

# 第十一届“认证杯”数学中国

## 数学建模国际赛

### 承 诺 书

我们仔细阅读了第十一届“认证杯”数学中国数学建模国际赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们允许数学中国网站([www.madio.net](http://www.madio.net))公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为：

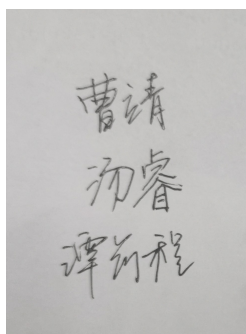
我们选择的题目是：

参赛队员（签名）：

队员 1：曹靖

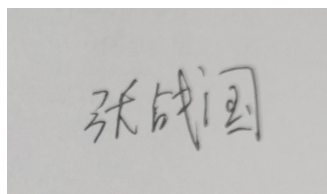
队员 2：汤睿

队员 3：谭前程



曹靖  
汤睿  
谭前程

参赛队教练员（签名）：张战国



张战国

# 第十一届“认证杯”数学中国

## 数学建模国际赛

### 编号专用页

参赛队伍的参赛队号：（请各个参赛队提前填写好）： 1040

竞赛统一编号（由竞赛组委会送至评委团前编号）：

---

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

## Human behavior classification based on multi-layer classifier model

**Abstract:** An important aspect of human behavior understanding is the recognition and monitoring of daily activities. Research on human activity recognition techniques based on human-computer interaction systems has attracted a lot of attention in recent years. In this paper, mathematical models based on data detected by wearable devices will be developed and 19 different human behaviors will be classified. And the validity and generalizability of the model will be evaluated.

For question 1: First, we consolidated the data into a csv file, considering the complexity of the data. Then the feature importance was calculated and 20 features with feature weights above 0.02 were selected to participate in the subsequent classification algorithm. Due to the large number of classification objects, using a single classification algorithm does not achieve perfect classification. At the same time, there is a certain similarity in human activity behavior itself, and we must do targeted research for similar behaviors. Therefore, our team proposes a multi-layer classifier model based on generalized discriminant analysis. The first layer of random forest classifier is used to classify all human activities. For behaviors that are easily confused in the results, the nonlinear features are then extracted using generalized discriminant analysis and given to the second layer SVM classifier for deeper classification. Finally, we compare with single classification algorithms such as KNN, logistic regression, etc. The results show that our multilayer classifier model has an accuracy rate of 92%, which far exceeds the accuracy of KNN and logistic regression.

For question 2: In the first question, only one validation set was used for our model validation, and the prediction accuracy may be high. Here, to further evaluate the generalization ability of the model, we used the K-fold cross-validation method to train the model. We divided the dataset into five copies, which were trained as the training and validation sets. The generalization ability of the model was evaluated by plotting the confusion matrix, calculating the recall, F1-score, etc. The results show that the generalized discriminant analysis-based multilayer classifier proposed by our team generalizes well.

For question 3: To explore and overcome the overfitting problem of the model, we plotted the learning curve of the multilayer classifier based on generalized discriminant analysis on top of the second question, and the curve shows that our model has good generalizability and does not have the overfitting problem.

**Key words:** Multi-layer classifier; Random Forest; Generalized Discriminant Analysis; SVM

# Contents

<b>1. Introduction</b>	5
1.1 Problem Background	5
1.2 Work	5
1.3 Data Preprocessing/Feature Extraction	5
1.4 Our Model	6
<b>2. The Description of Problem</b>	7
2.1 Analysis of question one	7
2.2 Analysis of question two	7
2.3 Analysis of question three	7
<b>3. Models</b>	8
3.1 Basic Model	8
3.1.1 <i>Terms, Definitions and Symbols</i>	8
3.1.2 <i>Assumptions</i>	8
3.2 Feature Extraction	8
3.2.1 <i>Extra Symbols</i>	8
3.2.2 Extraction of important features	9
3.3 Problem one modeling and solving	10
3.3.1 <i>Random Forest-based First-Level Classifier for Human Behavior</i>	10
3.3.2 <i>Feature mapping based on generalized discriminant analysis</i>	13
3.3.3 <i>SVM-based second layer classifier for human behavior</i>	15
3.4 Problem two modeling and solving	18
3.4.1 <i>Generalization capability</i>	18
3.4.2 <i>k-fold cross-validation</i>	19
3.4.3 <i>Selection of performance metrics</i>	20
3.4.4 <i>Confusion Matrix</i>	20
3.4.5 <i>Accuracy and F1-score</i>	22
3.4.6 <i>ROC curve</i>	23
3.5 Problem three modeling and solving	24
3.5.1 <i>Overfitting studie</i>	24
3.5.2 <i>Learning curve based model fitting problem determinatio</i>	24
<b>4. Strengths and Weakness</b>	25
4.1 Advantages of the model	25
4.2 Disadvantages of the model	25
<b>5. References</b>	26
<b>6. Appendix</b>	27

# **I. Introduction**

## **1.1 Problem Background**

The recognition and monitoring of daily activities is an important aspect of improving the understanding of human behavior. Today, wearable activity recognition systems have been rapidly developed and upgraded based on the rapid development of computers, improving the quality of life in many key areas such as dynamic monitoring, home rehabilitation and fall detection. We can use inertial sensor-based activity recognition systems to perform a variety of functions through personal alarm systems, such as remote monitoring and observation of the elderly, fall detection and classification, medical diagnosis and treatment, remote monitoring of children at home or school, rehabilitation and physical therapy, biomechanical research, ergonomics, sports science, ballet and dance, animation, film production, television, live entertainment, virtual reality, and computer games. Miniature inertial sensors and magnetometers located in different parts of the body can generate large streams of data about human activity that can be used to better understand human behavior and can be used to classify human activity.

## **1.2 Work**

1. Design a set of features and efficient algorithms to classify 19 human behaviors based on data from these human sensors.
2. Making the model generalize well with a limited dataset. We need to specifically study and evaluate this problem. and design a feasible method to evaluate the generalization ability of the model.
3. The overfitting problem is studied and overcome so that our classification algorithm can be widely applied to the problem of classifying people's behavior.

## **1.3 Data Preprocessing/Feature Extraction**

The data in this paper were obtained from measurements of human activity by miniature inertial sensors and magnetometers in different parts of the body. Sensor data were collected from a total of 8 subjects performing 19 different activities. The total signal duration for each subject for each activity was 5 minutes. The sensor unit was calibrated to acquire data at a 25 Hz sampling frequency. The 5-minute signal was

divided into 5-second segments, resulting in 480 ( $=60 \times 8$ ) signal segments for each activity.

We ended up with 9120 text files detailing the data measured by different sensors at the same time interval for different subjects performing different activities. We integrated these text data into a csv file with two columns for subject number and activity type.

By using a random forest-based feature importance ranking algorithm, important features are filtered out for subsequent classification.

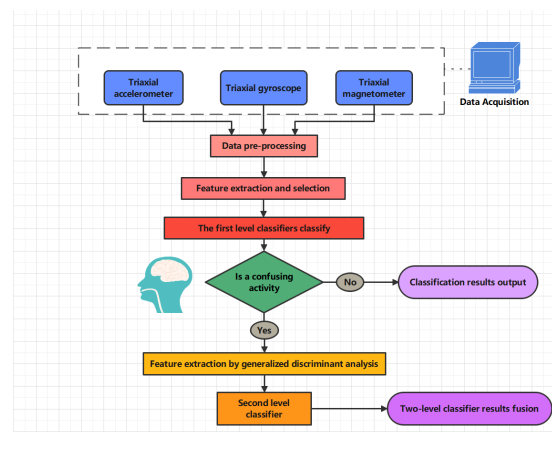
## 1.4 Our Model

After data pre-processing, based on the filtered features, our team designed a feasible algorithmic solution to classify 19 human behaviors. Due to the particularly large amount of data and the inherent similarity of human activities, direct classification of the 19 human behaviors using a single machine learning algorithm would easily result in confusion of similar behaviors and lead to degradation of classification accuracy. To solve this problem, our team proposes a multilayer classifier algorithm based on generalized discriminant analysis.

The algorithm flow is as follows:

- ①Data acquisition and pre-processing, feature extraction.
- ②The first level of classification identifies the full range of activities.
- ③Generalized discriminant analysis of similar activity characteristics.
- ④The second layer of classification identifies similar activities as well as the two-layer classifier weighted fusion.

The block diagram of the algorithm flow is as follows:



**Figure 1 Multilayer classifier based on generalized discriminant analysis**

## **II. The Description of the Problem**

### **2.1 Analysis of question one**

Since the problem involves an extremely large amount of data and a very large number of feature indicators. So we first process the data before performing feature extraction. Considering that the classification required by the problem is a 19-class segmentation task, the effect of using a single classification algorithm directly is definitely greatly reduced. Therefore, we propose a multilayer classifier algorithm based on generalized discriminant analysis. For the first layer of classifier we use random forest algorithm to classify all activities, and for the categories that are prone to misclassification in the classification results, we further use generalized discriminant analysis to extract the nonlinear features of these categories and subsequently use the second layer of classifier SVM to classify them deeply.

### **2.2 Analysis of question two**

The topic requires the evaluation of the generalization ability of the model, which is fundamentally for the model to have good prediction ability for new data. We evaluate the generalization performance of the model using the performance of the indicators of the test set, and firstly, we use the evaluation method of k-fold cross-validation to classify and predict the original data as the test set, and quantify the performance of the model in different aspects by evaluating the indicators, so as to derive the deviation between the results using model 1 and the actual, and evaluate the generalization ability of the model.

### **2.3 Analysis of question three**

For the possible overfitting of the proposed model, we determine whether the model has overfitting problems by plotting the model learning curve.

### III. Models

#### 3.1 Basic Model

##### 3.1.1 Terms, Definitions and Symbols

For the convenience of the algorithm that follows, the 19 human behaviors Sitting-  
StandingLying on backLying on right sideAscending stairsDescending stairs Standing  
in an elevator stillMoving around in an elevatorWalking in a parking lotWalking on a  
treadmill with a speed of 4 km/h in flat position and 15 deginclined positionsWalking on  
a treadmill with a speed of 4 km/h in 15 deg inclined positionsRunning on a treadmill  
with a speed of 8 km/h Exercising on a stepperExercising on a cross trainerCycling on  
an exercise bike in horizontal positionCycling on an exercise bike in vertical position  
Rowing Jumping Playing basketball are abbreviated in order as A1, A2, A3, ... etc.

##### 3.1.2 Assumptions

- 1) All participants in the experiment were correct in their judgments about the behavior.
- 2) Accurate and reliable sample data with no gross errors.
- 3) The causes that influenced behavior were all caused by the indicators in the data set, i.e. That is, the influence of other factors on human behavior was not considered.

#### 3.2 Feature Extraction

##### 3.2.1 Random forest based feature importance ranking algorithm

Random forest is an integrated learning algorithm that uses multiple decision trees for training and prediction. In order to measure the correlation between each data feature and Alzheimer's diagnosis, the random forest-based feature importance index (PIM) is calculated for each of the 48 data features according to the formula , and the importance of the data features is ranked according to the PIM value ,as follows:

- ①Construct M decision trees;
- ②When the current decision tree  $k_{tree} = 1$ , the corresponding out-of-bag data  $OOB_k$  is obtained;
- ③Compute the prediction error  $errOOB_k$  of the current decision tree for  $OOB_k$ ;
- ④Set the random perturbation of the  $i$ th data feature in  $OOB_k$  to  $OOB_k^i$ ;



- ⑤For each decision tree,  $k_{tree} = 2, \dots, M$ , repeat the steps ②to ④;  
 ⑥Calculate the importance of data features according to equation eqn1

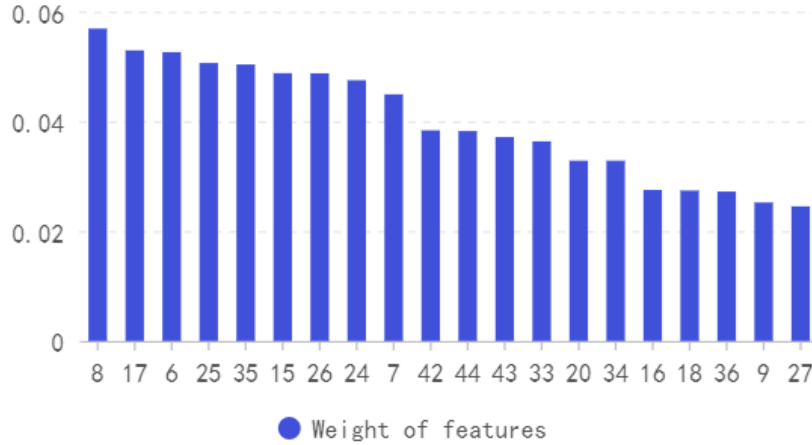
$$PIM = \sum_i^M (errOOB_k^i - errOOB_k) \quad (1)$$

where:  $M$  is the number of constructed decision trees,  $errOOB_k^i$  and  $errOOB_k$  denote the prediction error in the case of  $k_{tree}$  decision trees for the out-of-bag data after adding perturbations to the  $i$ th statistical covariate and the out-of-bag data without adding perturbations, respectively.

### 3.2.2 Extraction of important features

By using this algorithm to analyze 45 features in the dataset, the different importance of different features for 19 classification results can be obtained. We select features with importance greater than 0.02 as important features to participate in the subsequent multi-layer classifier based on generalized discriminant analysis. For the features with weaker importance, we screened them out to avoid the interference information that these features might carry to our classification accuracy.

The histograms plotted for the weights of the significant characteristics are as follows

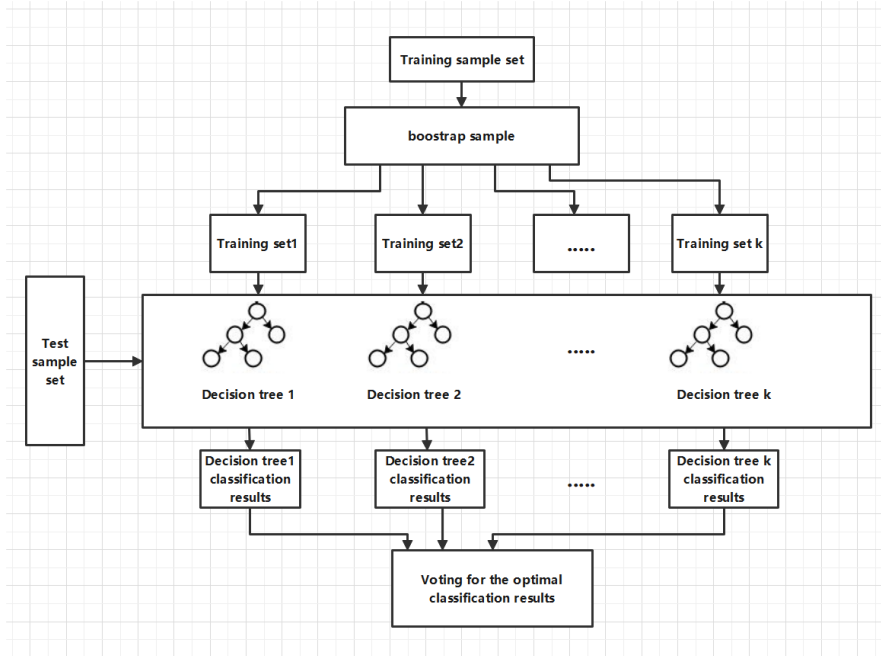


**Figure 2 Weights of important features chart**

### 3.3 Problem one modeling and solving

#### 3.3.1 Random Forest-based First-Level Classifier for Human Behavior

The main features that can classify human behavior have been extracted in the above steps. Considering the relationship between these data and the fact that the samples used for training are discrete and the amount of data is huge, the random forest algorithm is considered for network training, and its general algorithmic flow is shown in Figure fig:1.2. In order to initially identify multi-class activities, the random forest classification algorithm with excellent performance in supervised learning is chosen for the layer 1 classifier. Random forest (RF, Random forest) is a further extension and optimization of Bagging, which introduces a random attribute selection method based on decision tree based learners to build Bagging integration. A random forest is a series of unpruned decision trees (here, categorical regression trees)  $\{h(x, \Theta_k)\}$ . A classifier combined with  $\Theta_k$  is an independent identically distributed random vector, and each tree casts a vote for the most popular class to which the input vector  $X$  belongs. It improves the shortcomings of a single decision tree and does not cause overfitting easily. The specific process and flow chart are as follows



**Figure 3 Random forest algorithm flow chart**

Step1: From the feature-selected human sensor data  $D$ ,  $k$  sub-training samples  $(D_1, D_2, \dots, D_k)$  are selected by Bootstrap sampling, and build  $k$  decision trees.

Step2: At each node of the classification tree,  $m$  metrics are randomly selected from

$n$  metrics, and the optimal features are selected from the  $m$  candidate metrics to grow the nodes according to the principle of minimum node impurity, and the decision tree is allowed to grow until the impurity (i.e., Gini index) of each leaf node is minimized, while no pruning is performed on the decision tree.

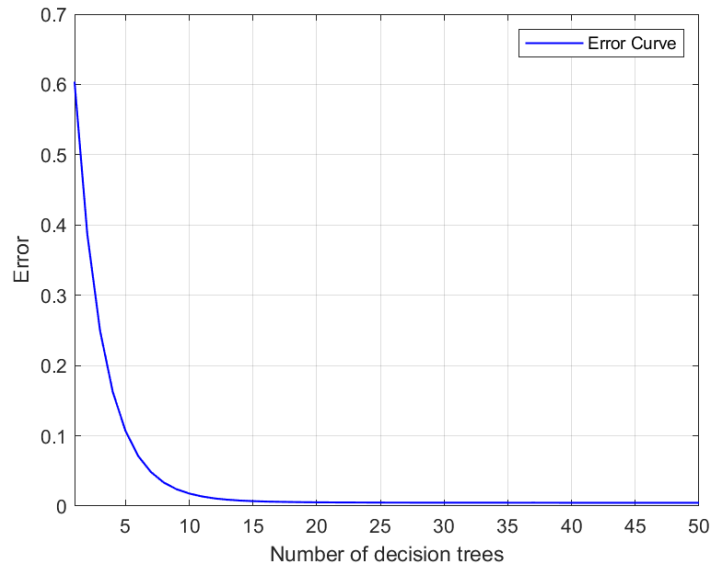
Step3: Repeat Step2 to traverse the pre-built  $k$  decision trees and form a random forest from  $k$  decision trees.

Step4: New unknown samples are predicted based on the grown  $k$  decision trees. The classification result of the sample to be tested is determined by the majority vote of the  $k$ -tree voting. The classification formula is :

$$f(x_i) = \text{majority vote } \{h_i(x)\} \quad (i = 1, 2, \dots, k) \quad (2)$$

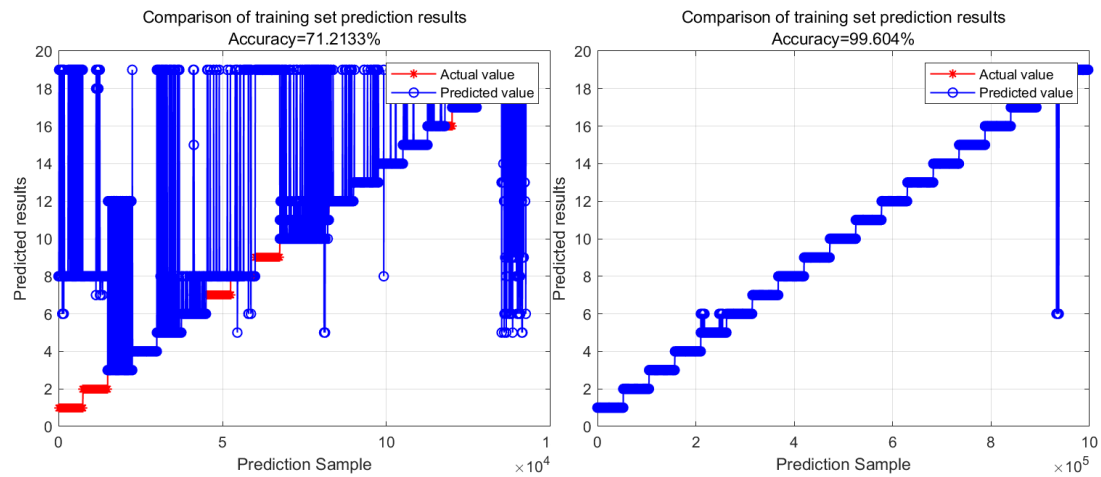
where: majority vote is the result of majority voting.

First, using the random forest model in matlab, the relationship between the number of decisions and the error is plotted using the number of decisions as the independent variable and the error as the dependent variable, as shown in the following figure



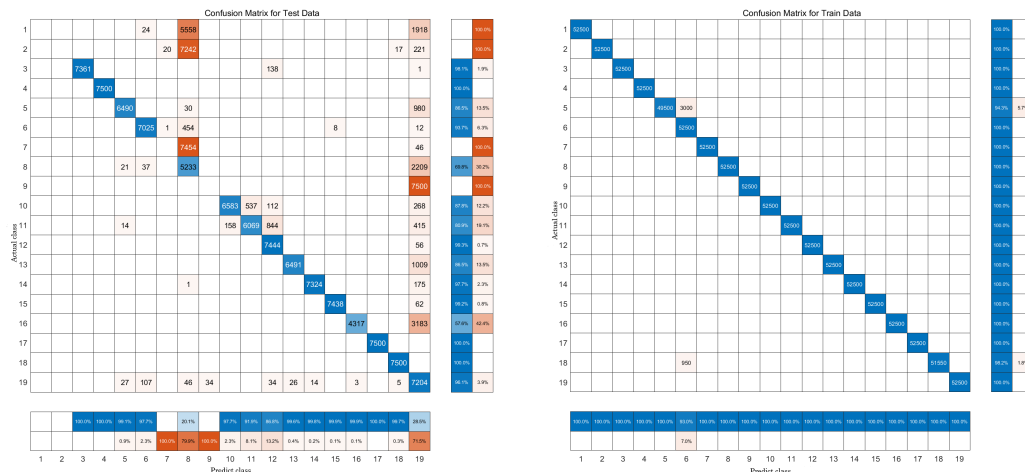
**Figure 4 Plot of the number of decisions versus error in the random forest model**

From the above figure we can see that in this data set, based on the computer performance and model accuracy, as well as the reliability analysis of the model, the number of decision trees was determined to be 50. In order to better analyze the classification effect of the random forest model, we set up a test set to test our training model, using MATLAB with numbers p1-p7 as the training set and p8 as the test set for random forest training and testing, and obtained the results shown in the following figure:



**Figure 5 Random forest model recognition results comparison chart**

To better analyze the above random forest identification results, a confusion matrix plot of the above two results was made using MATLAB as follows.



**Figure 6 Confusion matrix of test set and training set random forest training results**

Observing the above four figures, this paper believes that the main reason for the inconsistency between the action recognition results and the real results is that these actions all have similar characteristics and are extremely easy to be confused in the algorithm recognition process, such as going up and down stairs, standing and sitting down, except for the actions in this case mentioned above, the prediction rate of all other actions is close to 100%, which means that these actions can be recognized and classified in this layer of the classification model into the real. The remaining unidentified actions are divided into two main blocks, like A1, A2, and A7 are classified as A8 by the model, and A9 and A16 are similarly classified as A19 by the model. for the convenience of the next fine classification model, these confusing actions are divided into two main

categories as shown.

**Table 1 Confusion action classification table**

	Easily confused actions
ConfusionIcategory	A1,A2,A7,A8
ConfusionIIcategory	A9,A16,A18

### 3.3.2 Feature mapping based on generalized discriminant analysis

In order to solve the problem of similarity of the above 2 types of confusing actions, the nonlinear features are further extracted by using generalized discriminant analysis, which is a nonlinear extension of linear discriminant analysis, and the Fisher discriminant analysis is performed in the high-dimensional feature space by mapping the input features to a higher-dimensional feature space through a nonlinear mapping.

Let  $\emptyset$  be the nonlinear mapping from the input feature space to the high-dimensional feature space  $F$  :

$$\emptyset : R^d \rightarrow F, x \rightarrow x^\emptyset \quad (3)$$

Define the inter-class scatter matrix  $S_B\emptyset$  and the total intra-class scatter matrix  $S_W\emptyset$ :

$$\begin{aligned} S_W^\emptyset &= \frac{1}{N} \sum_{i=1}^c \sum_{j=1}^{N_i} \left[ \emptyset \left( x_i^j - m_i \right) \right] \left[ \emptyset \left( x_i^j - m_i \right) \right]^T \\ S_B' &= \frac{1}{N} \sum_{i=1}^c N_i (m_i - m) (m_i - m)^r \end{aligned} \quad (4)$$

where  $N$  represents a set of high-dimensional feature vector dimensions,  $C$  represents the number of categories,  $m_i$  represents the mean of the  $i$ th group of high-dimensional features,  $m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \Phi(x_j)$ ;  $m$  represents the mean of all categories of high-dimensional features,  $m = \frac{1}{N_i}$

Define the discriminant criterion  $J(w)$  :

$$J(w) = \max \left( \frac{(w^\emptyset)^T S_B^\emptyset w^\emptyset}{(w^\emptyset)^T S_W^\emptyset w^\emptyset} \right) \quad (5)$$

Solving the projection vector  $w_{opt}$  is the solution to the eigenvalue problem.

$$\lambda S_W^\theta w^\theta = S_B^\theta w^\theta \quad (6)$$

Due to the high dimensionality of the feature space  $F$  cannot be solved directly, the inner product kernel function RBF is invoked as the mapping function  $k$ :

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (7)$$

where  $x, y$  represent the corresponding eigenvalues and  $\sigma$  is a constant, representing the degree of nonlinearization.

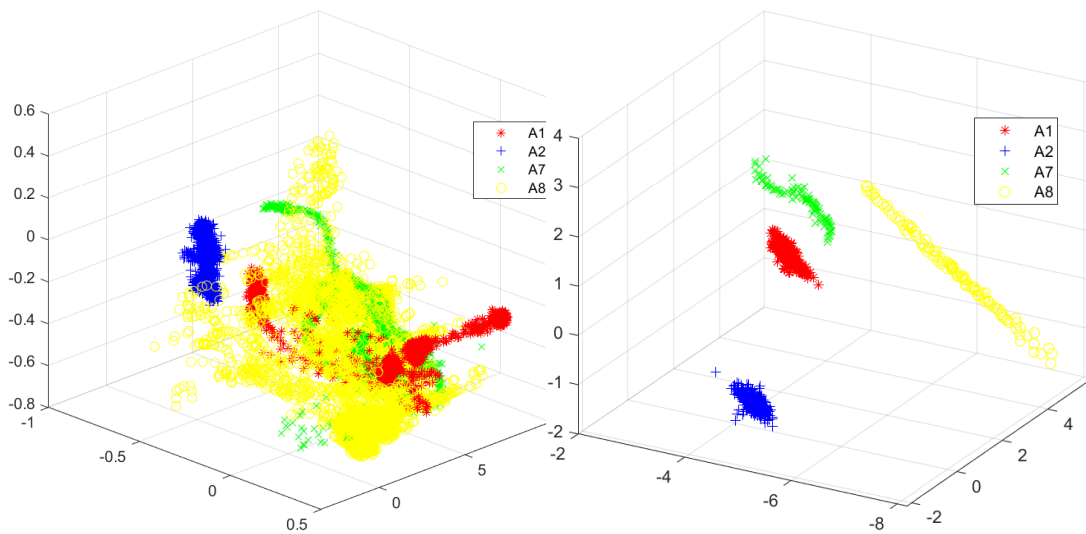
From the regenerative kernel theory, it follows that any solution in a high-dimensional space can be represented as a linear combination of training samples in that space:

$$w^\theta = \sum_{i=1}^c \sum_{j=1}^N a_{ij} \phi(x_{ij}) \quad (8)$$

Then the projection of the sample in the best projection direction is :

$$w^\theta \phi(x) = \sum_{i=1}^c \sum_{j=1}^N a_{ij} k(x_{ij}, x) \quad (9)$$

Taking the confusion I class as an example, first extract three indicators with large correlations from the dataset: RA\_ygyro, LA\_yacc, RA\_xmag as doing X-axis, Y-axis and Z-axis, and use MATLAB to get the scatter plot as shown in Figure creffig:111 shows, and then use SAS to perform generalized discriminant analysis on these three indicators to get as shown in Figure .



**Figure 7 Scatterplot before and after generalized discriminant analysis features**

### 3.3.3 SVM-based second layer classifier for human behavior

In order to further subdivide the confusion action into a specific action, this paper introduces SVM vector machine as a sub-classification model to divide the confusion action. The principle of SVM classifier is to take the hyperplane to maximize the feature distance between different categories so as to achieve the classification effect. As shown in the figure, the wider the width of the classification interval (i.e., maximizing), the lower the impact caused by the local interference in the training set. Therefore, it can be considered that the last classification method has the best generalization performance and generality. The model of SVM can be formulated as:

$$y = \text{sign}(w^T x + b) \quad (10)$$

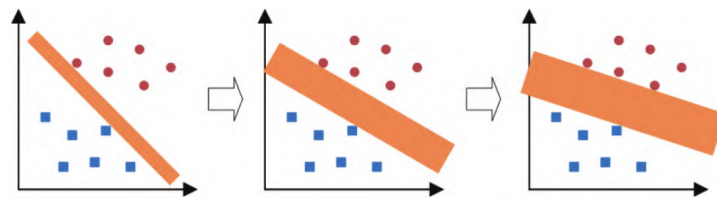
where,  $x$  is the feature vector,  $w$  is the weight vector,  $y$  is the marker vector, and  $\text{sign}(y)$  is the sign function.

When  $y = 1$ , the sample is positive; when  $y = -1$ , the sample is negative, i.e.

$$\begin{cases} w^T x + b > 0, y = 1 \\ w^T x + b \leq 0, y = -1 \end{cases} \quad (11)$$

As shown in Figure (2), SVM usually finds the optimal classification hyperplane by maximizing the classification interval. Assuming that the input of the training set is the set of  $x(i)$  vectors and the output is the set of  $y(i)$  vectors, the classification interval is twice the minimum distance from the full set of samples to the hyperplane, i.e., where  $m$  is the number of samples.

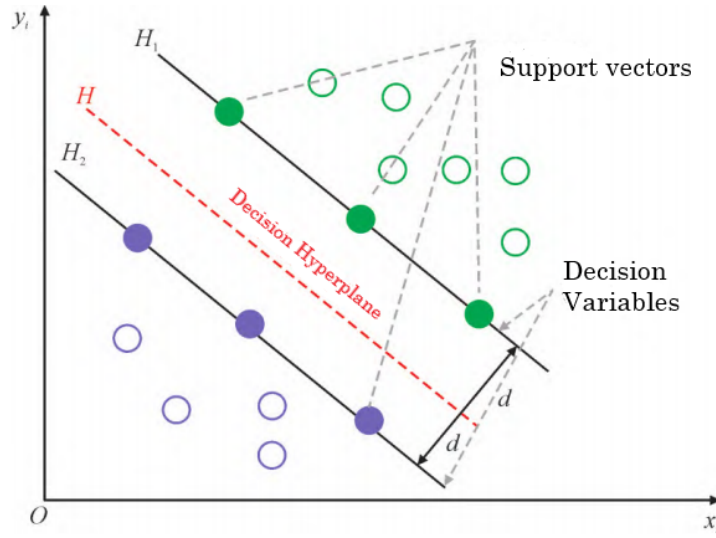
$$\gamma = \min_{i=1, \dots, m} 2y^{(i)} \left( \frac{w^T x^{(i)} + b}{\|w\|} \right) \quad (12)$$



**Figure 8 Vector machine classification flow chart**

Mathematically, all sample points that meet the requirements of equation (4) (i.e., sample points with the smallest Euclidean distance to the classification hyperplane) will be defined as support vectors, then the set of samples must satisfy the following two

cases: if the samples are positive, then  $w^T x^{(i)} + b = 1$  If the samples are negative, then  $w^T x^{(i)} + b = -1$ , as shown in Fig.



**Figure 9 Vector machine classification schematic**

Therefore, the characteristic samples in the sample set should satisfy when the discriminant equation is multiplied by the corresponding coefficients.

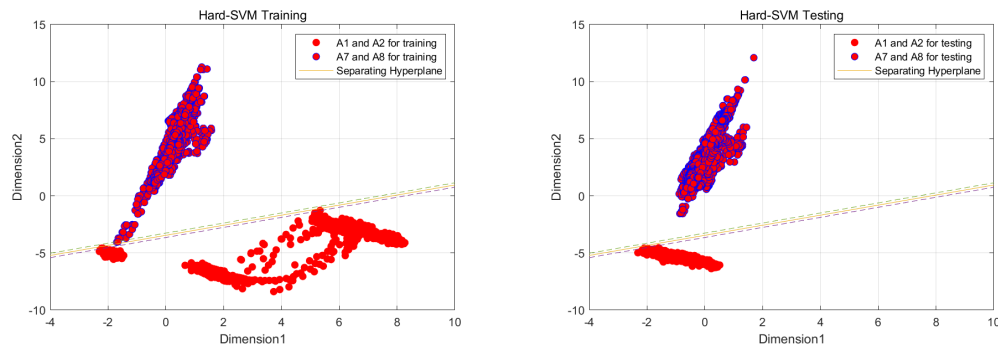
$$y^{(i)} (w^T x^{(i)} + b) \geq 1 \quad (13)$$

In this paper, we use MATLAB to sub-classify the above model, and input the indicators that have been generalized discriminant analysis into SVM as the original data, taking the confusion I class as an example, because A1 and A2 are more closely connected, and A7 and A8 are also more closely connected, so we first subdivide the confusion I class into two large classes A1, A2, and A7, A8, and then a second subdivision, we can subdivide the confusion I class into the more A1, A2, A7, and A8 classes by a two-layer SVM vector machine. A2, A7, and A8 which are the four classes of activities.

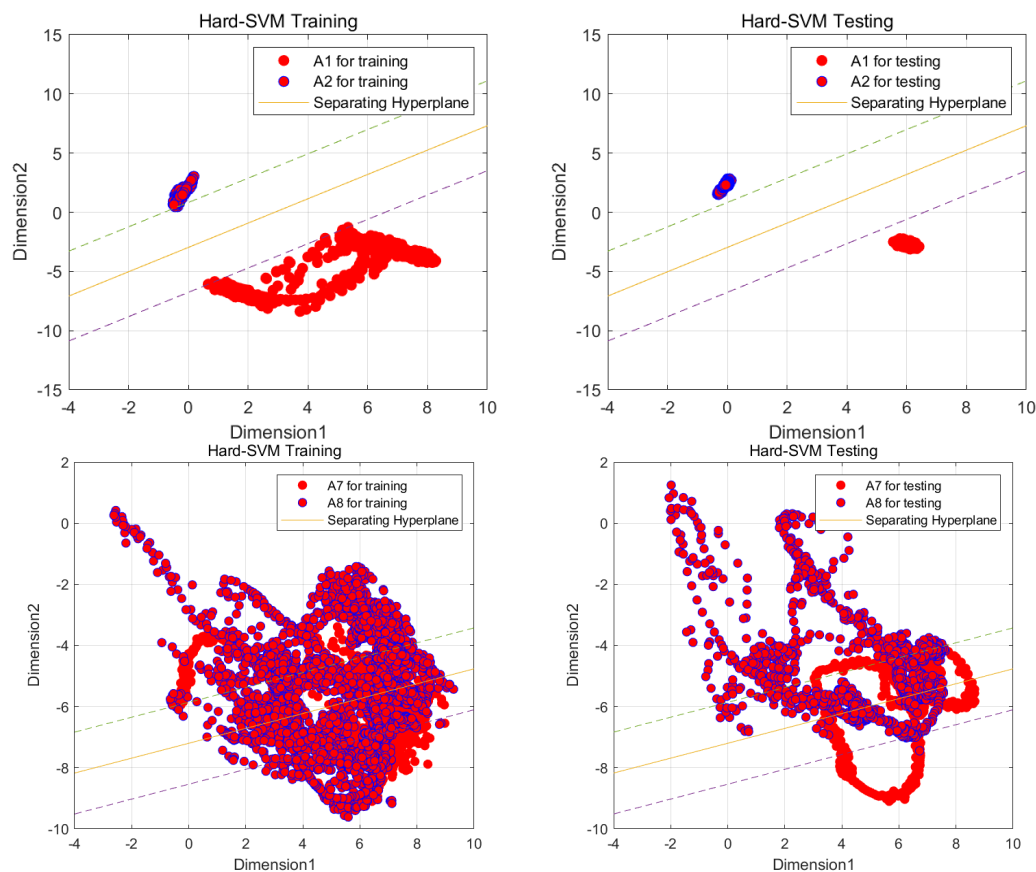
Through the above steps, the data of the confusion I class has been classified into two major classes A1, A2 and A7, A8 by SVM vector machine, and in order to classify them more carefully, this paper then performs a fine classification of these two major classes into specific activity classes. As shown in the figure.

Through the steps related to fig:7451, we are able to classify all the data of the confusion I class into specific active classes by the above SVM vector machine meticulous classification, although the effect of SVM vector machine fine classification A7, A8 is





**Figure 10 SVM preliminary segmentation obfuscation I class result graph**

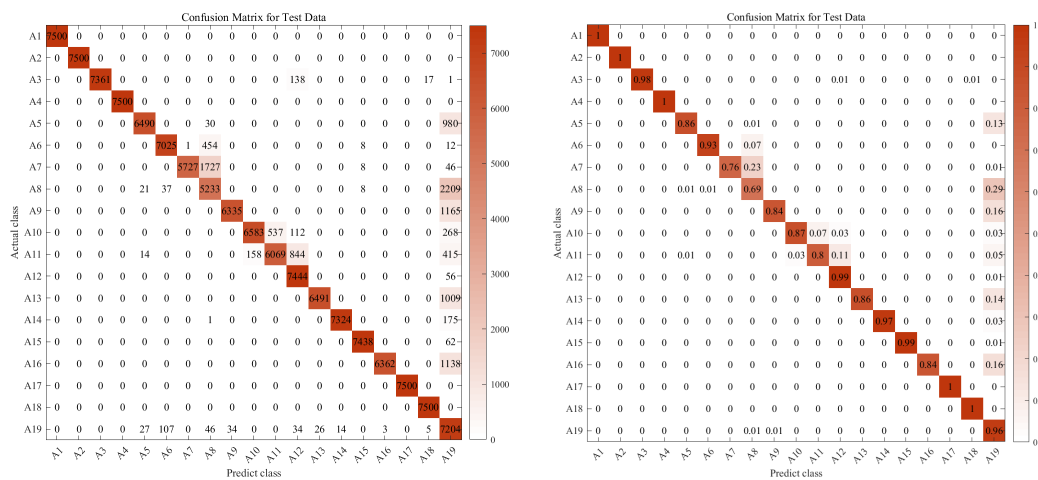


**Figure 11 SVM fine classification obfuscation I class result graph**

not significant, but it is much better than the initial random forest classification effect has been much better than the initial random forest.

In this paper, the same operation was also performed on the confusion II class, which was input to the second layer of the classification vector machine to obtain the recognition probabilities of the four similar actions, and the final recognition results of the four similar activities were obtained by weighted average with the recognition probabilities of the first layer classifier. The confusion matrix is shown in fig:666, and it can be seen that the original confusion-prone actions are improved a lot, and the overall

correct rate is improved from 71% to 91%.



**Figure 12 Confusion matrix diagram for two-level classifier**

In this paper, other classification machine learning models are also trained as shown in the table and compared with the two-layer classifier model, which has been shown to significantly improve the accuracy of classification.

**Table 2 Classification accuracy of each machine learning model**

Classification algorithm	Logistic regression	Lightgbm	KNN	double-layer classifier
Accuracy	52%	62%	65%	91%

### 3.4 Problem two modeling and solving

The topic requires the evaluation of the generalization ability of the model, which is fundamentally for the model to have good prediction ability for new data. We evaluate the generalization performance of the model using the performance of the indicators of the test set, and firstly, we use the evaluation method of k-fold cross-validation to classify and predict the original data as the test set, and quantify the performance of the model in different aspects by evaluating the indicators, so as to derive the deviation between the results using model 1 and the actual, and evaluate the generalization ability of the model.

#### 3.4.1 Generalization capability

The topic requires the evaluation of the generalization ability of the model, which is fundamentally for the model to have good prediction ability for new data. We evaluate

the generalization performance of the model using the performance of the indicators of the test set, and firstly, we use the evaluation method of k-fold cross-validation to classify and predict the original data as the test set, and quantify the performance of the model in different aspects by evaluating the indicators, so as to derive the deviation between the results using model 1 and the actual, and evaluate the generalization ability of the model.

Refers to the predictive ability of the model for unknown data. The generalization ability is analyzed theoretically. If the learned model is  $\hat{f}$  then the error measured with this model for unknown data is the generalization error

$$R_{\text{exp}} = E_P[L(Y, \hat{f}(X))] = \int_{x,y} L(y, \hat{f}(x))P(x, y)dxdy \quad (14)$$

The generalization error is also the expected risk of the learned model.

### 3.4.2 k-fold cross-validation

Cross-validation is a statistical analysis method used to verify the performance of classifiers. The basic idea is to group the original data into K groups, with one subset as the training set and the other as the validation set, and then use the training set to train the classifier, and then use the validation set to test the trained model as the performance indicator of the classifier. data are combined together as the training set. As shown in Fig.



**Figure 13 k-fold cross-validation principle**

K models are obtained by group training, and the average accuracy of the validation set of these K models is used as the performance indicator of the K-fold cross-validation

classifier. k-fold cross-validation can avoid the occurrence of over- and under-learning states, and the final results are more convincing.

### 3.4.3 Selection of performance metrics

For the established model which is a multi-classification model by nature, we choose the confusion matrix to visualize the classification accuracy of the model, and the higher classification indexes Accuracy and F1-Score obtained in the confusion matrix are used as the criteria to judge the overall classification model. And by drawing ROC curves, we compensate the problem of class imbalance that often occurs in the actual data set and get the probability value of correct classification of the model.

### 3.4.4 Confusion Matrix

In the field of machine learning, the Confusion Matrix, also known as the likelihood matrix or the error matrix. Confusion matrices are visualization tools, especially for supervised learning, and in unsupervised learning are generally called matching matrices. In image accuracy evaluation, it is mainly used to compare classification results with actual measured values, and the accuracy of classification results can be displayed inside a confusion matrix.

The structure of the confusion matrix is generally represented in the following way.

<div> <div>Predict</div> <div>Real</div> </div>	0	1
	0	1
0	TN	FP
1	FN	TP

**Figure 14 Confusion matrix structure**

The meaning to be expressed by the confusion matrix

(a) Each column of the confusion matrix represents the predicted category and the total of each column indicates the number of data predicted to be in that category

Each row represents the true category to which the data belongs, and the total number of data in each row indicates the number of data instances in that category; the value in each column indicates the number of true data predicted to be in that category.

True Positive (TP): true class. The true class of the sample is positive and the result of the model identification is also positive.

False Negative (FN): False negative class. The true class of the sample is positive, but the model identifies it as a negative class.

False Positive (FP): False positive class. The true class of the sample is negative, but the model identifies it as positive.

True Negative (TN): True negative class. The true class of the sample is negative, and the model identifies it as such.

The confusion matrix is a summary of the predictions for a classification problem. The number of correct and incorrect predictions is summarized using count values and broken down by each class, which is the key to the confusion matrix. The confusion matrix shows which parts of the classification model are confused when making predictions. It provides insight not only into the errors made by the classification model, but more importantly, into what types of errors are occurring. It is this decomposition of the results that overcomes the limitations associated with using classification accuracy alone.

After cross-validation, the output confusion matrix is shown in the figure.

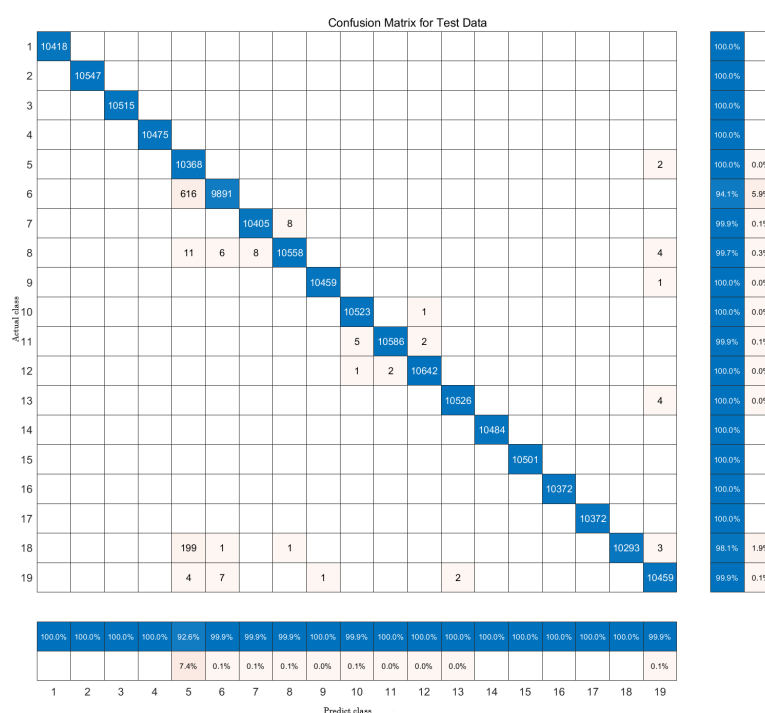


Figure 15 Confusion Matrix Diagram

From the confusion matrix, we can see that the accuracy of the model is 99% after cross-validation, which proves that the model has a good ability of classification prediction.

Meanwhile, we calculate the following evaluation indexes by the confusion matrix to better evaluate the generalization ability of the model.

### 3.4.5 Accuracy and F1-score

accuracy:

Accuracy rate is the most commonly used classification performance metric. It can be used to express the precision of a model, i.e., the number of correct identifications by the model/total number of samples. In general, the higher the precision of the model, the better the model is.

$$Accuracy = (TP + TN) / (TP + FN + FP + TN) \quad (15)$$

F1-score:

There is often a reciprocal relationship between recall and precision; when the model can find more positive samples, it tends to also result in classifying more negative samples as positive samples, i.e., when recall is high, precision tends to be lower, and when precision is high, recall tends to be lower. To strike a balance between these two metrics, we choose the F1 metric, which is the summed average of the above two. It is the summed average of precision and recall, with a maximum of 1 and a minimum of 0.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (16)$$

The accuracy and F1-score of the model after cross-validation are shown in the figure

Evaluation indicators	accuracy	F1-score
double-layer classifier	99.3%	97.2%

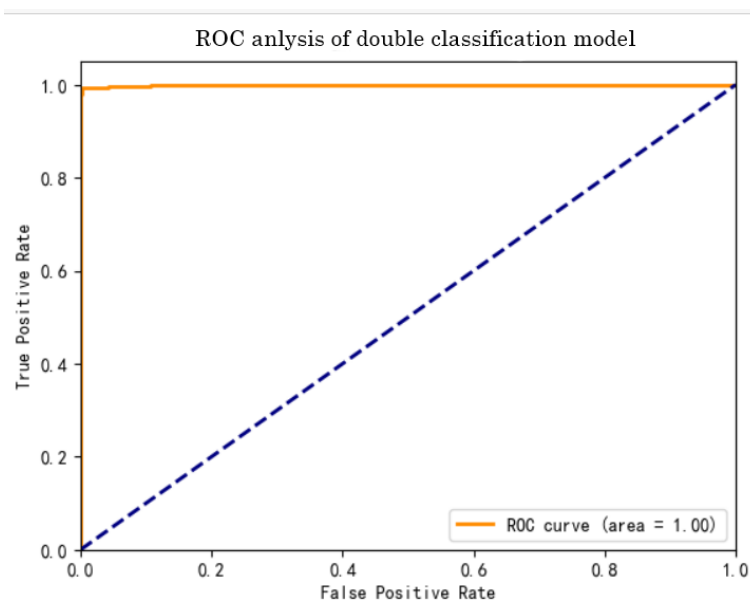
From the table, we can see that the model accuracy and F1-score are over 97%, indicating that the model performs well on the validation set and proves that the two-layer classification model has good generalization ability.

### 3.4.6 ROC curve

The ROC curve is a curve based on a series of different dichotomies (cut-off values or decision thresholds), with the true positive rate (sensitivity) as the vertical coordinate and the false positive rate (1-specificity) as the horizontal coordinate. Unlike traditional diagnostic test evaluation methods, which have a common feature that test results must be divided into two categories and then statistically analyzed, the ROC curve evaluation method does not require this restriction, but allows for an intermediate state according to the actual situation, allowing for the division of test results into multiple ordered categories. Therefore, the ROC curve evaluation method is applicable to a wider range.

The ROC curve combines sensitivity and specificity in a graphical way to accurately reflect the relationship between the specificity and sensitivity of an analytical method, and is a comprehensive representation of test accuracy. The more convex the ROC curve is near the upper left corner, the greater the diagnostic value, which facilitates the comparison between different indicators. The area under the curve can be used to evaluate the diagnostic accuracy.

The model ROC curve is shown in the figure



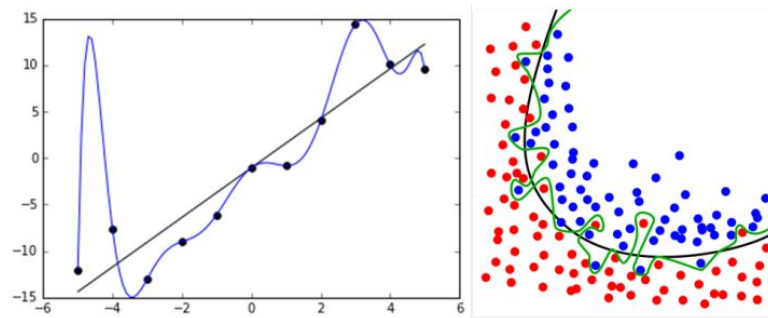
**Figure 16 ROC Curve Chart**

From the trend of roc curve in the figure, we can judge that the model output accuracy probability is close to 99%. It has good generalization ability.

### 3.5 Modeling and solving Problem three

#### 3.5.1 *Overfitting studies*

Overfitting is a problem in the process of fitting the model parameters. Since the training data contains sampling errors, the complex model is trained to take the sampling errors into account and fit the sampling errors well as well.



**Figure 17 Overfitting schematic**

This is demonstrated by the fact that the final model works well on the training set; it works poorly on the test set. The model generalization ability is weak.

Why do we need to solve the overfitting phenomenon? This is because the models we fit are generally used to predict unknown outcomes (not in the training set), and overfitting, while effective on the training set, is poor when used in practice (the test set). At the same time, in many problems, we cannot exhaust all states and it is impossible to include all cases in the training set. Therefore, the overfitting problem must be solved.

Why is it more common in machine learning? This is because machine learning algorithms generally have the ability to fit models much higher than the problem complexity in order to satisfy the most complex tasks possible, i.e., machine learning algorithms have the ability to fit the correct rules with the ability to further fit the noise.

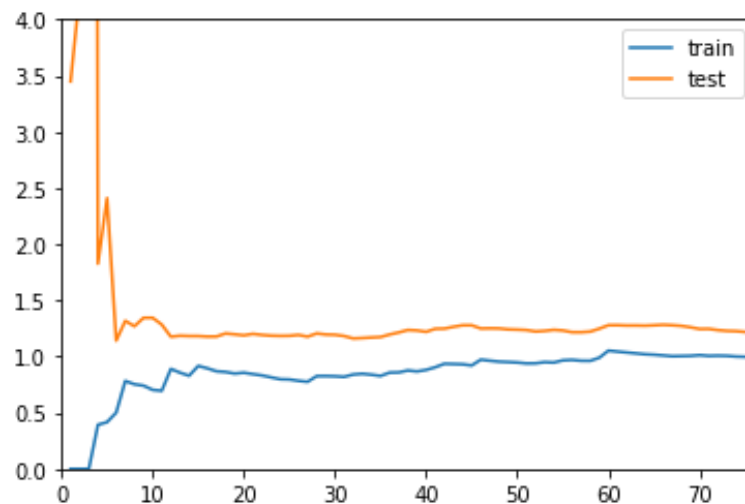
#### 3.5.2 *Learning curve based model fitting problem determination*

The learning curve is a way to see how the model performs on new data by plotting the accuracy of the training set and cross-validation at different training set sizes, and thus to determine whether the model has high variance or bias, and whether increasing the training set can reduce overfitting.

The model learning curve is shown in the figure:

As can be seen from the learning curve, the error curve of the test data set is closer to that of the training data set and tends to be stable, indicating that the model has





**Figure 18 Learning Curve Chart**

a strong generalization ability and has less prediction error for for new data without overfitting. It shows that our classification algorithm can be widely applied to people's behavior classification problems.

## **IV. Strengths and Weakness**

### **4.1 Advantages of the model**

1. Problem 1 uses a two-layer classifier fusion model, which has better classification performance and substantially higher accuracy compared to a single machine learning model.
2. In the process of evaluating the generalization ability of the model, Question 2 uses k-fold cross-validation for the characteristics of the multiclassification model and selects multiple metrics to reflect the model generalization performance comprehensively and objectively.
3. Problem 3 uses a learning curve for visual presentation and presents its own analysis when examining the overfitting problem.

### **4.2 Disadvantages of the model**

1. Data processing is relatively simple, and some important data may be lost.

## V. References

- [1] 邱阳,李盛,金亮,张咪咪,王杰. 基于统计特征混合与随机森林重要性排序的桥梁异常监测数据识别方法[J]. 传感技术学报,2022,35(06):756-762.
- [2] 冯昊,李树青. 基于多种支持向量机的多层级联式分类器研究及其在信用评分中的应用[J]. 数据分析与知识发现,2021,5(10):28-36.
- [3] 李辉,李瑞祥,张耀威,乐燕芬,施伟斌. 多层分类器模型的相似人体活动识别[J]. 小型微型计算机系统,2021,42(04):861-867.
- [4] 路佳佳. 基于交叉验证的集成学习误差分析[J]. 计算机系统应用: 1-8[2022-12-05]10.15888.
- [5] 许志兴,吴俊华,唐晓纹. 基于粗集的多层分类器的设计与实现[J]. 计算机工程与应用,2006(08):184-186.
- [6] 张洋,姚登峰. 人类的行为识别分类方法综述[C]. 中国计算机用户协会网络应用分会2019年第二十三届网络新技术与应用年会论文集.,2019:44-47.DOI:10.26914.2019.004425.
- [7] 周智强. 面向股票价格预测的深度学习过拟合问题研究及其优化[D]. 深圳大学,2020.DOI:10.2732.12020.000809.
- [8] 吕志浩. 多任务学习组合预训练模型泛化能力的研究[D]. 华东师范大学,2022.DOI:10.27149.2022.001283.

## VI. Appendix

Listing 1: The matlab Source code of Random Forest Classification Algorithm

```
warning off
close all
clear
clc

testdata = csvread('testdata.csv');
traindata = csvread('traindata.csv');
P_train = traindata(:,1:20)';
T_train = traindata(:,21)';
M = size(P_train, 2);
P_test = testdata(:,1:20)';
T_test = testdata(:,21)';
N = size(P_test, 2);
[p_train, ps_input] = mapminmax(P_train, 0, 1);
p_test = mapminmax('apply', P_test, ps_input );
t_train = T_train;
t_test = T_test ;
p_train = p_train'; p_test = p_test';
t_train = t_train'; t_test = t_test';
trees = 50;
leaf = 1;
OOBPrediction = 'on';
OOBPredictorImportance = 'on';
Method = 'classification';
net = TreeBagger(trees, p_train, t_train, 'OOBPredictorImportance',
    OOBPredictorImportance, ...
    'Method', Method, 'OOBPrediction', OOBPrediction, 'minleaf', leaf);
importance = net.OOBPermutedPredictorDeltaError;
t_sim1 = predict(net, p_train);
t_sim2 = predict(net, p_test );
writecell(t_sim1, 't_sim1.csv');
writecell(t_sim2, 't_sim2.csv');
T_sim1 = csvread('t_sim1.csv');
T_sim2 = csvread('t_sim2.csv');
```

```
error1 = sum((T_sim1' == T_train)) / M * 100 ;
error2 = sum((T_sim2' == T_test )) / N * 100 ;

figure
plot(1 : trees, oobError(net), 'b-', 'LineWidth', 1)
legend('Error Curve')
xlabel('Number of decision trees')
ylabel('Error')
xlim([1, trees])
grid
figure
bar(importance)
legend('Importance')
xlabel('Features')
ylabel('Importance')

[T_train, index_1] = sort(T_train);
[T_test , index_2] = sort(T_test );
T_sim1 = T_sim1(index_1);
T_sim2 = T_sim2(index_2);

figure
plot(1: M, T_train, 'r-*', 1: M, T_sim1, 'b-o', 'LineWidth', 1)
legend('Actual value', 'Predicted value')
xlabel('Prediction Sample')
ylabel('Predicted results')
string = {'Comparison of training set prediction results'; ['Accuracy='
    num2str(error1) '%']};
title(string)
grid
figure
plot(1: N, T_test, 'r-*', 1: N, T_sim2, 'b-o', 'LineWidth', 1)
legend('Actual value', 'Predicted value')
xlabel('Prediction Sample')
ylabel('Predicted results')
string = {'Comparison of training set prediction results'; ['Accuracy='
    num2str(error2) '%']};
title(string)
grid
```

```

fig = figure;
cm = confusionchart(T_train,
    T_sim1,'RowSummary','row-normalized','ColumnSummary','column-normalized');
cm.Title = 'Confusion Matrix for Train Data';
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';
fig_Position = fig.Position;
fig_Position(3) = fig_Position(3)*1.5;
fig.Position = fig_Position;
fig = figure;
cm = confusionchart(T_test,
    T_sim2,'RowSummary','row-normalized','ColumnSummary','column-normalized');
cm.Title = 'Confusion Matrix for Test Data';
cm.ColumnSummary = 'column-normalized';
cm.RowSummary = 'row-normalized';
fig_Position = fig.Position;
fig_Position(3) = fig_Position(3)*1.5;
fig.Position = fig_Position;

```

Listing 2: The matlab source code of SVM classification algorithm

```

clear, clc, close all;
group_1=normrnd(1,0.3,100,2);
label_1=ones(size(group_1,1),1);
group_2=normrnd(3,0.3,100,2);
label_2=-ones(size(group_2,1),1);
X=[group_1;group_2];
y=[label_1;label_2];
tg1=normrnd(1,1,1000,2);
tlabel_1=ones(size(tg1,1),1);
tg2=normrnd(3,1,1000,2);
tlabel_2=-ones(size(tg2,1),1);
test_group=[tg1;tg2];
test_label=[tlabel_1;tlabel_2];
%% Training a SVM(Support Vector Machine) Classifier
C = 10;
svm = mysvmtrain(X, y, C, 'linear');
result = mysvmtest(svm, test_group, 'linear');

```

```

fprintf('Accuracy: %f\n', mean(double(result.y_pred == test_label)));
tmp_b= (svm.alphas' .* svm.vec_y' * kernels(svm.vec_x', svm.vec_x',
    'linear'))';
total_bias = svm.vec_y - tmp_b;
true_b = mean(total_bias);
w=svm.vec_x'*(svm.alphas' .* svm.vec_y)';
subplot(1,2,1);
plot(group_1(:,1),group_1(:,2),'ro','MarkerFace','r');
hold on
plot(group_2(:,1),group_2(:,2),'bo','MarkerFace','r');
hold on
k=-w(1)./w(2);
bb=-true_b./w(2);
xx=-1:5;
yy=k.*xx+bb;
plot(xx,yy,'-')
hold on
yy=k.*xx+bb+1./w(2);
plot(xx,yy,'--')
hold on
yy=k.*xx+bb-1./w(2);
plot(xx,yy,'--')
title('Hard-SVM Training')
xlabel('Dimension1')
ylabel('Dimension2')
legend('Group1 for training','Group2 for training','Separating
    Hyperplane')
grid on;
subplot(1,2,2);
plot(tg1(:,1),tg1(:,2),'ro','MarkerFace','r');
hold on
plot(tg2(:,1),tg2(:,2),'bo','MarkerFace','r');
hold on
k=-w(1)./w(2);
bb=-true_b./w(2);
xx=-1:5;
yy=k.*xx+bb;

```

```

plot(xx,yy,'-')
hold on
yy=k.*xx+bb+1./w(2);
plot(xx,yy,'--')
hold on
yy=k.*xx+bb-1./w(2);
plot(xx,yy,'--')
title('Hard-SVM Testing')
xlabel('Dimension1')
ylabel('Dimension2')
legend('Group1 for testing','Group2 for testing','Separating
      Hyperplane')
grid on;

```

Listing 3: The matlab source code of k-fold cross-validation

```

function auc = plotroc(y,x,params)
    rand('state',0); % reset random seed
    if nargin < 2
        help plotroc
        return
    elseif isempty(y) | isempty(x)
        error('Input data is empty');
    elseif sum(y == 1) + sum(y == -1) ~= length(y)
        error('ROC is only applicable to binary classes with labels 1,
              -1'); % check the trainig_file is binary
    elseif exist('params') && ~ischar(params)
        model = params;
        [predict_label,mse,deci] = svmpredict(y,x,model) ;% the
            procedure for predicting
        auc = roc_curve(deci*model.Label(1),y);
    else
        if ~exist('params')
            params = [];
        end
        [param,fold] = proc_argv(params); % specify each parameter
        if fold <= 1
            error('The number of folds must be greater than 1');
        end
    end
end

```

```
        else
            [deci,label_y] = get_cv_deci(y,x,param,fold); % get the value
                of decision and label after cross-calidation
            auc = roc_curve(deci,label_y); % plot ROC curve
        end
    end
end
function auc = roc_curve(deci,label_y) %%deci=wx+b, label_y, true label
    [val,ind] = sort(deci,'descend');
    roc_y = label_y(ind);
    stack_x = cumsum(roc_y == -1)/sum(roc_y == -1);
    stack_y = cumsum(roc_y == 1)/sum(roc_y == 1);
    auc = sum((stack_x(2:length(roc_y),1)-
    stack_x(1:length(roc_y)-1,1)).*stack_y(2:length(roc_y),1))
    plot(stack_x,stack_y);
    xlabel('False Positive Rate');
    ylabel('True Positive Rate');
    title(['ROC curve of (AUC = ' num2str(auc) ' )']);
end
```