features.

To solve the problem caused by too many split points, LightGBM [4] adopts a Histogram algorithm to bucket the features. The continuous characteristic values are discrete into k integers, using floating feature values. At the same time, construct a histogram with the width k. According to the feature bin, accumulate its gradient and count its number. After traversing the data, the histogram gets the needed statistics. Then, according to the discrete value of the histogram, the optimal segmentation point is found by traversing the data.

LightGBM employs Gradient-based One-Side Sampling to solve the problems that three exist too many samples. This algorithm excludes most samples with relatively small weight and uses the rest of the sample to compute the gain. It balances the reduction of data and making sure the accepted accuracy.

LightGBM uses Exclusive Feature Bunding. Exclusive Feature Bunding reduces the dimension of features by means of bunding features.

LightGBM uses leaf-wise with depth limitation algorithm too. This algorithm finds the leave with the maximum gain among all leave nodes and then splits from this leave.

3.4. CatBoost

Russian search giant Yandex developed CatBoost in April 2017. Based on the GBDT algorithm framework, CatBoost can better deal with category-based features. At the same time, the feature dimension is greatly enriched by the combination category feature. CatBoost [5] uses more effective strategies to reduce overfitting and uses the entire data set to participate in training, efficiently using data information.

Assume the observation dataset $S = \{(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n)\}$, where $X_i = (x_{i1}, x_{i2}, \cdots, x_{im})$ is the m-dimension vector composed of numerical and categorical features, and Y_i is the label.

First, the CatBoost algorithm performs binarization processing on all numerical features. Using the oblivious tree as a base learner and binarize the float feature, statistical information, and one-hot code.

Then, transform categorical features into numerical ones:

Randomly arrange observations to generate multiple random sequences

Given a sequence, use the average marker of the training data set to replace corresponding categories:

$$x_{ik} = \frac{\sum_{j=1}^{n} [x_{jk} = x_{ik}] \cdot Y_j}{\sum_{j=1}^{n} [x_{jk} = x_{ik}]} \tag{6}$$

In this formulation, $[x_{jk} = x_{ik}] = 1$ if $x_{jk} = x_{ik}$ and equals 0 otherwise. The same category is placed before the given value in the permutation.

Let $\theta = (\theta_1, \theta_2, \cdots, \theta_n)$, transform categorical features to numerical value:

$$x_{\theta_{\mathrm{p}},k} = \frac{\sum_{j=1}^{\mathrm{p}-1}\left[x_{\theta_{\mathrm{p}},k}=x_{\theta_{\mathrm{p}},k}\right]\cdot Y_{\sigma j} + \mathrm{a}\cdot \mathrm{P}}{\sum_{j=1}^{\mathrm{p}-1}\left[x_{\theta_{\mathrm{p}},k}=x_{\theta_{\mathrm{p}},k}\right] + \mathrm{a}} \tag{7}$$

In the formulation above, the prior value P and parameter a (a>0) are added to help reduce the noise from the low-frequent category.

Finally, CatBoost algorithm uses a greedy strategy for feature combination:

(1) no combination is performed for the first split of the tree.

(2) At the second split, all the existing combination and classification features in the current tree and all the classification features in the data set are combined, and the combined value is instantly converted into numbers.

(3) All splits selected in the tree are treated as classifications with two values and used in combination to generate combinations of numeric and classification features.

3.5. Evaluation Metric

This paper employs the sum of absolute error of all data, called abs-sum for short in the following section, as the metric. For any given signal, if the actual values of the label are $[y_1, y_2, y_3, y_4]$ and the probability prediction values from the model are $[a_1, a_2, a_3, a_4]$, then the value of abs-sum for this model is:

$$abs - sum = \sum_{j=1}^{n}\sum_{i=1}^{4}|y_i - a_i| \tag{8}$$

4. Experiment and Result

In this section, this paper uses Python code to complete the classification task of the heartbeat signal.Here is a sample illustration and caption for a multimedia file:

4.1. Exploratory Data Analysis (EDA)

This part will conduct an overall analysis of the dataset employed to explore the features of data and data dimensions. This part can improve the overall data quality and boost algorithm speed and accuracy.

In the first place, split the object feature in the dataset to transform it into a numerical feature. The figure below shows the result of the transformation.

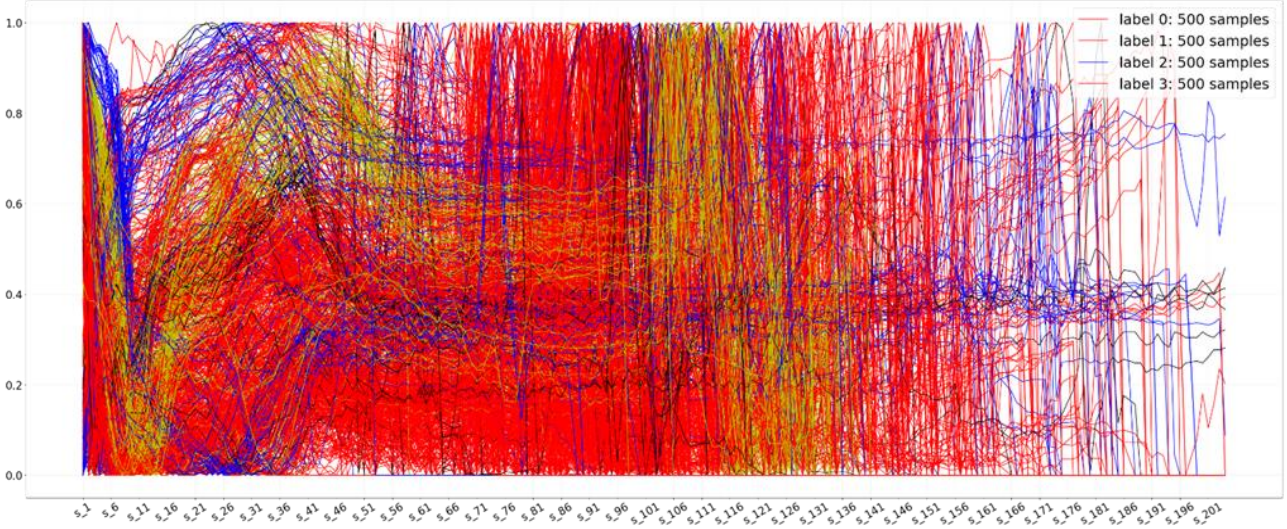


Figure 3. The Visualization of part Heartbeat Signal after feature Transformation

From figure 3, we can see that each type of signal has its unique but complex series of features. The following analysis should focus on the disposal of features.

The training set has 100000 records and 207 columns in the original data set. The test data consisted of 20,000 records and 206 columns.

Conduct a descriptive analysis for the split signal feature and visualize their standard deviation, maximum and minimum. The results of the visual descriptive analysis are shown below:

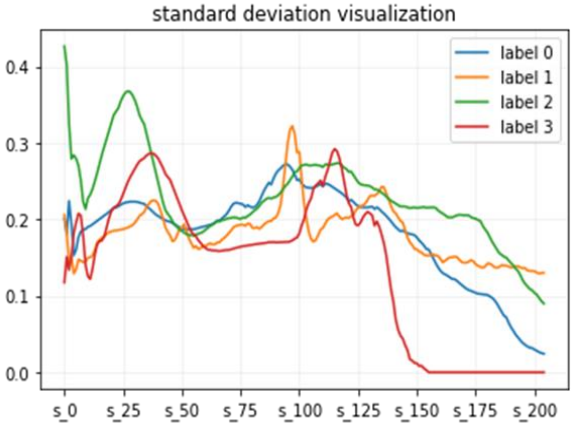


Figure 4. The visualization of Heartbeat Signal's Std.

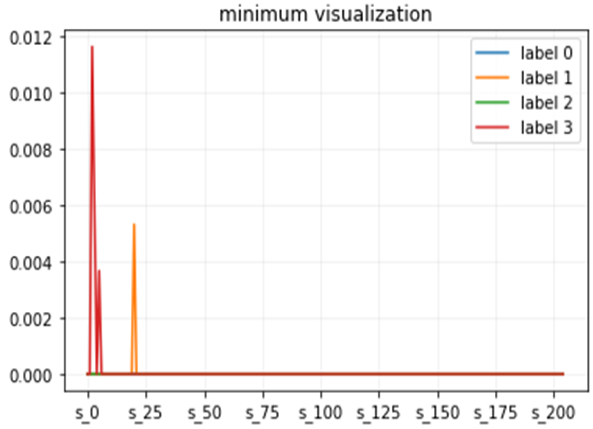


Figure 5. The visualization of Heartbeat Signal's Min

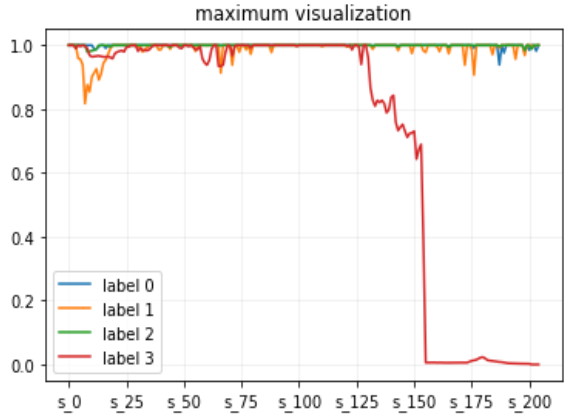


Figure 6. The visualization of Heartbeat Signal's Max

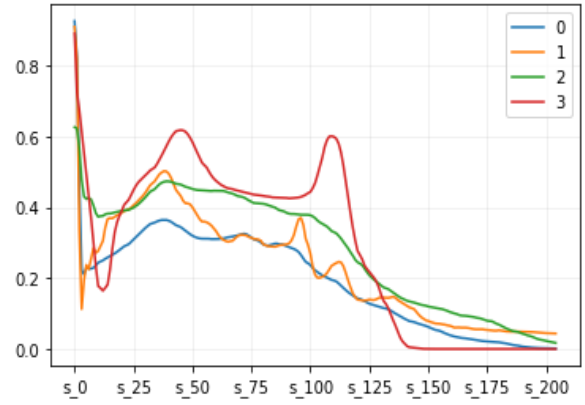


Figure 7. The visualization of Heartbeat Signal's Mean

The four figures above show some primary statistic data change with time of 4 signal types. For the statistical maximum and minimum shown in Figures 5 and 6, we can observe that the feature value of all signals falls into the range from 0 to 1. The mean and standard deviation, reflected in Figures 4 and 7 above for each feature value, illustrates that each feature has a significant difference.

From the result of the visualization, we can find:

- The primary feature data is one-dimensional signal amplitude, which has been normalized from 0 to 1, with a length of 205.
- There exists no accessible auxiliary or prior information. Ideally, this dataset has no missing value and needs to fill it.
- The default amplitude of uncollected signals is 0, so there is no problem with missing values.

According to the division of numerical and discrete features, the label is the only discrete variable. The distribution of the label is illustrated as follows.

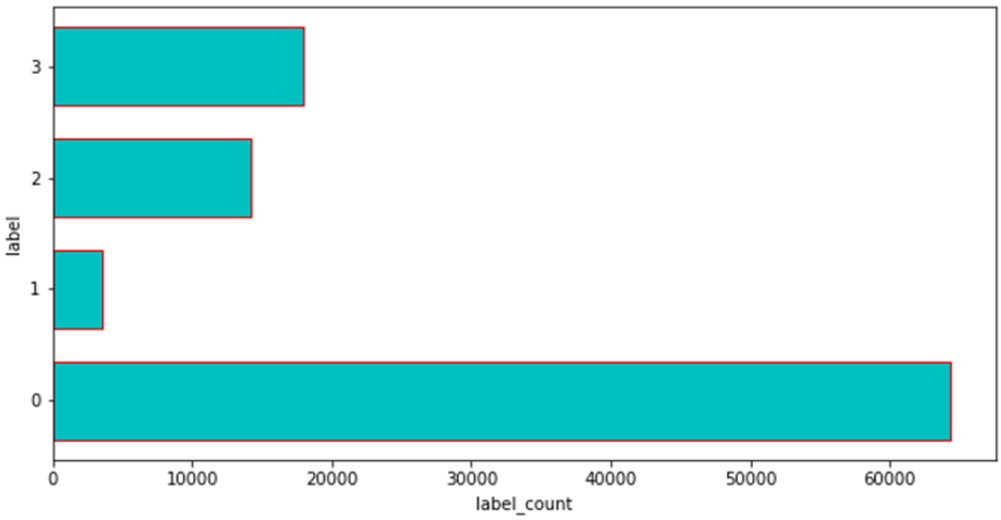


Figure 8. The Distribution of the Value of Label

From figure 8, the problem of category imbalance is undeniable. Label 0 accounts for most of the data records, while the sum of another label's records is relatively distant from label 0.

To address this problem, SMOTE algorithm [6] is employed in the sample with the label 0,1,2 to perform oversampling. The label of the train set after oversampling is shown as follows.

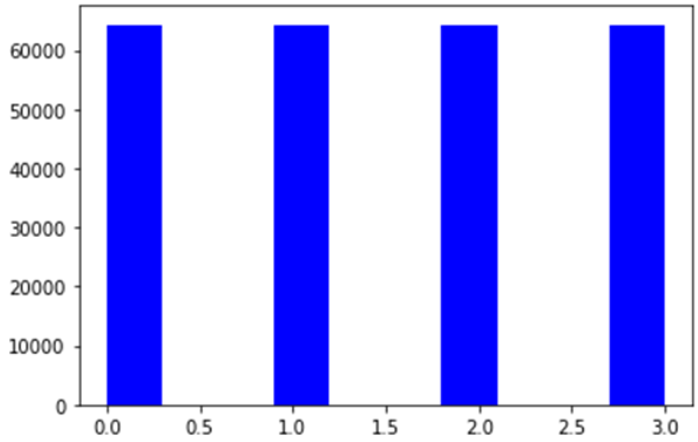


Figure 9. The Label Distribution after Oversampling

From figure 9, the imbalance disappears. Each label has the identical data records over 60000.

4.2. Feature Engineering based on EDA

Dispose of the feature first. Since this dataset has no missing value, missing value padding is unnecessary.

In terms of dealing with object type features, this work has been done in the EDA part. The object type feature heartbeat_signal has been transformed into a series signal with the length of 205 and the dimension of 1.

The analysis and oversampling of the label value in the EDA part have been finished for dealing with categorical features.

4.3. Model Construction Based on Feature Engineering

This part will use python code to conduct multiclass analysis for the dataset by achieving XGBoost, LightGBM, and CatBoost algorithms.

a. The multiclass classification and prediction based on XGBoost

Table 2. The Parameter of XGBoost

Parameter	Description	Value
booster	The base model employed	gbtree
objective	The objective of task	multi:softprob
num_class	The number of class	4
eval_metric	The evaluation metric	mlogloss
max_depth	Max depth of the tree	8
alpha	L1 regularization term	30
subsample	Random sampling ratio	0.7
colsample_bytree	The percentage of random samples per tree	0.5
colsample_bylevel	The proportion of samples taken by split columns at each level of the tree	0.5
eta	Learning rate	0.1
tree_method	Ways of data processing	gpu_hist

Using the above parameters, we train the multi-classification model based on XGBoost. We use a GPU-powered histogram algorithm to generate base trees and want this model to output the predicted probability of each signal type.

b. The multiclass classification and prediction based on CatBoost

Table 3. The Parameter of CatBoost

Parameter	Description	Value
learning_rate	Learning rate of the algorithm	0.15
depth	The depth of the tree	6
l2_leaf_reg	L2 regularization parameter	10
od_type	Type of overfitting check	Iter
random_seed	Random seed used in training	11
allow_writing_files	Whether the program is allowed to write analytical and snapshot files during training	False
iterations	The max number of trees	10000
task_type	The device used to perform the task	GPU
devices	The number of the device used	0:1
loss_function	The type of loss function	MultiClass

We train the multi-classification model powered by CatBoost and GPU using the parameter listed in the table above. This model stops training early in about 8000 iteration rounds in the training process. 6 is the depth limited to avoid overfitting.

c. The multiclass classification and prediction based on LightGBM

Table 4. The Parameter of LightGBM

Parameter	Description	Value
boosting_type	Type of boosting	gbdt
objective	The type of task	multiclass
num_class	The number of classes	4
num_leaves	The number of leave nodes	2**5
feature_fraction	Feature selection ratio in generating trees	0.8
bagging_fraction	The sampling ratio in tree generation	0.8
bagging_freq	The iteration times k between boosting	4
learning_rate	Learning rate of the algorithm	0.1
nthread	The number of threads allowed to work	-1

LightGBM shows its quick model convergence ability in a 250000+ records training set without using GPU to speed up training. Use GBDT, aka gradient boosting tree, to compute the loss. It stops early before achieving the max iteration times.

d. Model Evaluation

Divide the dataset into two parts: one part is the training set, and the other is the validation set. The training set is used to train the model, and the validation set is employed to evaluate the discriminant ability of the model for new samples.

Since the dataset processed by SMOTE oversampling has data records over 250 thousand, use 5-fold cross-validation and abs-sum of the average as evaluation method and evaluation metric.

For different models, five training sets with different divisions were used for training, and then the verification was performed on the verification set. The ABS-sum between the real value and the predicted value of the verification set was calculated. Get the average value of the five cross-verification. The evaluation results of three trained models are shown below.

Table 5. The Evaluation Results of three Classification Models

Algorithm Used	The avg. *abs_sum*	The Std. of *abs_sum*
XGBoost	2289.87	35.82
CatBoost	834.08	83.2
LightGBM	266.87	30.78

This table shows that the LightGBM has the lowest loss and deviation due to its innovative ways of feature disposal and sampling. CatBoost ranks second and XGBoost ranks third, respectively. These two models have lower performance than LightGBM because of the large dataset.

e. Model Ensemble

Use a weighted average method to fuse the three models. The magnitude of average abs-sum gives large weight to the model with less average abs-sum. Weights of XGBoost, CatBoost and LightGBM are 0.2, 0.3 and 0.5 respectively.

f. Prediction

Use the ensembled model to predict the label in the test set and then save the predicted result.

5. Conclusion

5.1. The work of this paper

This paper conducts a multiclass classification task in the heartbeat signal context using ensemble learning and SMOTE sampling. This paper explores the dataset by EDA section, analyzing and visualizing the heartbeat signal. This paper also trains three tree-based integrated learning models to compare their performance. Taking advantage of abundant data records, the paper employs a 5-fold cross-validation strategy to accurately evaluate three models—the paper ensembles three models to get a better classification result.

5.2. The edge of this model

a. By using the ensemble learning model, the result similar to that of a single complex model is obtained by using a simple base classifier combination, which saves the time of parameter tuning and reduces the performance requirement of equipment.
b. Use SMOTE sampling to solve the label imbalance problem. The result proves that a significant improvement can reduce the classification error.

Table 6. The Evaluation Results of three Classification Models

Model	Evaluation score after SMOTE sampling	Evaluation score before SMOTE sampling	Change in the score (%)
XGBoost	2289.87	1875.48	+22.2%
CatBoost	834.08	1014.63	-17.8%
LightGBM	266.87	583.42	-54%

From the table, we can see LightGBM has the most significant improvement after using the dataset by SMOTE sampling. In addition, CatBoost also has better performance using SMOTE-powered data. Moreover, XGBoost cannot seize the benefit of SMOTE mainly because the booming size of the dataset plus plentiful features bring about the bad performance.

c. Use model fusion to combine widely different models to obtain a more comprehensive classification and prediction ability.

5.3. Deficiencies and Prospects

a. The model is not tuned due to the bad performance of the experimental device. Using popular tuning methods such as grid search and Bayesian tuning may yield better classification models
b. The model does not carry out feature fusion. Adding a mean value feature or cross-term feature may further improve the classification ability of the model.

References

[1] Hu SHENGshou, Gao Runlin, Liu Lisheng, et al. Summary of The Chinese Cardiovascular Disease Report 2018. Chinese circulation hybrid Journal of Jilin University (Natural Science edition), 2019, 34(3): 209-220.

[2] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.

[3] Loh W Y. Classification and regression trees [J]. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2011, 1(1): 14-23.

[4] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree [J]. Advances in neural information processing systems, 2017, 30.

[5] Prokhorenkova L, Gusev G, Vorobev A, et al. CatBoost: unbiased boosting with categorical features [J]. Advances in neural information processing systems, 2018, 31.

[6] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique [J]. Journal of artificial intelligence research, 2002, 16: 321-357.