Problem Chosen	2022 ShuWei Cup Summary Sheet	Team Control Number
C		2022111414052

Summary

Alzheimer's disease (AD) is a progressive neurodegenerative disease among the elderly. The research on the diagnosis of AD is one of key directions in the field of psychiatry. Based on the Brain Structural Features and Cognitive Behavioral Features collected in earlier studies, we conducted further studies on intelligent diagnosis of AD.

For task one, ~~in order~~ to facilitate the follow-up research work, we preprocessed the previous research data. After analyzing the characteristics in dataset, we found that there are missing data, invalid data, and correlated data. First, we removed invalid data such as patient registration information. Then, we used **Cubic Spline Interpolation** to interpolate 91 selected indicators. Next, we do **Spearman correlation** ~~**analyse**~~ to verify the correlation between 91 groups of indicators. As a result ,we found that there is a nonlinear connection between diagnosis and features, which should be taken into account in subsequent studies.

For task two, the diagnosis of AD is a complex nonlinear problem, and it is necessary to improve the recognition efficiency and accuracy. Due to the high dimension of dataset, we establish an **Auto-Encoder(AE)** model which can compress dimension , and use Deep Learning method to explain relationships between input set and 5 types of diagnosis. Meanwhile, so as to improve the training accuracy, we use the **Random Forest(RF)** classificator to conduct simulations for the classification of 5 symptoms types. With verification, RF increases model **accuracy to 98.2695%**, indicating the practicability and reliability of AE-RF model.

For task three,to refine the clustering of SMC,EMCI and LMCI, we analyze the common features among 3 datasets. We extracted the ~~features in~~ Brain Structural Features and Cognitive Behavioral Features for the 3 types respectively. Based on 2 sets of comprehensive indicators' features, we established **K-Mediods clustering** model, and draw the subclass label graphs of 3 types of diseases as shown in Figs. 5-7. In addition, we test the clustering performance of the models in turn and ~~caculate~~ that**CH Index of SMC is 2605.78,of EMCI is 7376.13,of EMCI is 5834.86**,indicating clustering performance is excellent.

For task four,firstly, we extracted the patient samples of regular reexamination as ~~new~~ dataset for time series analysis. By plotting the time series curve, we found that there is a strong correlation between indicators characteristics and time for the same patient. ~~Afterwards,~~ we established **Gaussian process regression** model to reflect the ~~change~~ relationship of the four types of diseases with time through analytical functions, as shown in Figure 13-16.

For task five,based on the research results and existing literature, we analyzed the clinical diagnostic criteria of 5 types of patients. By consulting the relevant literature, we proposed intervention programs for patients with different conditions to provide ~~reference~~ for the early diagnosis and intervention of AD.

Key word: Alzheimer,Auto-Encoder, Random Forest, K-Mediods, Gaussian Process Regression

Content

1. Introduction

1.1 Background

Alzheimer's disease (AD) is a progressive neurodegenerative disease, which usually affects the elderly. The clinical symptoms are mainly dementia, personality alienation and aphasia. Most patients will gradually lose their ability to take care of themselves within 10 to 20 years after the onset . The preclinical stage of Alzheimer's disease, also known as mild cognitive impairment (MCI) stage, is the best intervention stage. How to accurately and timely identify and diagnose the disease in the early stage of AD has become a key direction in the field of psychiatry.

1.2 Restatement of the Problem

A set of medical records of Alzheimer's disease patients is available in the Attachment Data, which provides a total of 16208 records. The initial study divided patients into five categories, as shown below in Table 1:

Table 1 Patient Situaiton category Data

Stage	Patient category	Abbreviation	Sum of Cases
Elderly	Cognitive Normal Elderly	***CN***	4850
MCI	Subjective Memory Complaint	***SMC***	1416
	Early Mild Cognitive Impairment	***EMCI***	2968
	Late Mild Cognitive Impairment	***LMCI***	5236
Post	Alzheimer's disease	***AD***	1738
Total			16208

Based on the previous data, it is necessary for us to develop a mathematical AD identification model and design an intelligent medical diagnostic method to recognize AD, here are five tasks in detail below:

Task 1: The medical records in attachment were expected to be pre-processed,so as to investigate the characteristics of the indicators and the diagnosis of Alzheimer's disease Correlation.

Task 2: With the attachment data of structural brain features and cognitive behavioral features, now it is supposed to establish an intelligent diagnosis of AD.

Task 3: It is required to refined the clustering 3 classes, including SMC, EMCI and LMCI, into 3 new subclasses, subdivided from MCI category in the attachment.

Task 4: The same patient's situations reflects the characteristics at different time points. Please analyze the relationships between time points and sample characteristics, so as to reveal patterns in the evolution of different categories of diseases over time.

Task 5: For each types of "CN, SMC, EMCI, LMCI and AD", according to the relevant literature, please propose feasible early-intervention methods and diagnostic criterions.

1.3 Our Work

In order to further study the early diagnosis scheme of Alzheimer's disease, it is necessary to mine the relationships between AD and various indicators, and further guide the clinical diagnosis of AD according to the research results. First, we are expected to complete scientific data cleaning, and then, a reasonable AD recognition model is built based on the resulting dataset. In addition, it is necessary to establish a model to analyze the temporal change of AD disease, and we come up with a set of early rise intervention programs and diagnostic criteria.

2. Problem analysis

2.1 Analysis of Task 1

In task 1, it requires pre-processing the attached data and analyzing the correlation between the feature indicators. In order to make subsequent research easier to carry out,in the pre-processing stage, the effectiveness, representativeness, relevance and other characteristics of the selected indicators should be fully considered. In addition, if there is Data Missing , the data are expected to be filtered according to actual situations. For the retained data, it is necessary to strengthen the continuity of the data, which commonly fill with the methods including spline interpolation, average replacement and so on.

2.2 Analysis of Task 2

In task 2, the concepts of "structural brain features" and "cognitive behavioral features" are proposed, besides, these two groups of feature indicators have certain internal correlation , so that it need analysing by group. Next, it is required to design an intelligent diagnosis scheme for AD. As far as Task 2 is concerned, the evaluation mathematical model can be used to evaluate the indicators of medical records, and the AD stage can be divided according to the comprehensive evaluation.

2.3 Analysis of Task 3

In task 3, it has preliminarily classified the medical record diagnosis results——DX_bl. On this basis, it is required to further cluster the three types of MCI medical records and refine the index characteristics of diagnostic medical records. According to the requirements of the topic, we consider building a clustering mathematical model to divide the strongly related data into sets .

2.4 Analysis of Task 4

In task 4, we are required to analyze the relationship between physical examination time and indicator characteristics, and explore the evolution patterns of different types of diseases over time. The characteristics of disease evolution can be analyzed from macro and micro. Macroscopically, the age of patients with Alzheimer's disease reflects the rule of time and can be analyzed by the medical records of multiple patients with age gradients. Microscopically, the vast majority of individual patients have undergone multiple physical examinations, and the physical characteristics indicators will change in each round of physical examination, which can reflect the time variation rule of AD indicators.

2.5 Analysis of Task 5

In Task 5, early intervention In Task 5, early intervention methods and diagnostic criteria were proposed for CN, SMC, EMCI, LMCI and AD. Based on the previous research on characteristic indicators and AD recognition model, the diagnostic criteria of indicators can be refined by referring to relevant literature. One attachment is the guidance manual for the diagnosis of AD, which provides the intervention methods after onset. On the basis of existing literature and research model above , we considered the to integration, induct and innovate for intervention methods.

3. Assumption

1. The data in the attached medical records were objective and there were no detection errors.
2. The selected patient sample did not have other types of diseases, and the Brain Structural Feature were not affected by other diseases.
3. The samples in the attachment were evenly distributed in gender and age, and the sample conditions can represent the overall characteristics of the elderly group.
4. The selected patient sample did not receive external intervention, and the Brain Structural Feature were not optimized by lateral guidance.

4. Symbol Description

Symbol	Meaning	Unit
R_s	Spearman Correlation Coefficient	-
γ	The loss function of Auto-Encoder	-
$x_{(i)}$,$h_{(i)}$,$\tilde{x}_{(i)}$	The i^{th} neurons in input,hidden and output layer	-
$Cost(P,0)$	The loss function of K-Mediods	-
$W(k)$,$B(k)$	The k^{th} closeness value and separation value	-
X,Y,C	Input Set,Output Set, Kernel Value Matrix	-
P,p	Gaussian probability compensation function,Probability value	-

5. Model of Task 1

5.1 Observation of data Characteristics

Before dataset pre-processing, it is necessary to analyze the characteristics and abnormal conditions of the database. By observing the characteristics of the data, we find the following problems:

5.1.1 Data Missing

After browsing the attachment, we found that there were missing values in the attached data, which were divided into two situations:

(1) Most of the samples in the whole column were missing data.

For indicators such as FDG, PIB, TAU and other characteristic indicators, most of the data in the same column are empty. We believe that may be due to the fact that this characteristic is a part of the physical exam and only a small proportion of patients need to examine it. Therefore, the column will be mostly vacant. Due to the large proportion of vacant, they haveare not universal with low reference value for diagnosis .

(2) Only several rows of data are missing by column.

For example, ADAS11, ADAS13, RAVLT_forgetting and other indicators, each owns over 4000 pieces, we think that this may be because the patient did not be involved in the physical examination, forgetting detection or negligence in registration. For such indicators, it is necessary to fill vacant to facilitate the subsequent logical operation of AD recognition

5.1.2 Too many invalid Indicators

There are all 116 fields in the attachment. Clearly, not all indicators are pathologically associated with the identification of AD, such as all of registry numbers, testing time records, and some characteristic indicators consistent with the 1st case of data missing. In order to

reduce the Time and Space Complexity of the model and improve the recognition efficiency, it is necessary to eliminate the invalid indicators . Meanwhile, the reduction of weak correlation indicators can also improve the accuracy of recognition.

5.1.3 Indicator grouping association

In the attachment, we found that the feature has the characteristics of combinational existence. Taking the indicators ABETA_bl, TAU_bl, and PTAU_bl as examples, the indicators in this group basically appear or are absent at the same time, with obvious combination characteristics. We infer that it is caused by the correlation of the test items, the group of indicators belong to the same physical examination item, tend to have strong correlation. Due to the strong correlation, there will be convergence when used as diagnostic indicators.

5.2 Pre-processing of origin Data

5.2.1 Data Culling

According to the above analysis, we cleaned the data set.

First, we has eliminated invalid data.In section 5.1.2, we describe invalid data, which mainly includes: patient diagnostic registration information, diagnosis time, and fields with too few samples.In order to reduce the difficulty of correlation analysis, we adopted a manual deletion scheme for invalid data.

Then, we do sample screening. As for the problems reflected in section 5.1.3, through consulting relevant materials, we learned that the data in this medical record are divided into two categories, including "structural brain features" and "cognitive behavioral features". Since relevant data will be covered in the following topics, this part of data needs to be supplemented with missing values. We have screened the two categories of indicators respectively and roughly classified the categories of characteristic indicators as shown in Table 2.

5.2.2 Data Interpolation

For the second case of missing data, we consider to use **Cubic Spline Interpolation** according to the overall situation of the sample. Cubic spline interpolation is a commonly used **Piecewise Interpolation** method, which can take into account the influence of Runge phenomenon and achieve better interpolation.

Piecewise interpolation splits the interval $[a, b]$ into pieces as $[(x_0, x_1), (x_1, x_2), \cdots, (x_{n-1}, x_n)]$,with two endpoints $x_0 = a$ and $x_n = b$. A cubic spline means that the curve between each plot is a cubic equation. A cubic spline equation satisfies the following conditions:

(1)S(x)= s i(x) on each piece interval $[x_i, x_{i+1}]$ is a cubic equation.

(2)The interpolation condition is satisfied, as $S(x_i) = y_i$ (i=0,1,... ,n)

(3)The curve is smooth explained as $S(x), S'(x), S''(x)$ are all continuous.

With MATLAB, we programmed to fit the cubic Spline Interpolation functions of 91 groups

of characteristic indicators according to the cubic equation as shown in equation 5-2-1. Based on the analytic function, with the mean value method, we have filled in the missing data.

$$f(x) = a_i + b_i x + c_i x^2 + d_i x^3 \quad (a, b, c, d \quad here \quad are \quad constant) \tag{5-2-1}$$

Table 2 Table of Characteristics Category

	APOE4	FDG	ABETA	TAU	PTAU
Cognitive	CDRSB	ADAS11	ADAS13	ADASQ4	MMSE
Behavioral	RAVLT_immediate	RAVLT_learning	RAVLT_forgetting	RAVLT_percforgetting	LDELTOTAL
Features	DIGITSCOR	TRABSCOR	FAQ	MOCA	EcogPtMem
	EcogPtVisspat	EcogPtPlan	EcogPtOrgan	EcogPtDivatt	EcogPtTotal
	EcogSPMem	EcogSPLang	EcogSPVisspat	EcogSPPlan	EcogSPOrgan
	EcogSPDivatt	EcogSPTotal	EcogPtLang		
	FLDSTRENG	FSVERSION	IMAGEUID	Ventricles	Hippocampus
	WholeBrain	Entorhinal	Fusiform	MidTemp	ICV
	DX	mPACCdigit	mPACCtrailsB	EXAMDATE_bl	CDRSB_bl
Structural	ADAS11_bl	ADAS13_bl	ADASQ4_bl	MMSE_bl	RAVLT_immediate_bl
Brain	RAVLT_learning_bl	RAVLT_forgetting_bl	RAVLT_perc_forgetting_bl	LDELTOTAL_BL	DIGITSCOR_bl
Features	TRABSCOR_bl	FAQ_bl	mPACCdigit_bl	mPACCtrailsB_bl	FLDSTRENG_bl
	FSVERSION_bl	IMAGEUID_bl	Ventricles_bl	Hippocampus_bl	WholeBrain_bl
	Entorhinal_bl	Fusiform_bl	MidTemp_bl	ICV_bl	MOCA_bl
	EcogPtMem_bl	EcogPtLang_bl	EcogPtVisspat_bl	EcogPtPlan_bl	EcogPtOrgan_bl
	EcogPtDivatt_bl	EcogPtTotal_bl	EcogSPMem_bl	EcogSPLang_bl	EcogSPVisspat_bl
	EcogSPPlan_bl	EcogSPOrgan_bl	EcogSPDivatt_bl	EcogSPTotal_bl	ABETA_bl
	TAU_bl	PTAU_bl	FDG_bl	AV45_bl	

5.3 Pre-processing Analysis

5.3.1 Features of the Dataset

Based on the analysis above, we can see that the dataset has some characteristics to be taken into account when evaluating correlation of indicators:

(1) Discrete type

The medical records given in the attachment have the characteristics of random distribution, and the samples are random sampling, showing the characteristics of discrete random variables. Approximation processing is required for correlation analysis.

(2) Huge amount of data

More than 16,000 data are given in the attachment, which is a large sample set and the distribution will tend to steady state. According to the rules of medical record selection, we

speculate that the distribution of characteristic indicators follows a normal distribution, but it still needs to be tested.

(3) Nonlinearity

In the previous analysis, the feature metrics are correlated by group, so we can infer that there should be a many-to-many non-linear relationship between the feature metrics, which is the point of the correlation analysis.

When choosing a correlation coefficient, we should take these characteristics into account; for example, the Pearson correlation coefficient is not practical because the two variables must be continuous, normally distributed, and linear. Are there other coefficients that determine the correlation between two sets of variables? The answer is yes, it's the Spearman correlation coefficient,

5.3.2 Spearman Correlation Analysis

Spearman correlation analysis is applicable to determine the correlation between two continuous variables that are not normally distributed (or have outliers that cannot be eliminated),the definition is as follows:

$$R_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{5-3-1}$$

To use Spearman's correlation analysis, there are two criteria:

(1)Continuous variables that aren't normally distributed (or have outliers that can't be eliminated).

If there is any missing data, it is necessary to make up the missing data before further correlation analysis.

(2) There is a monotonic relationship between the variables.

According to the attached information, we carried out the monotony analysis and obtained a total of 91 pieces of valid data. It was found that most groups had monotony.

With the formula 3-1-1,we calculated the spearman correlation coefficient (R_s)of the 91 data. Due to space, only the first 14 pieces of data are shown here, please see Table 3 and complete data are shown in Attached Table 1.

We conducted a hypothesis test (NHST) on 91 groups of characteristic indicators, and the results showed that the significance coefficient of all 91 groups of indicators was less than 0.025, which passed the hypothesis test. This result shows that all indexes reject the null hypothesis, that is, all indexes are significantly different from 0.

At the same time, the spearman coefficient of most indicators showed a strong correlation, such as APOE4, CDRSB and so on. Spearman correlation coefficient $R_s > 0$ is positive correlation; Spearman correlation coefficient $R_s < 0$ indicates negative correlation. The closer it is to 1 and -1, the stronger the correlation is. When the Spearman correlation coefficient is 0, there is no tendency between the two sets of data.Obviously, all 91 indicators own correlation between each other.

Table 3 Table of Spearman coefficient of association(14 in all of 91)

Indicates	**AGE**	**APOE4**	**FDG**	**ABETA**	**TAU**	**PTAU**	**CDRSB**
Correlation R_s	0.033648	0.277973	-0.229796	-0.204629	0.141563	0.151851	0.607232
NHST	1.83E-05	2.28E-285	3.31E-193	9.05E-153	2.55E-73	3.30E-84	0
Indicates	**PTAU_bl**	**FDG_bl**	**AV45_bl**	**Years_bl**	**Month_bl**	**Month**	**M**
Correlation R_s	0.227032	-0.285216	0.175672	-0.171823	-0.171823	-0.172125	-0.172034
NHST	1.63E-188	5.67E-301	1.74E-112	1.26E-107	1.26E-107	5.30E-108	6.89E-108

6. Model of Task 2

6.1 Alzheimer's disease diagnosis pre-analysis

6.1.1 Dataset analysis

Based on the correlation analysis and cleaning dataset of task 1, we further study the intelligent diagnosis method of Alzheimer's disease. In the Data Culling of section 5.2.1, we preliminarily classified two types of indicators: "structural brain features" and "cognitive behavioral features". By comparing the two groups of characteristic indicators, some physical examination items were corresponding to repeated occurrences, such as TAU&TAU_bl, MMSE&MMSE_bl, etc. In our opinion, in order to distinguish duplicate physical examination items, "_bl" will be added to the end of the "structural brain features" category indicator.

In this regard,a reasonable AD diagnosis model needs to be able to generalize the characteristics of the same class of indicators. Meanwhile, in section 5.3.1, we have stated that: The data set presents a nonlinear relationship, and the amount of data is huge, which needs to be considered when designing the model.

6.1.2 Model expectation

Combining the above analysis with practical requirements, we draw the following advantages of intelligent diagnosis model:

(1)High precision

In order to ensure the practical application value of the AD diagnosis model, it is necessary to improve the accuracy of symptom recognition as much as possible. When building a model, you can consider using a test set to test the effect of the model.

(2)Multi-class classification

In Table 1, the type of patient diagnosis is divided into five categories. Due to the large number of diagnostic results, it is necessary to ensure that each type of symptoms can be well

recognized. Combined with the nonlinearity of the characteristic index, the model needs to adapt to the processing of complex nonlinear problems.

(3)Strong correlation parsing ability

In section 5.3, we conducted Spearman test on the feature indicators in the attachment, which proved that all 91 groups of indicators had some correlation. For the evaluation object with a large number of indicators, it is necessary to select a mathematical model with strong analytical ability to reflect the nonlinear association between indicators.

In summary, we consider the use of deep learning algorithms to build mathematical models. Deep learning is a special kind of machine learning, which is a framework formed by using the characteristics of the human brain composed of many neurons for reference, and has stronger learning ability and excellent portability. For the huge data volume of this problem, the model is more expressive; The neural network has more layers and wider width, which can solve the mapping problem with complex logic.

6.2 Model Building——Auto-Encoder

Due to the large number of feature indicators in this problem, we hope that the model can compress the dimensions of the indicators during the training process to make the model more robust. At the same time, the model should conform to the expectations in section 6.1. Eventually, we consider using an Auto-Encoder as the core model.

6.2.1 Introduce of Auto-Encoder

Auto-Encoder is an unsupervised data dimension compression and data feature expression method.The algorithm includes an encoding stage and a decoding stage, and has an antisymmetric algorithm structure.

Auto-Encoder usually has a three-layer structure, as shown in Figure 1 below, which includes an input layer, a hidden layer, and an output layer. The first two layers are connected by an encoder, and the last two layers are connected by a decoder. When the number of neurons in the hidden layer is less than that in the input layer, fewer features can always be used to represent the input data and realize data dimensionality reduction analysis.

6.2.2 The mathematics of Auto-Encoder

In Figure 1,we assume that the input layer neurons are respectively called as $x = [x_{(1)}, x_{(2)}, \cdots, x_{(d_x)}]$.The hidden layer neurons are mapped as $h = [h_{(1)}, h_{(2)}, \cdots, h_{(d_h)}]$. The output layer neurons are $\tilde{\mathrm{x}} = [\tilde{\mathrm{x}}_{(1)}, \tilde{\mathrm{x}}_{(2)}, \cdots, \tilde{\mathrm{x}}_{(d_x)}]$.

Let W be the weight matrix of $d_h \times d_x$ and B be the bias vector. The activation function of the decoder can be as sigmoid, tanh, or rectified.Above all,the mapping relationships from input

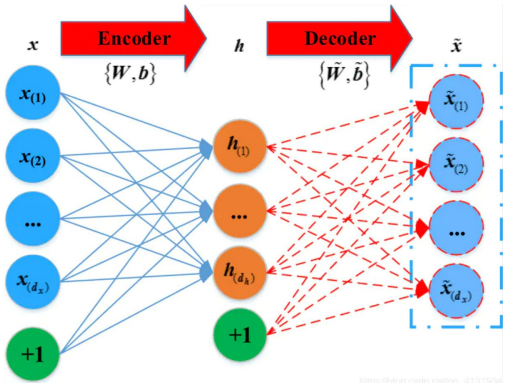

Figure 1 Diagram of the algorithm structure of the Auto-encoder

layer to hidden layer and hidden layer to output layer are as follows:

$$\begin{cases} h = f(x) = s_f(x \cdot W + B) \\ \tilde{x} = \tilde{f}(x) = s_{\tilde{f}}(x \cdot \tilde{W} + \tilde{B}) \end{cases} \tag{6-2-1}$$

According to equation 6-2-1, we programmed the encoder and decoder programs respectively, and established the mapping between input data and target classification data, which can realize the function analysis of complex nonlinear problems.

6.2.3 Cost function Design

Due to the mechanism of deep learning, it is necessary to set a Cost function to mark the convergence of the model. To commit diagnostic classification efficiency as much as possible, the loss function should be as logically concise as possible. Combined with the built-in functions of MATLAB, we choose Mean Square Error(MSE) as the loss function to evaluate the convergence.

$$\Gamma = \sum_{n=1}^{N} \|x^n - g(f(x^n))\|^2 \tag{6-2-2}$$

At the same time, the optimization function for model convergence is to minimize the cost function value:

$$min \quad \Gamma \tag{6-2-3}$$

When MSE tends to 0, it indicates that the prediction results of the model change slightly, which means that the model training has converged. Based on this, the convergence curve of the trained model can be plotted in Figure 2.

6.2.4 Random Forest classifier

Random forest(RF), as the name suggests, is to build a forest in a random way. There are many decision trees in the forest, and each decision tree in the random forest is not related to each other. After the forest is generated, when a new sample is input, each decision tree in the forest will make a judgment separately and obtain the classification of the sample. Random

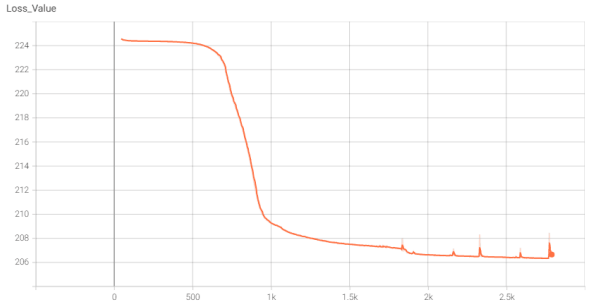

Figure 2 Gradient descent curves for the Auto-Encoder

forests can handle both discrete and continuous values. In addition, random forests can be used for unsupervised learning clustering and outlier detection. On the other hand,AE happens to be an unsupervised compressed parsing algorithm, which is perfectly compatible with random forests.

Step 1. First, start with the selection of random samples from a given dataset.

Step 2. Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3. In this step, voting will be performed for every predicted result.

Step 4. At last, select the most voted prediction result as the final prediction result

6.3 Model solving process

Based on the above theory, the **AE-RF Intelligent Recognition Model** can be established, which can realize high-precision intelligent diagnosis of AD clinical.

In Figure 3 below, according to the above analysis, we designed the confusion matrix of random forest. Under this parameter scheme, we simulated and verified the accuracy of disease diagnosis, and finally concluded that the accuracy of random forest reached **98.2695%**.

Figure 3 Graphic of random forest confusion matrix

7. Model of Task 3

7.1 Criteria for Diagnosing Types

According to the meaning of the question, we preliminarily divided the two levels and five types of disease diagnosis types in Table 1. Based on In addition, it is necessary to continue the refinement to achieve the secondary clustering of the three classes including SMC,EMCI and LMCI.

In Section 5.2, we analyzed the characteristics of these metrics: for example, the data distribution is discrete, and there are a few discrete points that cannot be clustered using K-means clustering.For the disease diagnosis scenario, the clustering model should have strong robustness. To achieve a reasonable classification of discrete point sets, we consider to use **K-Mediods Cluster**.

7.2 Model Building——K-Mediods

7.2.1 The characteristics of K-Mediods Cluster

K-medoids Clustering Algorithm, namely the Center Point Clustering Algorithm, which is based on the improvement of the k-means clustering algorithm. As we know, the k-means algorithm starts by randomly choosing the initial centroids, and only the first randomly chosen initial centroids are the actual points in the set of points to be clustered. If some of the non-centroid points are outliers, or cause the centroid to deviate from the cluster center, k-medoids clustering algorithm solves this problem well.

7.2.2 Cost Function to evaluate convergence

K-medoids clustering algorithm obtains the clustering results by means of partition, and it uses **the Sum of Absolute Differences (SAD)** to measure the quality of the results,as the total Cost Function. In the n-dimensional Euclidean space, the formula for calculating SAD is as follows:

$$\begin{aligned} Cost(P,O) = SAD &= \sum_{m=1}^{k} \sum_{P \in C_i} dist(P_i, O_i) \\ &= \sum_{m=1}^{k} \sum_{P \in C_i} \sqrt{\sum_{j=1}^{nC_i} (P_{ij} - O_{ij})^2} \end{aligned} \tag{7-1-1}$$

When the clustering results tend to converge, the optimization function should be satisfied, that is, the loss value should be minimized:

$$min \quad Cost(P,O) \tag{7-1-2}$$

7.2.3 Description of the clustering process

Partitioning Around Medoids (PAM) is a common approach, where you specify a maximum number of iterations and use a greedy strategy to choose the best cluster. Then, the non-central point is assigned to the nearest central point. If the SAD value is smaller, the clustering quality is better.The clustering process based on PAM method is described as follows:

Step 1. Randomly select k points from the data set to be clustered as the initial center points.

Step 2. Assign the points in the data set to the nearest center point.

Step 3. Enter the iteration until the clustering fitness curve converges (by calculating the SAD) so that the total cost decreases. For each central point O and each noncentral point p, perform the following computation:

(1) Swap points O and P, and recalculate the cost of the partition.

(2) If the exchange increases the cost, cancel the exchange.

The algorithm described above traverses the central point set and the non-central point set in order to calculate the loss value of the new partition. Since the size of the point set to be clustered is different, we consider a random strategy every time we pick a point. In general, if the iteration covers all cases, the final partition tends to be the optimal partition, that is, the best clustering result. It is generally suitable for clustering small point sets. However, if the set of points to be clustered is too large, it is necessary to terminate the iteration by limiting the number of iterations and obtain a clustering result that meets the accuracy requirements.

7.3 Model solving process

7.3.1 Selection of amount of target sets

We mark the classification amount of clustering results as k, which has been indicated in the question. Finally, in order to subdivide the three sets ,including SMC, EMCI and LMCI,for each into three subclasses,it is necessary to cluster sample sets once in each. Therefore, the number of input sets is 1,and the number k of target cluster sets is 3.

7.3.2 Convergence of the fitness function

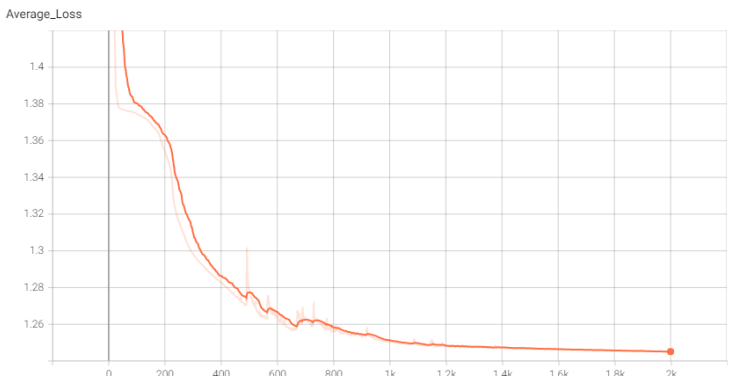

Figure 4 Label graph of the refining clustering

In Section 7.2.2, we design the cost function and optimization objective function of K-Mediods clustering model. For 1416 SMC cases, 2968 EMCI cases and 1738 LMCI cases, we for each trained the clustering model and plotted the average convergence curve as Figure 4:

According to the above model, the clustering results of SMC, EMCI and LMCI datasets are respectively reduced in dimension, presented in a two-dimensional plane, and refined cluster label graphs are defined respectively, as shown in Figure 5, Figure 6 and Figure 7:

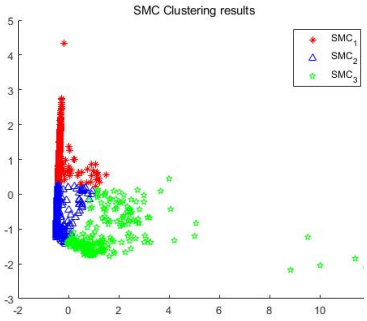

Figure 5 Label graph of the SMC refining clustering

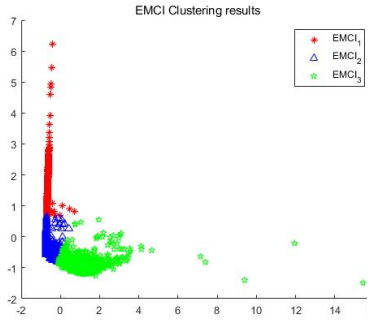

Figure 6 Label graph of the EMCI refining clustering

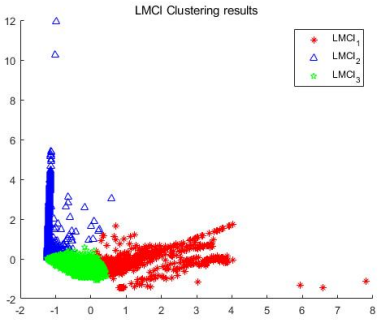

Figure 7 Label graph of the LMCI refining clustering

7.3.3 Convergence of the fitness function

In order to evaluate the clustering model objectively, we evaluate the performance of the model. Since the clustering algorithm is an unsupervised learning algorithm, the evaluation index is more complex. In order to visualize the model performance, we used the Calinski-Harabaz method (CH) to analyze model complexity and goodness.

The CH index measures the closeness by calculating the sum of squared distances between the non-center and the center point, and measures the separation by calculating the sum of squared distances between the center of each class and the center point of the set. CH is defined as the ratio of separation and closeness.

The closeness formula is as follow:

$$W(k) = \sum_{n=1}^{k} \sum_{C(j)=n} \|x_j - \bar{x_n}\| A \tag{7-3-1}$$

The separation formula is as follow:

$$B(k) = \sum_{n=1}^{k} a_n \|\bar{x_n} - \bar{x}\| \tag{7-3-2}$$

The CH index is defined as as below:

$$CH(n) = \frac{B(n)(k-n)}{W(n)(n-1)} \tag{7-3-3}$$

With above theory, we finally calculated that **CH Index of SMC is 2605.776657,of EMCI is 7376.125483,of EMCI is 5834.847534**. The CH index is a positive coefficient, with higher values indicating better clustering performance. Combined with the cluster label graphs shown in Figs. 5, 6, and 7, it can be argued that the K-Medoids clustering model has excellent classification goodness.

8. Model of Task 4

8.1 Sequence analysis of disease evolution

Because CN type patients do not have cognitive impairment and most of them do not have reexamination after the first diagnosis, this problem does not need to consider CN patients in the study of the condition. For the other four groups of patients, we selected the samples from each group who underwent five or more physical examinations to make an observation set (testing every six months), and drew the time series curves of characteristic indicators as shown in the following four figures. We were pleasantly surprised to find: Most of the indicators have linear relationships in time, and there is a time correlation between the indicators.

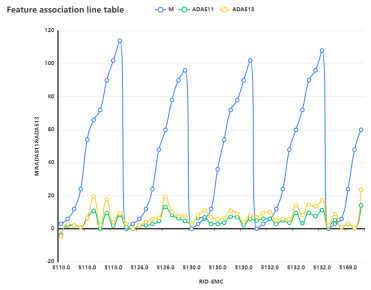

Figure 8 Time series curve of SMC sample characteristic indicators

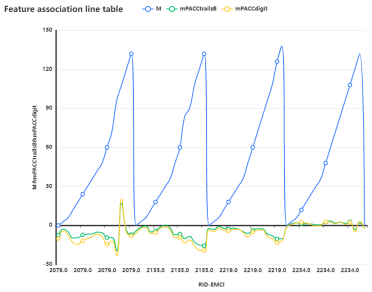

Figure 9 Time series curve of EMCI sample characteristic indicators

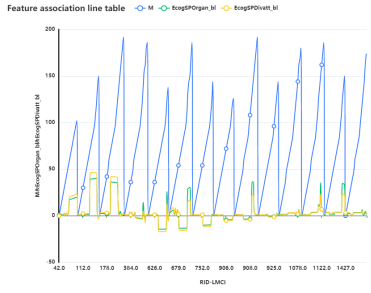

Figure 10 Time series curve of LMCI sample characteristic indicators

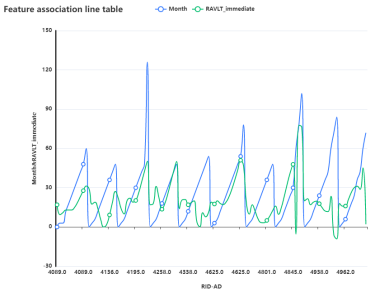

Figure 11 Time series curve of AD sample characteristic indicators

8.2 Model Building——Gaussian process regression time-series analysis

8.2.1 Gaussian process

In statistics, a Gaussian process refers to a probabilistic model in which observations occur in a continuous domain (Figure 12). In a Gaussian process, a finite set of any random variables is required to satisfy a normal distribution, and continuous points in the input space are correlated with normally distributed random variables. Gaussian process is commonly used in machine learning to measure the similarity of kernel function in sample space. Based on a large amount of training data, Gaussian process can be used to predict points distributed at the edge of probability space. The Gaussian process model is highly portable and suitable for regression prediction of nonlinear models.

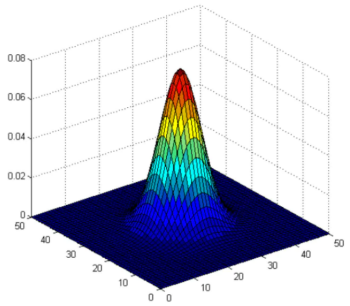

Figure 12 Multivariate Gaussian distribution domain image

8.2.2 Principle of multivariate Gaussian distribution

Multivariate Gaussian distribution extends Gaussian distribution from one-dimensional variable to high latitude space, which makes Gaussian process regression model have stronger applicability. Multivariate Gaussian distributions in Gaussian probability compensation functions are derived as Equation 8-2-1 below:

$$\begin{aligned} P(x_1, x_2 \cdots, x_n) &= \prod_{i=1}^{n} p(x_i) \\ &= \frac{1}{\sigma\sqrt{(2\pi^{\pi})}} exp(-\frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - \bar{x}_i)^2}{2\sigma_i^2}) \end{aligned} \tag{8-2-1}$$

8.2.3 Gaussian process regression Model

Step 1. Determine the sampling point:

$$\begin{aligned} X &= [x_1, x_2, \cdots, x_n] \\ Y &= [y_1, y_2, \cdots, y_n] \\ X &\sim N(\bar{x}(t), C(t, t')) \end{aligned} \tag{8-2-2}$$

Step 2. Identify the sample centers:

In this model, in order to facilitate model derivation, the mean value $\bar{x_i}$is taken here so that the Gaussian distribution center is located at the origin

Step 3. Kernel function of Gaussian regression:

$$c(x, x') = \sigma_f^2 e^{\frac{-(x-x')^2}{2l^2}} \tag{8-2-3}$$

$$C(x, x') = \begin{bmatrix} c(x_1, x_1') & c(x_1, x_2') & \cdots & c(x_1, x_n') \\ c(x_2, x_1') & c(x_2, x_2') & \cdots & c(x_2, x_n') \\ \vdots & \vdots & \ddots & \vdots \\ c(x_n, x_1') & c(x_n, x_2') & \cdots & c(x_n, x_n') \end{bmatrix} \tag{8-2-4}$$

Step 4. Calculate the posterior probabilities:

According to Bayes theorem, based on the data set of observations (X, Y), the posterior probability distribution of the Gaussian process can be calculated as follows:

$$p(f \mid X, Y) \propto p(Y \mid X, f)p(f) \tag{8-2-5}$$

Step 5. Regression fitting was performed:

Based on the posterior probability distribution, time series samples in the data set can be collected and the selected feature indicators can be analyzed in turn:

$$y_*^{(k)} \mid X \sim N\left(0, \begin{bmatrix} C & C_*^T \\ C_* & C_{**} \end{bmatrix}\right) \tag{8-2-6}$$

The matrix representation of covariance matrix, $C_*^T = C_*$,is predicted data collection observation covariance matrix between the data sets, C_*said prediction covariance matrix of the data set; The resulting predicted values satisfy the following probability constraints:

$$p(y^* \mid x^*, X, Y) = \int p(y^* \mid f^*)p(f^* \mid x^*, X, Y)df^* \tag{8-2-7}$$

8.3 Model solving process

According to the multiple Gaussian regression model established above, we successively regressed the characteristic indicators of SMC, EMCI, LMCI and AD patients, obtained the mathematical analytical model of the following formula, and drew the regression GPR curve.

8.3.1 Multiple Gaussian regression mathematical model

Based on the Gaussian regression model above, we generalize the analytic functions for the four types of patients as shown in Equation 8-3-1:

$$f(x) = a_1 \cdot e^{-(\frac{x-b_1}{c_1})^2} + a_2 \cdot e^{-(\frac{x-b_2}{c_2})^2} \tag{8-3-1}$$

For each group of SMC,EMCI,LMCI and AD patients,the regression parameter is as below:

Table 4 Regression parameter for for SMC

Parameter	Min	Average	Max
a_1	1.92	1.93	1.94
b_1	46120.00	46130.00	46130.00
c_1	2249.00	2261.00	2274.00
a_2	1.11	1.11	1.11
b_2	41360.00	41370.00	41370.00
c_2	3422.00	3452.00	3483.00

Table 5 Regression parameter for for EMCI

Parameter	Min	Average	Max
a_1	1.12	1.14	1.15
b_1	45240.00	45260.00	45270.00
c_1	1262.00	1276.00	1290.00
a_2	1.13	1.13	1.13
b_2	42030.00	42050.00	42060.00
c_2	5245.00	5300.00	5354.00

Table 6 Regression parameter for LMCI

Parameter	Min	Average	Max
a_1	-2.40E+14	2.87E+11	2.40E+14
b_1	-1.90E+06	1.06E+05	2.11E+06
c_1	-1.86E+05	1.21E+04	2.11E+05
a_2	2.96	3.09	3.22
b_2	39120.00	39510.00	39910.00
c_2	4670.00	5768.00	6866.00

Table 7 Regression parameter for for AD

Parameter	Min	Average	Max
a_1	2.69	2.88	3.07
b_1	44580.00	44640.00	44700.00
c_1	1670.00	1766.00	1861.00
a_2	4.91	4.92	4.93
b_2	39520.00	39570.00	39610.00
c_2	5475.00	5804.00	6133.00

8.3.2 The fit GRP test for Gaussian process regression

According to the four sets of regression models, we sequentially plot the GPR of the images of SMC,EMCI,LMCI and AD patients over time as shown in Figures 13-16:

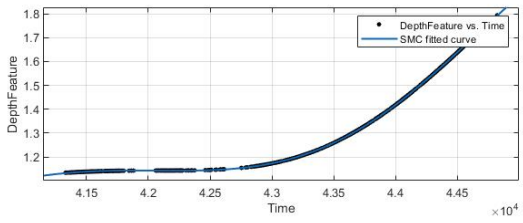

Figure 13 Time series regression of SMC

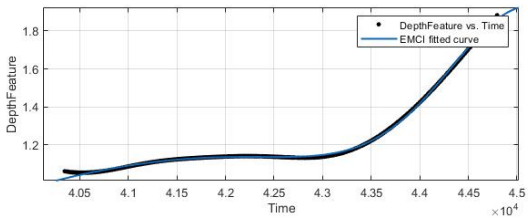

Figure 14 Time series regression of EMCI

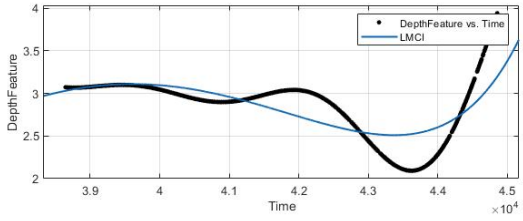

Figure 15 Time series regression of LMCI

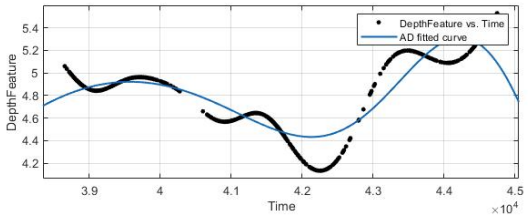

Figure 16 Time series regression of AD

9. Review on Task 5

Based on previous studies, we presented diagnostic schemes for each of the five categories of diseases. In order to be more in line with the clinical diagnosis scenario, we reviewed and integrated the research findings and existing achievements by reviewing the literature, and listed the early intervention measures and diagnostic criteria for five types of diseases including CN, SMC, EMCI, LMCI and AD as follows:

9.1 Diagnosis and intervention of the Cognitive Normal elderly

In the article[1], patients without obvious cognitive function defects are diagnosed by questionnaire evaluation, and the psychological state of the patients can be reflected by designing a structured questionnaire in accordance with psychology. This study showed that for the elderly group (over 60 years of age), if the MMSE test score of the patient is 28.7 points or above, the patient's cognitive ability is normal and very likely belongs to the CN category. If it is lower than this branch, there are some cognitive difficulties,which can be used as a criterion for preliminary diagnosis.

Table 8 Comparison table of MMSE preliminary diagnostic indexes

Stage	Type of Diagnosis	Lower Edge	Average score	Upper Edge
Elderly	***CN***	27	28.7	30
MCI	***SMC***	27	28.9	30
	EMCI	25	27.6	28
	LMCI	24	25.7	26
Post	***AD***	0	21.55	24

In the article[2],it emphasizes that SMC symptoms are a precursor to dementia and that SMC differs from the CN patient population in that SMC may evolve into AD. In this article, patients' memory ability was tested using the CAMCOG memory subscale: "In general, the predictive value of subjective memory complaints is inversely related to the quality of the objective memory Assessment."[2] We can distinguish whether a patient belongs to SMC by analyzing the memory component Ecog in the physical examination data.

If the patient has no tendency for memory extinction and no tendency for subjective paranoia, it can be diagnosed as CN class.CN class does not need too much intervention in pathology, but it should increase the psychological comfort of CN patients, and timely psychological review and communication treatment for this group of patients.

9.2 Diagnosis and intervention for patients with Subjective Memory Complaint

The diagnostic method to distinguish CN from SMC is given in section 9.1. SMC patients have subjective cognitive dysfunction. Clinical manifestations show that their own cognitive dysfunction exists, but there is no significant abnormal cognitive dysfunction stage in objective examination. Therefore, this kind of patients may be able to achieve high scores in the MMSE assessment, but actually have cognitive impairment.

For patients with SMC, timely intervention is needed at an early stage. In literature[3] 's abstract, for patients with SMC, timely intervention is needed at an early stage. It is assumed that the memory training course has an ideal effect in treating the elderly SMC group, and the expectancy change sessions can significantly prevent the deterioration of SMC. The study also indicate that "**memory training programmes** should teach strategies to facilitate memory while addressing older adults' expectations about memory performance"[3].

9.3 Diagnosis and intervention for patients with Early Mild Cognitive Impairment

When the patient's **MMSE and COMCOG** scores are poor, and there is a large gap between the patient and the standard value, the patient should be considered to be diagnosed with EMCI.

In article[4], the authors describe the symptoms of patients with early (EMCI) and late (LMCI) MCI. "Mild cognitive impairment Patients should be identified and monitored for cognitive and functional decline, which is an early sign of Alzheimer's disease, with an increased risk of dementia later in life.

Biomarkers in cerebrospinal fl uid under study include **total tau, phosphotau epitopes, and the 42 aminoacid form of amyloid**.71 kinds of specific phosphotau epitopes have met criteria for an ideal biological marker candidate, with properties for both classification and early diagnosis.

For patients diagnosed with EMCI, it is necessary to take timely **drug intervention** . Clinical trials have shown that the failure of amnestic EMCI is more than 6 months and more than 3 years."From amnestic mild cognitive impairment to Alzheimer's disease, patients taking vitamin E or donepezil were favorably compared with those taking placebo, although significant differences were documented.[4] ".If drug intervention is not taken in time, the patient's memory will gradually decline, and if it reaches the stage of LMCI, the difficulty of treatment will increase greatly.

9.4 Diagnosis and intervention for patients with Late Mild Cognitive Impairment

Patients with LMCI have a history of EMCI and a gradual deterioration of symptoms in the absence of effective treatment. The diagnostic criteria were similar to EMCI. For multiple reexaminations, the scores of each evaluation index decreased.

In addition to the memory training and drug intervention mentioned above, the treatment of LMCI patients also requires long-term care from relatives and friends. Patients at this stage have poor memory skills and may forget important information at any time. In the literature[5] ,it design a comprehensive intervention scheme which combines **Music Training Therapy** and

Associative Emotion Therapy ,proved to be very effective.

9.5 Diagnosis and intervention of Alzheimer's disease

The diagnosis of Alzheimer's disease requires multiple comprehensive tests. In Chapter 5, we developed an intelligent diagnostic model for AD, which can accurately diagnose a patient with AD based on the features collected from multiple physical examinations and the existing model results. The literature[4] states that the diagnosis of AD should be made "in cases with clear evidence of progressive and marked deterioration in intellectual, social, or occupational functioning." This definition was given by Mc Khann and colleagues. In addition, patients with Alzheimer's disease should include deficits in two or more cognitive domains accompanied by progressive deterioration of memory and other cognitive functions.

Because AD patients have completely lost the ability of daily living, they need to be accompanied by people with certain professional knowledge to avoid unexpected safety accidents. In literature[6], studies have shown that giving **Family Care Interventions** is more humanistic and safe for AD patients than allowing AD patients to live in nursing homes. "A program of counseling and support can substantially increase the time spspouse -caregivers are able to care for AD patients at home particularly during the early to middle stages of dementia.[6] "Relatives should regularly arrange for doctors to check the mental state and health of AD patients, so as not to delay treatment due to complications.

10. Strengths and Weakness

10.1 Evaluation of Model 1

10.1.1 Strengths of Cubic Spline Interpolation

Cubic Spline Interpolation is a piecewise low-degree interpolation method, which avoids Runge phenomenon caused by high-degree fitting. Because the interpolation times are reduced, the programming implementation is relatively simple. Compared with other interpolation methods such as Nearest Neighbor Interpolation and Bilinear Interpolation , the interpolation effect is the best. It can overcome the shortcomings of the other two interpolation methods, produce clearer image edges, and has higher accuracy.

10.1.2 Weakness of Cubic Spline Interpolation

The cubic spline interpolation algorithm has high complexity and requires high computational performance. At the same time, the data set to be interpolated must be derivable to the third order.

10.2 Evaluation of Model 2

10.2.1 Strengths of Auto-Encoder&Random Forest

Auto-Encoder is a feature that can learn the data set. When the encoder can make the input data set converge, it can be considered that the model has finished learning. The model has

strong generalization ability, and it is also applicable to data sets with similar structure without retraining.

The Random Forest model can be highly parallelized while training, and the convergence time is short for the large training set in this problem. RF algorithm divides the features by randomly selecting decision tree nodes, and it can still train the model efficiently for high-latitude feature sets. After the RF model is trained, an objective evaluation (CH) of the training performance can be made, and the importance of the features for the output can be known. Due to the use of random sampling, the trained model has small variance, extremely high accuracy and strong generalization ability.

10.2.2 Weakness of Auto-Encoder&Random Forest

The Auto-Encoder is lossy, which also means that after decoding the output data is degraded compared to the input data.Simultaneously ,Auto-Encoder is a deep learning algorithm, which takes a long time to train the model in the early stage, and the algorithm is complex, and it depends on large data sets of high quality as the training set. Using the learning algorithm to make the learning converge by gradient descent, the Phenomenon of Overfitting is easy to occur in the training process.

When the number of decision trees in the random forest is large, the space and time required for training will be large. It is easy to fall into overfitting on a training set with too much noise. RF algorithm is very sensitive to the characteristics of the training set. When there are many characteristics of the input set, RF's decision is greatly affected, and the fitting effect is not good.

10.3 Evaluation of Model 3

10.3.1 Strengths of K-Mediods

The K-mediods algorithm can handle large datasets, and the resulting clusters are compact and well separated from each other. It has low sensitivity to noise, and can effectively avoid the influence of a few outliers and ensure that the clustering center is in the sample set.

10.3.2 Weakness of K-Mediods

K-mediods algorithm must determine the number of clusters and center points in advance, and the selection of the number of clusters and center points has a great impact on the results. The complexity of the algorithm is one order higher than K-Means, and the program is more complex. It is only applicable to the data sets whose clustering results are convex.

10.4 Evaluation of Model 4

10.4.1 Strengths of Gaussian process regression

The Gaussian process regression model is analyzed by the observation results using a kernel function, which reflects the characteristics of the input set. The output data set finally presents Gaussian distribution, which can be adjusted by adjusting the confidence interval and so on, and the model can be generalized by adjusting the kernel function.

10.4.2 Weakness of Gaussian process regression

The Gaussian process regression model is easy to fall into the curse of dimensionality for high latitude feature cases, and the prediction is invalid, the error is too large, and the noise is difficult to eliminate. GPR is a probabilistic analysis method based on random process, but it also has fatal shortcomings: high space complexity and large amount of calculation. When the amount of data reaches a certain scale, the model is difficult to converge and easy to fall into the trap of "overfitting".

11. Improvement Direction

11.1 Improvement of Model 1

(1)Reduce the complexity of interpolation operation

In literature[7] ,Hong Shaohua,Wang Lin, Truong and Trieu-Kien with together provided an improve approach to Cubic Spline Interpolation.They proposed "CSI scheme combines the least-square method with six-point CCI function to improve performance"[7] . For purpose to simplify the decimation and achieve improvement of the computational efficiency., they proposed novel low-complexity implementation algorithm,which can be taken as a direction for improvement.

(2)Add correlation test methods

Only through the Spearman correlation test, the conclusions obtained are one-sided. By adding the correlation test method, the correlation analysis of the characteristic index can be carried out, and the conclusion of comprehensive and objective analysis can be drawn.

11.2 Improvement of Model 2

(1)Improve the Generalization Ability

Increasing the number of encoder and decoder layers, improving the depth of the neural network, and improving the anti-noise ability of the model can improve the generalization ability.

(2) Adjust the sparsity of encoder input data

The use of an undercomplete encoder can be considered for subsequent improvement. If the number of neurons in the hidden layer of the autoencoder is less than the number of input layers, it is called undercomplete, otherwise it is called overcomplete. The incomplete neurons are few and dense. The over-complete structure has many neurons and is sparse, so the under-complete structure is recommended and the sparse penalty is introduced. Adjust the sparsity.

(3)Introduce errors to optimize the convergence mechanism

The randomization method is used to introduce errors to form base learners, and the diversity between base learners is increased to optimize the structural performance of learners. In addition, the error mechanism is introduced, which can also be used to improve the gradient descent method, allowing a certain disturbance of the error, avoiding the premature convergence, "premature" phenomenon.

11.3 Improvement of Model 3

(1) Simplify the objective function

The complexity of the problem is determined by the objective function of clustering. The intra-class scatter matrix and inter-class scatter matrix of K-medoids clustering can be adjusted, and the multi-objective clustering evaluation function can be transformed into a single objective evaluation function.

(2) Introducing the clustering evaluation function

According to the compactness and separability clustering evaluation index measures in Silhouettes validity index, the subsequent improvement can optimize the compactness and separability of evaluation function measures.

11.4 Improvement of Model 4

(1)Reduce denoising misjudgment

For the error problem of Gaussian regression: more accurate and sensitive denoising can be achieved by introducing filters. Literature[8] describes an optimization scheme of Gaussian process regression denoising using the "UKF improved square root" method, which can effectively reduce the prediction error and denoising misjudgment.

(2)Reduce the complexity

If the complexity of the model is too large, we can consider reducing the dimensionality of the prediction set and the observation set. The commonly used dimensionality reduction methods can be divided into three categories: data subset methods, reduced-rank approximation methods, and sparse pseudo-input methods. Literature[9] provides a novel dimensionality reduction method, which can use the "thin plate spline hidden variable" to realize the dimensionality reduction of Gaussian process, and then perform supervised gradient learning, which can better reduce the complexity of regression model.

12. Conclusion

In conclusion,for the sake of further study the early diagnosis scheme of Alzheimer's disease, based on Brain Structural Features and collected in the early study Cognitive Behavioral Features, we studied the correlation of the data collected in the early stage and found that there was a strong nonlinear association. Then, based on the deep learning method, an intelligent diagnosis model of Auto-Encoders-Random Forest (AERF) AD was established, and the accuracy of the model was tested. In order to refine the intervention guidance for AD symptoms, we performed K-Mediods clustering on SMC,EMCI and LMCI patient groups in turn, and obtained subclasses label figures with clear boundaries. It is worth mentioning that we used the sample medical record data to conduct time series analysis, and found that the condition indicators of the same patient mostly changed in the same trend with time, which verified the strong correlation between

the indicators. We established a Gaussian process regression model for the regression analysis of SMC,EMCI,LMCI and AD patients, and analyzed the change trend of the disease with time. Finally, based on the above research conclusions and the results of the existing literature, we give the diagnostic criteria and intervention measures for patients with the five types of diseases, which provides practical guidance for the prevention of Alzheimer's disease.

References

[1] KINSELLA G J, MULLALY E, RAND E, et al. Early intervention for mild cognitive impairment: a randomised controlled trial[J]. Journal of Neurology, Neurosurgery & Psychiatry, 2009, 80(7): 730-736.

[2] SCHMAND B, JONKER C, HOOIJER C, et al. Subjective memory complaints may announce dementia[J]. Neurology, 1996, 46(1): 121-125.

[3] BEST D L, HAMLETT K W, DAVIS S W. Memory complaint and memory performance in the elderly: The effects of memory-skills training and expectancy change[J]. Applied Cognitive Psychology, 1992, 6(5): 405-416.

[4] GAUTHIER S, REISBERG B, ZAUDIG M, et al. Mild cognitive impairment[J]. The lancet, 2006, 367(9518): 1262-1270.

[5] WU T, MA J. Intervention study of music memory training on memory impairment of mild alzheimer's disease: a case study of shanghai m welfare house[J]. Social work and administration, 2018, 18(5): 33-43.

[6] MITTELMAN M S, FERRIS S H, SHULMAN E, et al. A family intervention to delay nursing home placement of patients with alzheimer disease: a randomized controlled trial [J]. Jama, 1996, 276(21): 1725-1731.

[7] HONG S H, WANG L, TRUONG T K. An improved approach to the cubic-spline interpolation[C]//2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 1468-1472.

[8] LI P, SONG S, DUAN G. Improved square-root ukf and its application in rendezvous and docking[J]. Electric Machines And Control, 2010, 14(11): 5.

[9] JIANG X. Research on dimensionality reduction method based on gaussian process[D]. Huazhong University of Science and Technology, 2012.

Appendix

Annex A: Attachment List

Table 9 List of supporting materials in Attachment

Folder Name	File Content	File Description
FirstQuestion		Accessory Package Of Task 1
	code1.m	Code Of Task 1
	data.mat	Dataset Of Task 1
	kinds.mat	Dataset Of Category
SecondQuestion		Accessory Package Of Task 2
	AutoEncoding	Code Package Of Auto-Encoder
	RandomForest	Code Package Of Random Forest
ThirdQuestion		Accessory Package Of Task 3
	code3.m	Code Of K-Medoids
	plotKmediods.m	Code to Draw Graph
	iterative_process.m	Gradient descent iterative method
	findClosest.m	Map for nearest neighbors
	compueCentroids.m	Code Package Of Random Forest
	data_SMC.mat	Dataset Of SMC
	data_EMCI.mat	Dataset Of EMCI
	data_LMCI.mat	Dataset Of LMCI
FourthQuestion		Accessory Package Of Task 4
	code4.m	Code Of Gaussian process Regression
	wc.m	Calculate GRP
	GRP.m	Gaussian Regression Process
	img.m	Code to Draw GRP
	data.mat	Dataset Of Select Sample
	x_predict.mat	Dataset Of Prediction
	ModelParameter.xlsx	Parameter Of Regression Model

Annex B: Attached Table

Attached Table1:Table of Spearman coefficient of association

Indicates	**AGE**	**APOE4**	**FDG**	**ABETA**	**TAU**	**PTAU**	**CDRSB**
Correlation R_s	0.033648	0.277973	-0.229796	-0.204629	0.141563	0.151851	0.607232
NHST	1.83E-05	2.28E-285	3.31E-193	9.05E-153	2.55E-73	3.30E-84	0
Indicates	**ADAS11**	**ADAS13**	**ADASQ4**	**MMSE**	**RAVLT_immediate**	**RAVLT_learning**	**RAVLT_forgetting**
Correlation R_s	0.523210	0.543019	0.522384	-0.481954	-0.502014	-0.407498	0.105255
NHST	0	0	0	0	0	0	3.70E-41
Indicates	**RAVLT_perc_forgetting**	**LDELTOTAL**	**DIGITSCOR**	**TRABSCOR**	**FAQ**	**MOCA**	**EcogPtMem**
Correlation R_s	0.438768	-0.596381	-0.229122	0.359211	0.532265	-0.238238	0.184684
NHST	0	0	4.67E-192	0	0	6.23E-208	2.54E-124
Indicates	**EcogPtLang**	**EcogPtVisspat**	**EcogPtPlan**	**EcogPtOrgan**	**EcogPtDivatt**	**EcogPtTotal**	**EcogSPMem**
Correlation R_s	0.127893	0.121306	0.131362	0.094118	0.086267	0.159749	0.302999
NHST	4.49E-60	3.49E-54	2.64E-63	3.23E-33	3.72E-28	4.18E-93	0
Indicates	**EcogSPLang**	**EcogSPVisspat**	**EcogSPPlan**	**EcogSPOrgan**	**EcogSPDivatt**	**EcogSPTotal**	**IMAGEUID**
Correlation R_s	0.268571	0.249376	0.244868	0.245985	0.269294	0.299936	-0.159097
NHST	8.22E-266	2.84E-228	6.49E-220	5.70E-222	2.75E-267	0	2.36E-92
Indicates	**Ventricles**	**Hippocampus**	**WholeBrain**	**Entorhinal**	**Fusiform**	**MidTemp**	**ICV**
Correlation R_s	0.172055	-0.273745	-0.107115	-0.242867	-0.190110	-0.197811	0.069639
NHST	6.50E-108	1.77E-276	1.43E-42	2.92E-216	9.45E-132	1.08E-142	6.94E-19
Indicates	**mPACCdigit**	**mPACCtrailsB**	**CDRSB_bl**	**ADAS11_bl**	**ADAS13_bl**	**ADASQ4_bl**	**MMSE_bl**
Correlation R_s	-0.574094	-0.573913	0.791973	0.618651	0.651048	0.622840	-0.582933
NHST	0	0	0	0	0	0	0

Attached Table1:Table of Spearman coefficient of association

Indicates	**R_immediate_bl**	**R_learning_bl**	**R_forgetting_bl**	**R_perc_forgetting_bl**	**LDELTOTAL_BL**	**DIGITSCOR_bl**	**TRABSCOR_bl**
Correlation R_s	-0.575767	-0.478747	0.182820	0.495962	-0.817399	-0.086958	0.402825
NHST	0	0	8.04E-122	0	0	1.39E-28	0
Indicates	**FAQ_bl**	**mPACCdigit_bl**	**mPACCtrailsB_bl**	**IMAGEUID_bl**	**Ventricles_bl**	**Hippocampus_bl**	**WholeBrain_bl**
Correlation R_s	0.641059	-0.755006	-0.752652	-0.022005	0.221354	-0.377237	-0.132897
NHST	0	0	0	0.005084856	4.54E-179	0	9.21E-65
Indicates	**Entorhinal_bl**	**Fusiform_bl**	**MidTemp_bl**	**ICV_bl**	**MOCA_bl**	**EcogPtMem_bl**	**EcogPtLang_bl**
Correlation R_s	-0.323875	-0.217691	-0.236901	0.061851	-0.234936	0.133209	0.122649
NHST	0	4.08E-173	1.46E-205	3.24E-15	4.21E-202	4.63E-65	2.33E-55
Indicates	**EcogPtVisspat_bl**	**EcogPtPlan_bl**	**EcogPtOrgan_bl**	**EcogPtDivatt_bl**	**EcogPtTotal_bl**	**EcogSPMem_bl**	**EcogSPLang_bl**
Correlation R_s	0.116655	0.116580	0.116509	0.083452	0.150134	0.231351	0.191877
NHST	3.24E-50	3.74E-50	4.29E-50	1.90E-26	2.46E-82	7.09E-196	3.21E-134
Indicates	**EcogSPVisspat_bl**	**EcogSPPlan_bl**	**EcogSPOrgan_bl**	**EcogSPDivatt_bl**	**EcogSPTotal_bl**	**ABETA_bl**	**TAU_bl**
Correlation R_s	0.220968	0.172953	0.154400	0.175571	0.227362	-0.265182	0.226298
NHST	1.95E-178	4.86E-109	4.99E-87	2.33E-112	4.52E-189	5.95E-259	2.80E-187
Indicates	**PTAU_bl**	**FDG_bl**	**AV45_bl**	**Years_bl**	**Month_bl**	**Month**	**M**
Correlation R_s	0.227032	-0.285216	0.175672	-0.171823	-0.171823	-0.172125	-0.172034
NHST	1.63E-188	5.67E-301	1.74E-112	1.26E-107	1.26E-107	5.30E-108	6.89E-108

Annex C: Code of Mathematical Model

Task 1– Code of Pre-Process&Analysis

code1.m

```
clc;clear
load data.mat
load kinds.mat
[m,n]=size(data);
new_data=zeros(m,n);
x=1:m;
for i =1:n

    x_flag=isnan(data(:,i));
    b=find(x_flag==0);
    new_data(:,i)=spline(b,data(b,i),1:m);
end

%% correlation coefficient
```

```
aa=cat(2,kinds,new_data);
[c,p]=corr(aa,'type','spearman');
```

Task 2– Code of Auto-Encoder&Random Forest

AE.py

```
# Import related library-------------------------------------
import numpy
import numpy as np
import pandas as pd
import torch
from torch.utils.data import Dataset, DataLoader
from torch.utils.tensorboard import SummaryWriter
import Model


# Import data
# ----------------------------------------------------------------
def data(dir):
    df = pd.read_excel(dir, sheet_name='Sheet3', usecols='L:DB')
    columns = df.columns
    taget = df[columns[1]]
    column_need = columns[3:-1]
    data_need = df[column_need]
    return data_need, taget, column_need # Returns the required data column.


# Import data address
# ----------------------------------------------------------------
dir = './origindata.xlsx'

# ----------------------------------------------------------------
trian_data, taget, column_need = data(dir)
print(trian_data)
data = []
print(len(taget))
for i in range(len(taget)):
    data.append(trian_data.iloc[i, :]) # Load data into list
data = numpy.array(data)
# ----------------------------------------------------------------
```

```
# standardization
# ----------------------------------------------------------------
data -= np.mean(data, axis=0)
data /= np.std(data, axis=0)
# ----------------------------------------------------------------
datalist = []
for i in data:
    datalist.append(i) # It becomes 16,208 (1,91) matrices.
print(datalist)



# ----------------------------------------------------------------
# Build data set
# ----------------------------------------------------------------
class dataset(Dataset):
    def __init__(self, datalist):
        self.data = data

    def __getitem__(self, idx):
        data = torch.Tensor(self.data[idx])
        return data

    def __len__(self):
        return len(self.data)



# ----------------------------------------------------------------
# ----------------------------------------------------------------
# instantiation
data = dataset(datalist)
# Network data loader
data_loader = DataLoader(data, batch_size=100, shuffle=False)
# Create a network model
model = Model.AE(in_channels=91).cuda()
# Set the loss function
Loss_Fn = torch.nn.MSELoss().cuda()
# --------
# set optimizer
# Set the learning rate
```

```
Learning_rate = 1e-3
optimer = torch.optim.SGD(model.parameters(), lr=Learning_rate, momentum=0.9)
# Set some parameters of the training network
# Record the number of training sessions
Total_train_step = 0
# Record the number of running rounds of training
Total_train_epoch = 2000
# add tensorboard
writer = SummaryWriter("train1")
# Record the loss of test
total_loss = 0
# Training steps begin
# Establish the receiving list of dimensionality reduction data
s = []
i = 0
# ----------------------------------------------------------------
# Start training--------------------------------------------------------------------
for i in range(Total_train_epoch):
    print("-------------The {} round of training
        begins.-------------------".format(i + 1))
    total_loss = 0
    item = 0
    for data in data_loader:
        item = item + 1
        if torch.cuda.is_available():
            data = data.cuda()
        K, y_predict = model(data)
        if torch.cuda.is_available():
            data = data.cuda()
            y_predict = y_predict.cuda()
        if i == 1999:
            s.append(K.cpu().detach().numpy())
        # print(y_predict)
        # Calculate the actual and result error function
        Loss = Loss_Fn(y_predict, data)
        total_loss = total_loss + Loss.item()
        # optimizer model
        optimer.zero_grad()
        # Backward propagation
```

```
        Loss.backward()
        optimer.step()
    Total_train_step = Total_train_step + 1
    if Total_train_step % 1 == 0:
        print("Training times:{},Average_Loss:{}".format(Total_train_step,
            total_loss / item))
        writer.add_scalar("Average_Loss", total_loss / item,
            global_step=Total_train_step)
# model save
# ----------------------------------
if Total_train_epoch == 2000:
    torch.save(model, "model.pth")
    print(s)
    print("Model Save")
    writer.close()
# ----------------------------------
# Conversion of dimensionality reduction data
# ----------------------------------
k = s[len(s) - 1] # Get the last matrix
del s[len(s) - 1]
c = np.vstack(s)
s = np.row_stack([c, k]) # Convert the matrix in the list vertically into a
    large matrix.
# ----------------------------------
list1 = s[:, 0]
list2 = s[:, 1]
list3 = s[:, 2]
list4 = s[:, 3]
df = pd.DataFrame({'F1': list1, 'F2': list2, 'F3': list3, 'F4': list4}) # data
    loaded into DataFrame
df.to_csv('new_data1.csv', index=False)
# ----------------------------------
```

Model.py

```
import torch
from torch import nn
class AE(nn.Module):
  def __init__(self,in_channels):
```

```
    super(AE, self).__init__()
    #size(100,91)
    self.encoder = nn.Sequential(
        nn.Linear(in_channels,64),
        nn.ReLU(),
        nn.Linear(64,32),
        nn.ReLU(),
        nn.Linear(32,16),
        nn.ReLU(),
        nn.Linear(16,8),
        nn.ReLU(),
        nn.Linear(8,4),
        nn.ReLU())
    self.decoder = nn.Sequential(
            nn.Linear(4, 8),
            nn.ReLU(),
            nn.Linear(8, 16),
            nn.ReLU(),
            nn.Linear(16, 32),
            nn.ReLU(),
            nn.Linear(32,64),
            nn.Linear(64, in_channels),
            nn.Sigmoid(),)
  def forward(self,x):
      '''
      :param x: [b,1,a]
      :return:
      '''
      s = self.encoder(x)
      Y = self.decoder(s)
      return s,Y
```

code2.m

```
%% RF (random forest)
clc;clear
treeNum = 20;% number of decision trees
load data.mat%Load data
%The first 1: end-1 column of the data is characterized by
%End data column is the classifier result.
```

```
temp=randperm(size(data,1));%Random scrambling of known sample data
num=round(size(data,1)*0.8);%The first 0.8*size(data,1) data is used as the
    training set.
dataTrain=data(temp(1:num),:);
dataTest=data(temp(num+1:end),:);
% dataTest=data;
load predict_data.mat%Features of loading forecast data

[dataNum,featureNum]=size(dataTrain);
featureNum=featureNum-1;
[y,RF_model,featuremat]=RF(treeNum ,featureNum,dataNum ,dataTrain,dataTest);
fprintf('\nThe accuracy of random forest classification is:\%f \%\%\n',y*100);
% Forecast and output
load RF_model.mat
load featuremat.mat
% RF_prection = RFprection(RF_model,featuremat,predict_data);
% disp('Random forest classification results are as follows')
% disp(RF_prection);

%%Confusion Matrix

target=categorical(dataTest(:,end));
predict_label=categorical(RFprection(RF_model,featuremat,dataTest(:,1:end-1)));
plotconfusion(target,predict_label)
```

buildCartTree.m

```
function note = buildCartTree(data,k)
 k = k + 1;
 [m,n] = size(data);

  if m == 0
     note = struct();
  else
     currentGini = calGiniIndex(data);
     bestGini = 0;
     featureNum = n - 1;
     for a = 1:featureNum
         feature_values = unique(data(:,a));
         [m1,n1] = size(feature_values);
```

```
        for b = 1:m1
            [D1,D2] = splitData(data,a,feature_values(b,n1));
            [m2,n2] = size(D1);
            [m3,n3] = size(D2);

            Gini_1 = calGiniIndex(D1);
            Gini_2 = calGiniIndex(D2);
            nowGini = (m2*Gini_1+m3*Gini_2)/m;
            gGini = currentGini - nowGini;

            if gGini > bestGini && m2>0 && m3>0
               bestGini = gGini;
               bestFeature = [a,feature_values(b,n1)];
               rightData = D1;
               leftData = D2;
            end

        end
    end
    if bestGini > 0
       note = buildCartTree(rightData,k) ;
       right = note;
       note = buildCartTree(leftData,k) ;
       left = note ;
       s1 = 'bestFeature';
       s2 = 'value';
       s3 = 'rightBranch';
       s4 = 'leftBranch';
       s5 = 'leaf';
       leafValue = [];
       note =
           struct(s1,bestFeature(1,1),s2,bestFeature(1,2),s3,right,s4,left,s5,leafValue);
    else
       leafValue = data(1,n);
       s1 = 'leaf';
       note = struct(s1,leafValue);
    end
  end
end
```

buildRandForest.m

```
function RF = buildRandForest(dataTrain,treeNum)
    RF = [];

    fprintf('Training random forest with %d lessons \n',treeNum);
for a = 1: treeNum
    data = dataTrain(:,:,a);
    note = buildCartTree(data,0);
    fprintf('Lesson %d Tree Training Completed \n',a);
    RF = [RF,note];
    fprintf('= = = = = = = = = = = demarcation line = ================\n');
end
   disp('Random forest training is complete!')
end
```

calAccuracy.m

```
function accuracy = calAccuracy(dataTest,RF_prection)
     [m,n] = size(dataTest);
     A = dataTest(:,n);
     right = sum(A == RF_prection);
     accuracy = right/m;
end
```

chooseSample.m

```
function [data,feature] = chooseSample(data1,featureNum,dataNum)
   [m,n] = size(data1);
    B = randperm(n-1);
    feature = B(1,1:featureNum);
   C= zeros(1,dataNum);
       A = randperm(m);
       C(1,:) = A(1,1:dataNum);
   data= data1(C,feature);
   data = [data,data1(C,n)];
end
```

dataSet.m

```
function [dataAll,featureAll] = dataSet(dataTrain,treeNum,featureNum,dataNum)%
    Dataset Building Sub-function
   dataAll = zeros(dataNum,featureNum+1,treeNum);
   featureAll = zeros(featureNum,1,treeNum);
   for a = 1: treeNum
    [data,feature] = chooseSample(dataTrain,featureNum,dataNum);
    dataAll(:,:,a) = data;
    featureAll(:,:,a) = feature';
   end
end
```

labels_num2.m

```
% Count the number of different types of labels.
function labelsNum = labels_num2(data)
     [m,n] = size(data);

    if m == 0
       labelsNum = 0;
    else
     labels = data(:,n);

    A = unique(labels,'sorted');
    [m1,n1] = size(A);
    B = zeros(m1,2);
    B(:,1) = A(:,1);
    for a = 1:m1
         B(a,2) = size(find(labels == A(a,1)),1);
    end
    labelsNum = B;
    end
end
```

prection.m

```
function A = prection(RF_single,sample)
   if isempty(RF_single.leaf) == 0
      A = RF_single.leaf;
   else
      B = sample(1,RF_single.bestFeature);
```

```
      if B >= RF_single.value
          branch = RF_single.rightBranch;
      else
          branch = RF_single.leftBranch;
      end
      A = prection(branch,sample);
   end
end
```

RFprection.m

```
function RF_prection_ = RFprection(RF,featureGrounp,dataTrain)
    [m,n] = size(RF);
    [m2,n2] = size(dataTrain);
    RF_prection = [];

    for a = 1:n
        RF_single = RF(:,a);
        feature = featureGrounp(:,:,a);
        data = splitData2(dataTrain,feature);
        RF_prection_single = [];
        for b =1:m2
            A = prection(RF_single,data(b,:));
            RF_prection_single = [RF_prection_single;A];
        end
        RF_prection = [RF_prection,RF_prection_single];
    end
    RF_prection_ = mode(RF_prection,2);
end
```

RF.m

```
%% Random Forest Framework
%accuracy is the counted quantity.
%RF_out is a random forest structure model.
%featureGrounp is the characteristic value of each decision tree.
function [accuracy,RF_out,featureGrounp] = RF(treeNum ,featureNum,dataNum
   ,dataTrain,dataTest)
[dataAll,featureGrounp] = dataSet(dataTrain,treeNum,featureNum,dataNum);
RF_out = buildRandForest(dataAll,treeNum);
RF_prection = RFprection(RF_out,featureGrounp,dataTest);
```

```
accuracy = calAccuracy(dataTest,RF_prection);
end
```

splitData.m

```
function [Data1,Data2] = splitData(data,fea,value)

    D1 = [];
    D2 = [];
    [m,n] = size(data);
    if m == 0
      D1 = 0;
      D2 = 0;
    else
       D1 = find(data(:,fea) >= value);
       D2 = find(data(:,fea) < value);
       Data1 = data(D1,:);
       Data2 = data(D2,:);
    end
end
```

splitData2.m

```
function data = splitData2(dataTrain,feature)
   [m,n] = size(dataTrain);
   [m1,n1] = size(feature);
   data = zeros(m,m1);

   data(:,:) = dataTrain(:,feature);
end
```

Task 3– Code of K-Mediods code3.m

```
%% Kmediods clustering algorithm
clc;clear
load data_EMCI.mat
load data_SMC.mat
load data_LMCI.mat
data_EMCI=zscore(data_EMCI);
data_SMC=zscore(data_SMC);
data_LMCI=zscore(data_LMCI);
```

```
K=3;%Number of clusters
[kinds_EMCI,loss_EMCI] = iterative_process(K,data_EMCI,10);
[kinds_SMC,loss_SMC] = iterative_process(K,data_SMC,10);
[kinds_LMCI,loss_LMCI] = iterative_process(K,data_LMCI,10);

plotKmediods(cat(2,data_EMCI,kinds_EMCI),'EMCI')
plotKmediods(cat(2,data_SMC,kinds_SMC),'SMC')
plotKmediods(cat(2,data_LMCI,kinds_LMCI),'LMCI')
```

compueCentroids.m

```
function centroids = compueCentroids(data,cx,K)
% Calculate a new cluster center
    centroids = zeros(K,size(data,2));
    for i = 1:K
% Find the current cluster center with the lowest cost.
        temp = data((cx==i),:);
        [~,I] = min(sum(squareform(pdist(temp))));
        centroids(i,:) = temp(I,:);
    end
end
```

findClosest.m

```
function [cx,cost] = findClosest(data,centroids)
% Divide the sample into the nearest cluster center.
    cost = 0;
    n = size(data,1);
    cx = zeros(n,1);
    for i = 1:n
% euclidean distance
        [M,id] = min(sqrt(sum((data(i,:)-centroids).^2,2)));
        cx(i) = id;
        cost = cost+M;
    end
end
```

iterative_process.m

```
function [cx,cost] = iterative_process(K,data,num)
% generates the best cluster to cluster data into K classes.
```

```
% K is the number of clusters, data is the data set, and num is the number of
    random initialization.
    [cx,cost] = Kmedoids(K,data);
    for i = 2:num
        [cx1,min] = Kmedoids(K,data);
        if min<cost
            cost = min;
            cx = cx1;
        end
    end
end
```

Kmedoids.m

```
function [cx,cost] = Kmedoids(K,data)
% Gather the classified data set data into K classes
%   [cx,cost] = kmeans(K,data)
% K is the number of clusters and data is the data set.
% cx is the cluster to which the sample belongs, and cost is the generation
    value of this cluster.
% Select the number of clusters you want.

% randomly select the cluster center.
    centroids = data(randperm(size(data,1),K),:);
% iterative clustering
    centroids_temp = zeros(size(centroids));
    num = 0;
    while (~isequal(centroids_temp,centroids)&&num<20)
        centroids_temp = centroids;
        [cx,cost] = findClosest(data,centroids);
        centroids = compueCentroids(data,cx,K);
        num = num+1;
    end
%     cost = cost/size(data,1);

end
```

plotKmediods.m

```
function []=plotKmediods(test_data,name)
id1=find(test_data(:,end)==1);
```

```
id2=find(test_data(:,end)==2);
id3=find(test_data(:,end)==3);
figure
hold on
plot(test_data(id1,1),test_data(id1,2),'r*');

plot(test_data(id2,1),test_data(id2,2),'b^');

plot(test_data(id3,1),test_data(id3,2),'gp');

title([name ' Clustering results'])
legend([name '_1'],[name '_2'],[name '_3'])
end
```

Task 4– Code of Gaussian process regression code4.m

```
%% Gaussian regression GRP
clc;clear;
load data.mat;
load x_predict;
[m,n]=size(x_predict);
result=zeros(m,1);
model_test=zeros(1,1);

[result(:,1),model_test(1)]=GRP(1,data,x_predict);

Time=x_predict;
DepthFeature=result;
```

GRP.m

```
%% Gaussian regression GRP
%r indicates the target goodness of fit.%i represents which GRP model
function [y_predict,r0]=GRP(i,data,x_predict)

[m,n]=size(x_predict);
x=data(:,1);
y=data(:,end);%Dependent variable data set
temp=randperm(size(x,1));%Random scrambling of known sample data
num=round(size(x,1)*0.8);
train_x=x(temp(1:num),:);%Take m*0.8 training samples
```

```
train_y=y(temp(1:num),:);%Output of training samples
test_x=x(temp(num+1:end),:);% test sample
test_y=y(temp(num+1:end),:);% test sample output

%Normalized dimensionless
[train_x0,P1]=mapminmax(train_x',0,1);
train_x0=train_x0';
[train_y0,P2]=mapminmax(train_y',0,1);
train_y0=train_y0';
test_x0=mapminmax('apply',test_x',P1)';
test_y0=mapminmax('apply',test_y',P2)';
    %Model training
    yi=train_y0(:,i);
    gprMdl = fitrgp(train_x0,yi,'Basis','pureQuadratic',...
      'FitMethod','exact','PredictMethod','exact');
    %Test set test
    test_predict=predict(gprMdl,test_x0);
    r0 =wc(test_predict,test_y0(:,i),n);\%Fit goodness


    img(gprMdl,train_x0,i,P2);
    disp(['Model' num2str(i) 'Training completed,The test set correction
        goodness of fit is'])
    disp([ 'R_adjusted=' num2str(r0)])
%    img(gprMdl,mapminmax('apply',x_predict',P1)',i);
    %Output forecast
    out=predict(gprMdl,mapminmax('apply',x_predict',P1)');%2*1
    out1=mapminmax('reverse',out',P2)';%2*4
    y_predict=out1(:,i);
end
```

img.m

```
%% draw designs
function []=img(gprMdl,train_x0,i,P2)
[pred,~,ci] = predict(gprMdl,train_x0);
xtest=1:size(train_x0,1);
  pred1=mapminmax('reverse',pred',P2)';%2*4
  pred=pred1(:,i);
  ci1=mapminmax('reverse',ci(:,1)',P2)';
```

```
 ci2=mapminmax('reverse',ci(:,2)',P2)';
 ci(:,1)=ci1(:,i);
 ci(:,2)=ci2(:,i);
figure
plot(xtest,pred,'r','DisplayName','Prediction');
hold on;
plot(xtest,ci(:,1),'c','DisplayName','Lower 95% Limit');
plot(xtest,ci(:,2),'k','DisplayName','Upper 95% Limit');
legend('show','Location','Best');
end
```

wc.m

```
%% Goodness-of-fit calculation function
%R refers to the goodness of fit after adjustment, and is the degree of
   fitting of the regression line to the observed value.

function R2=wc(reverse_out,test_y,k)
U = sum((reverse_out-test_y).^2)/(size(reverse_out,1)-k-1);
P = sum((mean(test_y)-test_y).^2)/(size(reverse_out,1)-1);
R2 = 1 - U/P;
end
```