Problem Chosen	2021 ShuWei Cup Summary Sheet	Team Control Number
D		202111026764

Analysis of the Box Office Volatility Model of the Movie Market

Abstract

In recent years, our country's film industry has continued to show signs of prosperity, and movie box office is an important indicator to measure the success of a movie, and it also reflects the market demand and investment attractiveness of the movie. However, there are many factors that affect the box office of a movie. The factors of the movie itself (the classification characteristics of the movie) and the evaluation of the movie by the network navy and the emergencies will all affect the box office of the movie.

In question 1, we only consider the factors of the movie itself, mainly studying 9 movie characteristics such as duration, prime time, first day box office, and total box office. Then the 9 features were scored and evaluated by the TOPSIS method, and then k-means clustering analysis was performed to divide them into 5 categories, and the effectiveness of the classification was verified.

In the second question, we established a BP neural network model, found the relevant data of 41 movies in 2018 and 2019 through the official website, and put it as a training sample into the neural network. After getting the trained neural network, we started to predict the final box office of the movie, and then conducted an error analysis and found that the error was small.

In the third question, we first crawl the comment data of the face flap and cat's eye through the python crawler, and then use the positive and negative score recognition of the network to establish the DNN neural network, establish the topic word, topic classification and other important index extraction models, use the word frequency statistical method to calculate the importance of a word is used as a feature word, and the top 200 features are displayed in a word cloud diagram. Using 5000 movie data as a sample and doing linear regression analysis, the correlation degree obtained is not very high. Finally, based on the logistic regression method, we proposed a feasible algorithm to identify the network navy.

In the fourth question, because the available data is relatively small, we have established a more accurate SVM neural network prediction model. At the same time, in order to make the prediction results more reliable, we adopted the final box office prediction data of the "Maoyan Professional Edition" as 26 The final box office of each movie is solved, and the results of regression prediction are reasonably explained based on the actual situation.

Keywords: TOPSIS method k-means cluster analysis BP neural network SVM neural network python crawler

Content

1 Introduction

1.1 Background

With the increase in cultural consumption demand of our people, the number of cinemas and screens continues to increase, and my country's film industry continues to show a prosperous scene. The box office of a movie not only reflects the economic value created by the movie for investment companies from the positive side, but also reflects the artistic quality and business strategy of the movie from the side. It is an important factor in measuring the success or failure of a movie. This naturally reflects the market demand and investment attractiveness of film works. If we can predict in advance the acceptance and profitability of film products in the market, it will have a huge impact on the decision-making in all links of the film industry chain. Therefore, accurate prediction of movie box office undoubtedly has important practical significance for risk control and decision-making.

1.2 work

Task 1: Choose effective predictive features that measure the box office of movies, cluster and classify the movies in the provided data set according to the classification features, and verify the effectiveness of the classification.
Task 2: Establish a movie box office prediction model that has a positive impact on the movie box office, and give a classification model for the box office of the movie market based on the collected data. And give the estimated box office forecasts and overall box office forecasts for each category in advance.
Task 3: Collect movie online public opinion evaluation data, establish an algorithm to identify the positive and negative scores of online public opinion, and then establish a model for extracting important indicators such as subject terms and topic classification. Secondly, establish an analysis of the correlation between online public opinion and box office and the box office Finally, according to the problem and current situation, design ideas and specific methods, determine the navy movie scoring network.
Task 4: Analyze the impact of the new crown virus on the movie box office, realistic impact and future predictions. Use the provided data to analyze the impact of different audience needs (30%, 50%, 75%) on movie box office forecasts after the epidemic has stabilized through models.

2. Problem analysis

2.1 Data analysis

We randomly selected 67 movies through major platforms such as Douban and Maoyan. According to the characteristics of movie classification: real-time box office, duration, sequel, Douban score, movie format, number of first-line stars and well-known directors, prime time, first day box office , Total box office and other classification features, to classify the data of the 67 films extracted. We put the

classification results in the supporting material. Among them, we approximated some data, which can make the calculation easier.

2.2 Analysis of Problem One

In question 1, we are based on the classification characteristics of the movie: real-time box office, duration, sequel, Douban score, movie format, number of first-line stars and well-known directors, prime time, first day box office, total box office and other 9 features[1] Used a multi-objective decision analysis method-TOPSIS method to score 26 movies, and then k-means cluster analysis on the scores [2] and finally successfully verified the effectiveness of our classification.

2.3 Analysis of Problem Two

In the second question, we mainly used the BP neural network algorithm [3]. Train the network according to the historical data, input the training set (that is, various information about the box office), and get the trained neural network to predict the movie in 2021. The difference between the prediction result and the actual box office is only 3.5%, indicating that our model establishment is successful.

2.4 Analysis of Problem Three

In the third question, we first use the python crawler to crawl the comment data of the face flap and the cat's eye, and then use the positive and negative score recognition of the network to establish a DNN neural network, use 5000 movie data as samples, and perform linear regression analysis. After chemical processing, it is found that box office has a strong positive correlation with budget, number of votes, and popularity, but not too strong correlation with ratings, and the correlation coefficient is only 0.19.

2.5 Analysis of Problem Four

In the fourth question, we use the more accurate SVM neural network model [4] to predict the box office situation when the occupancy rate is 75%, 50%, and 30%, that is, when the limit rate is 75%, For high box office (such as category A movies), the box office has the greatest impact, but for relatively low box office movies, the box office will increase; but when the limit rate is 50%, the box office of all kinds of movies has a certain degree of decline; when the limit When the rate is 30%, that is, when the epidemic is more serious, the economic situation of the film and television industry is in depression, and the box office has dropped severely. Finally, we reasonably analyzed the results.

3 symbols and assumptions

3.1 Symbol description

Symbolic representation	Symbolic meaning
C_i	Influence

X_{ij}	Low optimization characteristics
X_{ij}^{1}	Highly optimized features
p	Classification features
t	box office

3.2 Basic assumptions

1. Assume that, except for the film classification features that affect the box office of the film studied in the question, other film classification features have no effect on the box office of the movie.
2. Assuming that the final box office predicted by Maoyan Professional Edition is consistent with the real final box office.
3. Assume that there is no intervention by the network navy in the selected film reviews.
4. Assume that cinemas in high- and medium-risk areas are closed and not open.

4. Model

4.1 Model establishment and solution of problem 1

4.1.1 Model analysis

Now there are several movie information in the table. Use each indicator to score these movies with the topsis method, and then perform k-means cluster analysis on the scores.

4.1.2 Model establishment

The TOPSIS method is a simple and efficient multi-index evaluation method. By trending and normalizing the original data, the influence of different influencing factors on the results is eliminated, the optimal value and the worst value are found, and then the evaluations are compared. The degree of closeness between the object and the optimal value and the worst value is used to evaluate the excellence level. The advantage of the TOPSIS model is that it does not require an objective function, nor does it need to pass inspection. It only relies on a few sets of data to find the degree of closeness to the optimal solution and the worst solution, and the good and bad of the target can be compared. We define the degree of closeness to the optimal value as the degree of influence. The value is between 0 and 1. The closer the value is to 1, the closer the evaluation object is to the optimal solution, and the greater the influence; the closer the value is to 0, the closer the evaluation object is to the worst solution, and the smaller the influence. We are ready to analyze the predicted classification features of the movies in the provided data set, so as to obtain classification features that can effectively predict the box office of the movie. The TOPSIS flow chart is shown in Figure (1).

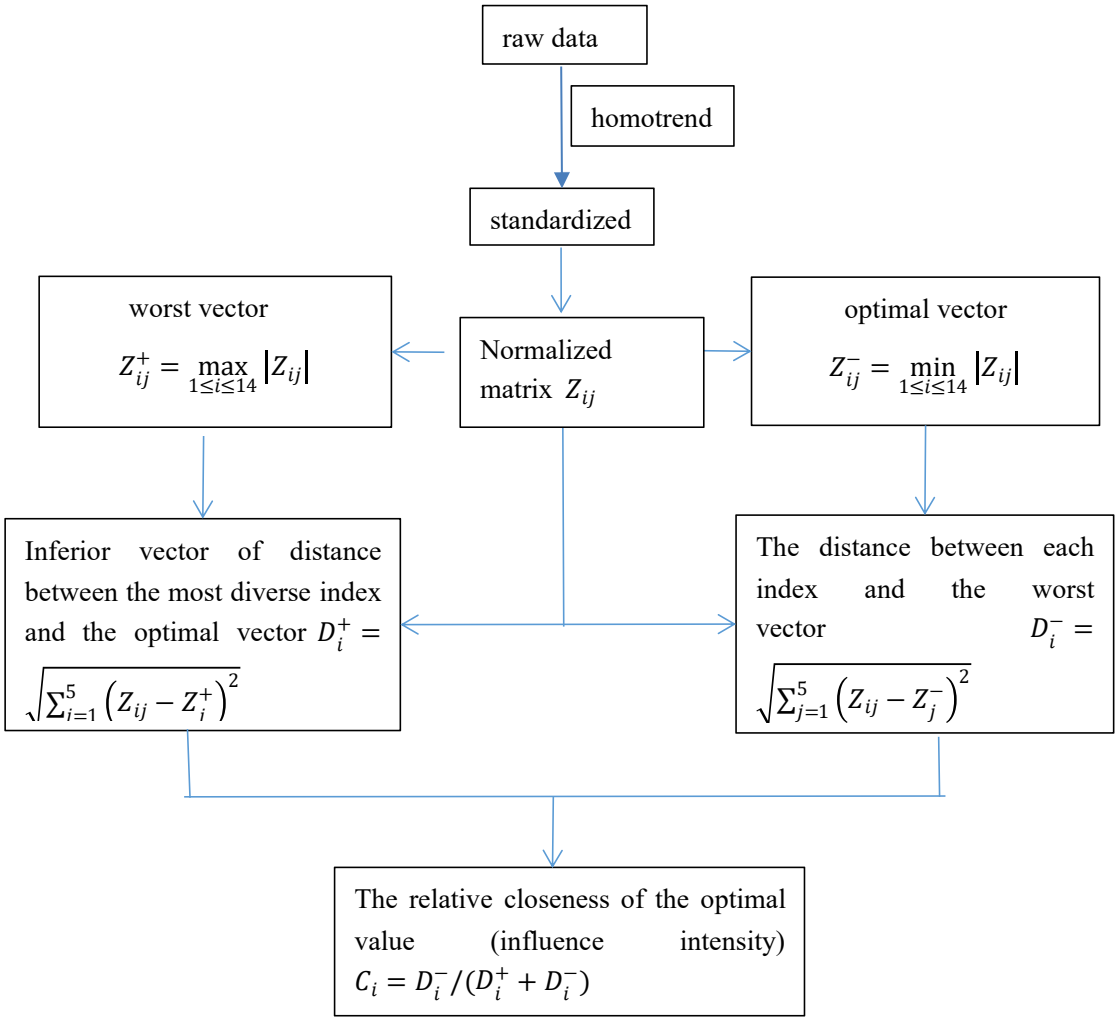


Figure (1) TOPSIS flow chart

4.1.3 Model solution

First, we find the real-time box office, duration, sequel, Douban score, movie format, number of first-line stars and well-known directors, prime time, first day box office, total box office data of 26 movies in 2021. Then the 9 features in Table 2 are treated with homochemistration, that is, the low-optimized features in Table 2 are transformed into high-optimized features, and then the original data matrix of homotrend is established, and we can get the standardized data matrix .

Next, we establish a normalized matrix, which is the matrix obtained after normalizing each column of the standardized data matrix, using the normalized formula:

$$Z_{ij} = X_{ij} / \sqrt{\sum_{i=1}^{26} X_{ij}^2} \ (i = 1,2\cdots,\ 25;\ \ j=1,2,\ 8)$$

Calculate to get the normalized matrix

According to the matrix, the optimal vector and the worst vector are:
Optimal vector Z^+:(0.283, 0.707, 0.255, 0.361, 0.445, 0.408, 0.872, 0.879)
Worst vector Z- (0.130, 0, 0.119, 0.12, 0, 0.022, 0, 0.07).

In $Z_{ij}^1 = \max_{1\le i\le 15}\left|Z_{ij}\right|$, $Z_{ij}^2 = \min_{1\le i\le 15}\left|Z_{ij}\right|$, $i = 1,2,\cdots,25$, j=1,2,$\cdots$, 25.

Calculate the distance between each classification index and the optimal vector and the worst vector, the distance between the worst vector and the relative closeness to the optimal solution, and sort by value.

$$D_i^1 = \sqrt{\sum_{j=1}^{25}\left(Z_{ij} - Z_j^1\right)^2}, \quad D_i^2 = \sqrt{\sum_{j=1}^{25}\left(Z_{ij} - Z_j^2\right)^2}, \quad C_i = D_i^1 / \left(D_i^2 + D_i^1\right)$$

The score evaluation of each movie is shown in Figure (2), and the detailed score table is shown in Appendix 1.

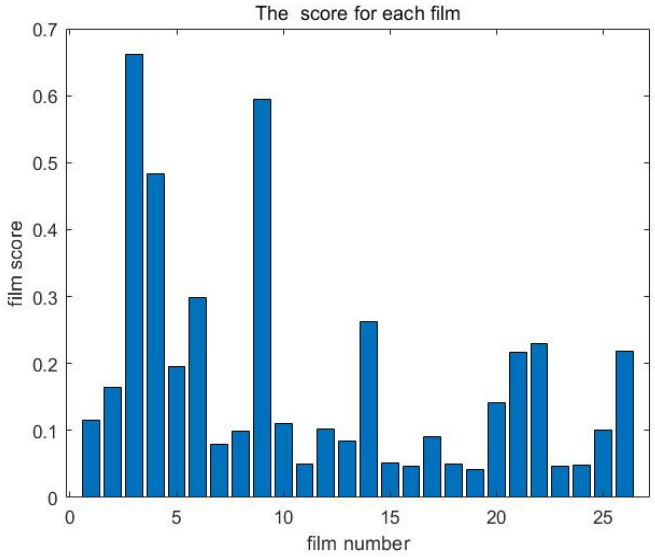


Figure (2) Score evaluation of 26 movies

We use the k-means clustering algorithm for these 26 results. We first randomly select the k-th point as the initial cluster center, and then calculate the distance from each data object to each cluster center, and classify the data object to the nearest The cluster where the cluster center is located; calculate the new cluster center after adjustment. If there is no change in the two adjacent cluster centers, it means that the adjustment of the data object is over. After the data is adjusted, modify the cluster center and enter the next iteration. If all data objects are correctly classified, there will be no change in the cluster center and the algorithm ends [3].

First, calculate and solve the contour value to determine the number of clusters. Since there are 26 data, we think it is more appropriate to divide 4-6. According to the calculation of the contour value, we find that the contour value is larger when the cluster is clustered. The effect is better. Figures (3) and (4) are the contour value maps corresponding to clusters 1-10 and the cluster maps of the scores of each movie when clustering 5 categories.

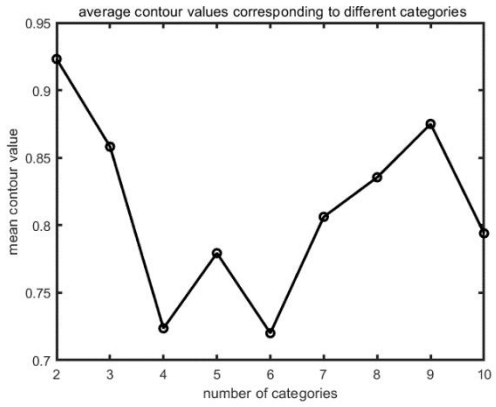


Figure (3) Contour value map movie

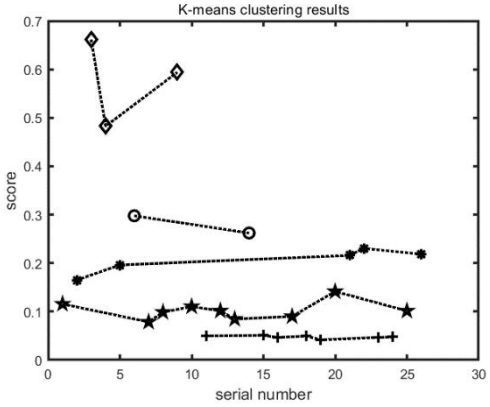


Figure (4) Score cluster map when clustering 5 categories

Perform cluster analysis on the movie scores obtained by the TOPSIS method to obtain 5 cluster centers. According to the scores, we divide them into 5 categories: A, B, C, D, and E. The cluster centers are shown in Table (1):

Table (1) Cluster Center Score Table

cluster center	A	B	C	D	E
score	0.579	0.279	0.205	0.047	0.102

The classification of each movie is shown in Table (2):

Table (2) Classification of movies

A	3 4 9
B	6 14
C	2 5 21 22 26
D	1 7 8 10 12 13 17 20 25
E	11 15 16 18 19 23 24

In order to verify the effectiveness of the classification, we use the Maoyan Professional Edition to average the final box office predictions of these 26 movies with the corresponding categories. as shown in Table 3.

4.1.4 Model checking

Table (3) Average forecast box office table

category	A	B	C	D	E
Average forecast box office (ten thousand yuan)	274801.9	21383.75	8342.478	702.9966	244.671

From the above table, we can see that when the grade category is lower, the box office becomes less and less, so our classification has certain validity.

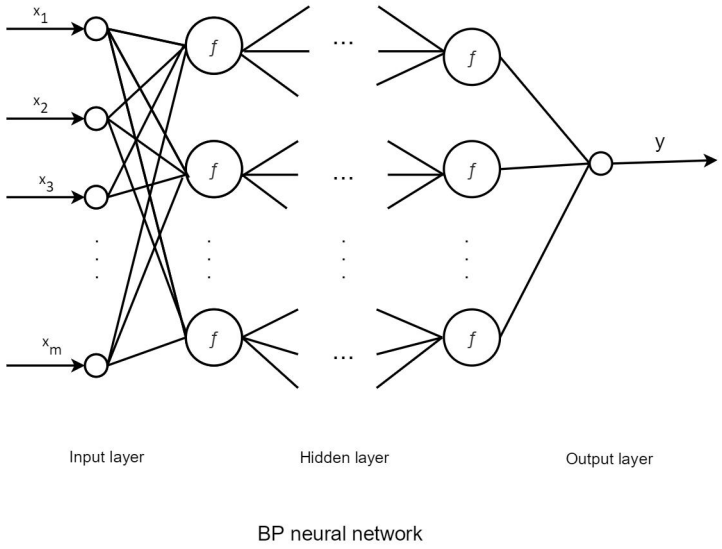
4.2 Model establishment and solution of problem2

4.2.1 Analysis of model

The basic idea of the BP neural network is: the learning process is composed of two processes: the forward propagation of the signal and the backward propagation of the error. In the forward propagation, the input samples are input from the input layer, processed by each hidden layer layer by layer, and then passed to the output layer. If the actual output of the output layer does not meet the expected output, then it goes to

the error back propagation stage. Error backpropagation is to pass the output error back to the input layer layer by layer through the hidden layer in some form, and apportion the error to all neurons in each layer, so as to obtain the error signal of each layer of neurons. This error signal is The basis for revising the weight of each neuron.

The learning and training process of the neural network is the weight of each layer of the error signal forward propagation and error back propagation The adjustment process is a process of repeated weight adjustments. This process has been going on Until the network output error is reduced to an acceptable level.



BP neural network

4.2.2**Establish of model**

In the first step, we need to initialize the network, and assign a range (-1, 1) to each connection weight. Random number, set error function e, given calculation precision value and maximum learning times M.

In the second step, we need to randomly select the k-th input sample and its corresponding expected output.

$$d(k)=\left[d_1(k),d_2(k),...,d_q(k)\right]$$

$$x(k)=\left[x_1(k),x_2(k),...,x_q(k)\right]$$

In the third step, we can calculate the various neuronal input and outputs of the hidden layer.

In the fourth step, we can calculate the partial number of the derivatives of the error function to the neurons of the output layer $\delta_o(k)a$,And then use the connection weights from the hidden layer to the output layer, the $\delta_o(k)$ and the output layer The output of the hidden layer calculates the partial derivative $\delta_i(k)$ of the error function to each neuron in the hidden layer.

In the fifth step, we use the $\delta_o(k)$ of each neuron in the output layer and the output of each neuron in the hidden layer to modify the connection weight $w_{i0}(k)$, and then use the partial derivative $\delta_i(k)$ of each neuron in the hidden layer and the input of

each neuron in the input layer Amend connection rights. Then calculate the global error

$$E=\frac{1}{2m}\sum_{k=1}^{m}\sum_{o=1}^{q}\left[d_o(k)-y_o(k)\right]^2.$$

Finally, judge whether the network error meets the requirements.When the error reaches the preset accuracy or the number of learning times is greater than the set most The number of big steps will end the algorithm.

4.2.3Solution of model

We use the principle to train neural networks using MATLAB and then compare trained data to predict already For known data, the input set is 8 indicators such as movie length and sequel, and the output set is 1 index of predicted box office Standard, after repeated debugging, the number of nodes in the hidden layer is 8 or 9 is more appropriate. The results of using the trained network data to predict the known movie data are shown in Figure (5):

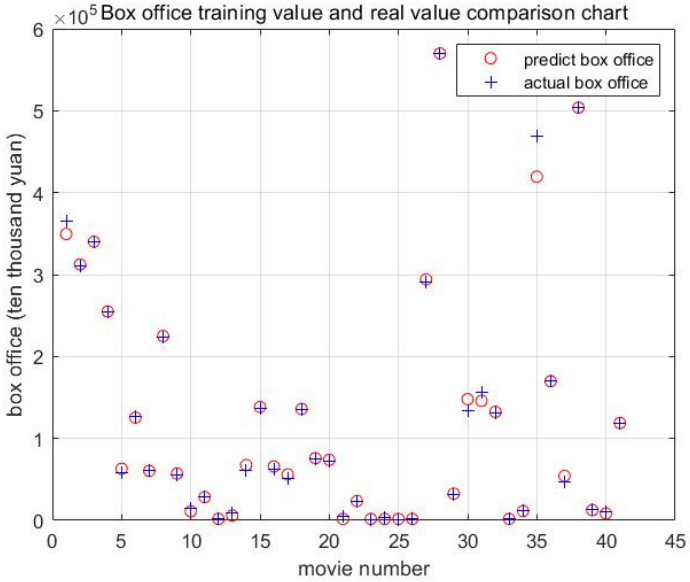


Figure (5) Predict the result of known movie data

From the above figure, we can see that the error between the predicted value of the BP neural network and the true value is very small. On-training is more reliable.

We use formula error calculation formula|each predicted value-corresponding true value|/true value to estimate the error Calculate, get the error of each sample point, and the visualization diagram of error distribution is shown in Figure (6):

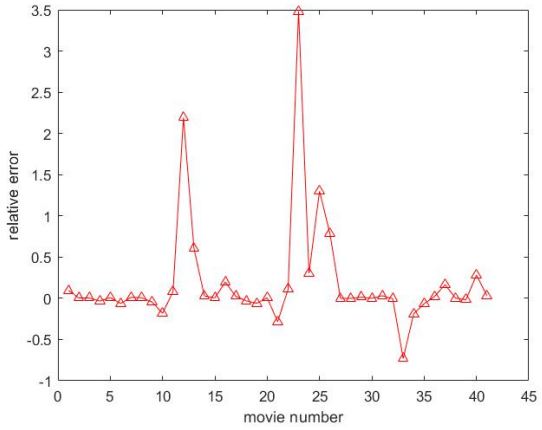


Figure (6) Error analysis diagram

We can intuitively see that the error is within 3.5%, and the vast majority of the error is concentrated within 0.5%, So the training effect is relatively ideal.We then sum the error, yielding the average relative error of 0.1974%。 Thus, the results obtained by this training are ideal, through error analysis and can be used Make real future box office predictions.

Finally, we use the trained BP neural network to predict the box office of the 26 movies. forecast result See appendix 1.

4.3 Model establishment and solution of problem3

4.3.1 Analysis of model

First, we use the python crawler to crawl the comment data of the face and cat's eye. The theoretical structure is shown in Figure (7). The input layer is the crawled raw data, and the output layer is the score calculated by the algorithm.

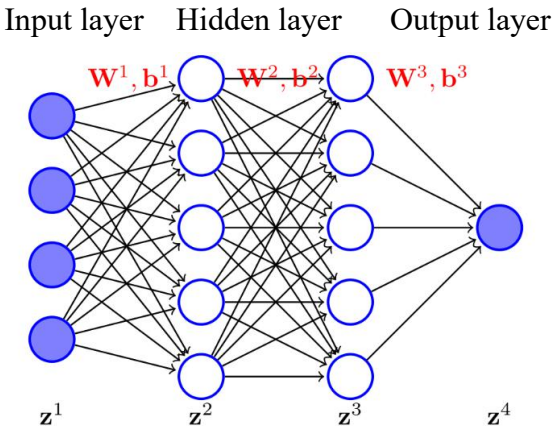
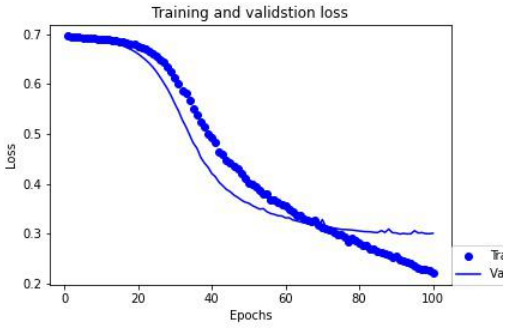


Figure (7) Theoretical structure diagram

DNN neural network (10000, 500, 500, 100, 16, 1) is used for emotion recognition and classification.

The changes in epoch and loss values are shown in Figure (8) and epoch and accuracy in Figure (9) See.



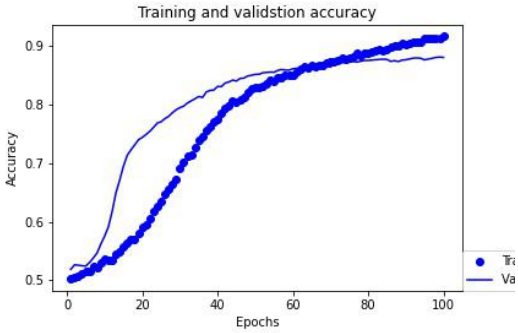
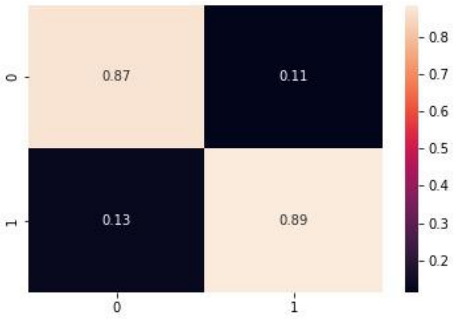


Figure (8) The relationship between poch and loss value Figure (9) Relationship between epoch and accuracy rate.

From the above figure, we can see that the curve gradually converges, indicating that the model effect is relatively ideal, and the accuracy rate is close to 90% The confusion matrix of the prediction result can be obtained by calculation as:



Using this working principle, we can score the data set we crawled with the crawler, and the result is placed in the H column of the file scoring.xlsx-H.

4.3.2 Establish of model

We use the word frequency statistics method to find the importance of words as feature words, and display the top 200 features in word cloud diagrams.

1)Positive comments (score>0) 2) Negative comments (score <0)

3) Intermediate comments (rating=0)

4.3.3 Solution of model

We take 5,000 movie data as a sample and perform linear regression analysis.

Missing values were first processed, and rows with zero data were removed, and then normalized.

	budget	popularity	revenue	runtime	vote_average	vote_count
0	0.516635	0.138638	0.956485	0.172657	0.143022	0.787005
1	0.682424	0.125669	0.301181	0.196226	0.094635	0.256135
2	0.537687	0.089457	0.272369	0.125519	-0.002139	0.253662
3	0.550845	0.095095	0.345636	0.182758	0.207538	0.591092
4	0.577161	0.016990	0.058401	0.071647	-0.034397	0.083347
...	...	...	...	...	...	...
4773	-0.106978	-0.010625	-0.042385	-0.063033	0.175280	-0.016209
4788	-0.107018	-0.027980	-0.041363	-0.059666	-0.018268	-0.063115
4792	-0.106997	-0.032938	-0.043479	0.000940	0.175280	-0.066533
4796	-0.107031	-0.006560	-0.043363	-0.113538	0.094635	-0.023263
4798	-0.106470	-0.016883	-0.042783	-0.100070	0.046248	-0.053806

Correlation analysis is performed after normalization. The scatter diagram is shown in Figure (10):

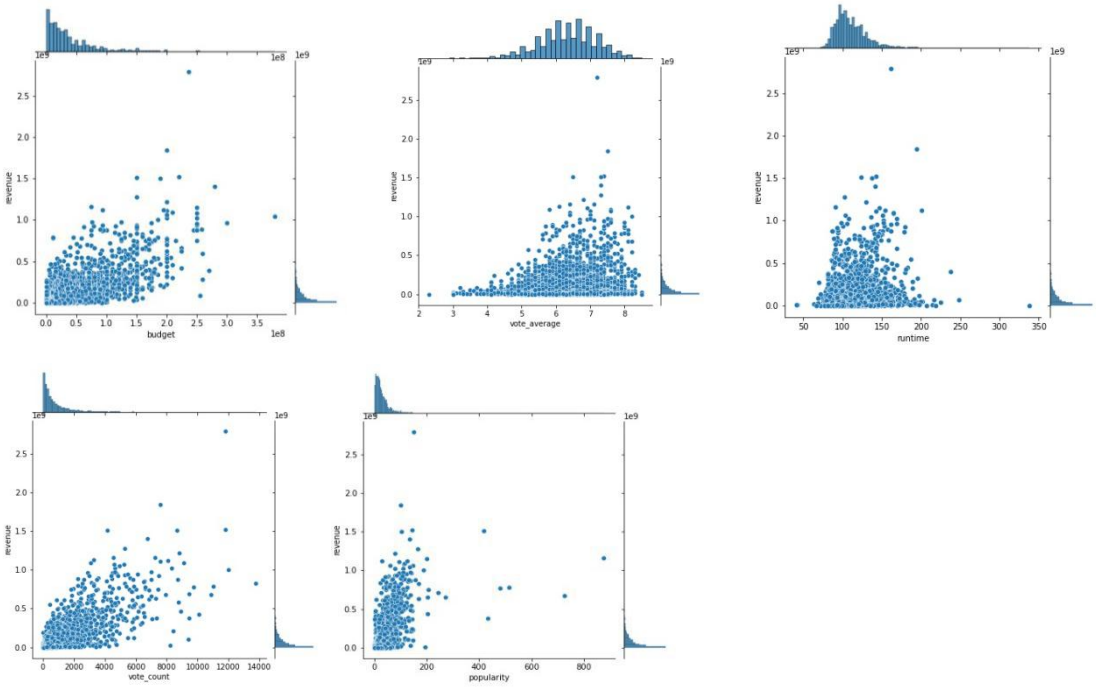


Figure (10) Scatter plot

Based on this, the relationship between the available movie ratings and the predicted value of the box office is shown in Figure (11):

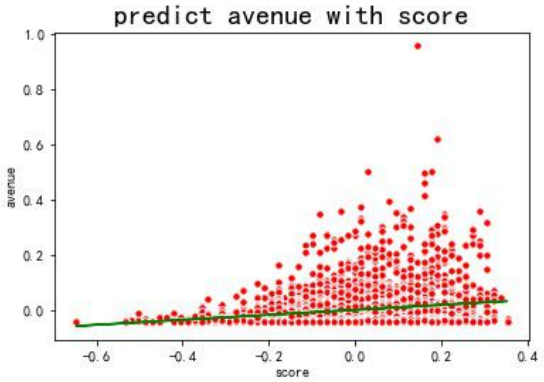


Figure (11) The relationship between movie ratings and box office forecasts

We can get the correlation coefficient as:

	budget	popularity	revenue	runtime	vote_average	vote_count
budget	1.000000	0.431744	0.705306	0.229712	-0.035757	0.539997
popularity	0.431744	1.000000	0.602122	0.182388	0.288189	0.749005
revenue	0.705306	0.602122	1.000000	0.233236	0.188014	0.756143
runtime	0.229712	0.182388	0.233236	1.000000	0.386199	0.258101
vote_average	-0.035757	0.288189	0.188014	0.386199	1.000000	0.380825
vote_count	0.539997	0.749005	0.756143	0.258101	0.380825	1.000000

According to the correlation coefficient and then the heat map can be obtained, as shown in Figure (12):

Figure (12) Heat map

Through the above, we can find that box office has a strong positive correlation with budget, number of votes, and popularity. There is no strong correlation between scores (coefficient of 0.19).

We can perform regression prediction to get:
['budget','popularity','runtime','vote_average','vote_count'] correlation system The numbers are in order:
[2.38588887e-01 , 8.99146254e-02 , 9.13900703e-04 , 1.70145745e-04 , 3.17651387e-01]

4.3.4 Design ideas and specific methods

Ideas and specific methods for designing and identifying movie scoring network navy.Since the purpose of the navy and normal users is essentially different, and their behavior differences will be more obvious, we will convert the navy identification problem into a two-category problem. According to account attributes (number of friends, number of fans, ratio of fans and friends) and user behavior (frequency of posting, offline time, number of mentions and mentions), the navy is distinguished from normal users to collect relevant data. Finally, Use logistic regression algorithm to realize a recognition method to meet the requirements of accuracy and efficiency.

x is a multi-dimensional vector, which is the account content characteristics and user behavior characteristics, and θ is the corresponding parameter of x.

Let N denote the collection of users of a certain movie platform:

$$N=\{x1,x2,x3,...xi,...,x|N|\}$$

xi is the i-th user, all users are set A={a,b}, set a is the normal user set, b is the navy user set, then the objective function of the two classification is, the simplified objective function is

$$h(xi)=\begin{cases}1,xi\in A\\0,xi\in B\end{cases}$$

The simplified objective function is the mapping of G→{0，1} to represent the navy. Now suppose 1 is the navy user and 0 is the normal user.

Introduce the Sigmoid function

$$h_\theta(x)=\frac{1}{1+e^{-\theta Tx}}$$

x={x1,x2,...x ,...,x },

When $h_\theta(x)=0$, the detected user is a normal user, and when $h_\theta(x)=1$, the detected user is a navy. Finally, the maximum likelihood method is used to solve the value of θ to find the best regression parameter θ.

4.4 Model establishment and solution of problem4

4.4.1 Analysis of model

In the second question, we used the BP neural network prediction model. Since there is less data available in this question and there is no box office data for movies that limit the occupancy rate for training (the training set is insufficient), we use a

more accurate SVM The neural network model respectively predicts the box office situation under the restrictions of 75%, 50%, and 30% occupancy rates, and analyzes them reasonably.

The support vector machine (svm) neural network is a type of neural network model. Its core idea is to establish a classification hyperplane as a decision surface, and then divide the countless sample points into two major categories: positive and negative. Class isolation, choose a suitable isolation plane, so that the average distance between the two types of sample points from the plane is the longest. That is, divide a set of sample points into two groups as far as possible. The theoretical basis of support vector machines is the theory of statistical learning. More precisely, support vector machines are the embodiment of structural risk minimization. This principle is based on the error rate of the learning machine on the test data and the sum of a term that depends on the VC dimension. In the separable mode, the support vector machine has a value of 0 for the previous term and minimizes the second term .

The support vector machine model has the following advantages:

(1) Versatility: It can fit and regress the optimal solution function in a wide variety of data sets

(2) Accuracy: The prediction accuracy and accuracy of the SVM support vector machine neural network model is generally higher than that of the BP neural network

(3) Simplicity: the realization of the method only needs to use simple optimization algorithm technology

Based on the advantages and characteristics of the support vector machine (SVM) neural network prediction model, we can use it to predict the box office situation under the restrictions of 75%, 50%, and 30% occupancy.

4.4.2 Establish of model

In order to make the prediction results more reliable, we adopted the final box office prediction data of "Maoyan Professional" as the final box office of 26 movies. Since we are doing regression prediction, each variable is treated as both an independent variable and a dependent variable. Due to the restriction on the occupancy rate, we will forecast by multiplying the number of viewers index by the restriction rate. We mainly reflect the solution process when the occupancy rate is 75%, and the solution process for other restriction rates is similar.

First, we select the independent variable and the dependent variable according to the model. Select each movie's duration, sequel, Douban rating, movie format, number of first-tier stars, prime time, first day box office and number of movie viewers as the original independent variables, and finally predict box office as the original dependent variable. Each movie's duration, sequel, Douban Score, movie format, number of first-tier stars, prime time, first day box office and number of movie viewers are predictive independent variables, and final predictive box office is the predictive dependent variable.

Then preprocess the data. Use the mapminmax function in MATLAB to normalize

the data. Choose the best SVM parameter c&g for regression prediction analysis. First make a rough selection, and then make a fine selection based on the results of the rough selection. We use the libsvm toolbox in MATLAB for parameter selection.

The input set is: training set label train_label, training set train, the minimum value of the penalty function change range cmin, the maximum value of the penalty function change range cmax, the minimum value of the parameter g change range gmin, the maximum value of the parameter g change range gmax, The selection of parameter v is to divide the test set into several parts, the step size cstep of parameter c, the step size of parameter g, and the step size when the mse diagram is displayed.

The output set is: the lowest mean square error (mse) in the Cross Validation process. The best value of parameter c and the best selected value of parameter g.SVM parameters obtained by running the program. Among them, the roughly chosen c=16, g=8, mse=0.052624. Finely selected c=16, g=8, mse=0.043761.

The result of parameter selection is shown in Figure (13):

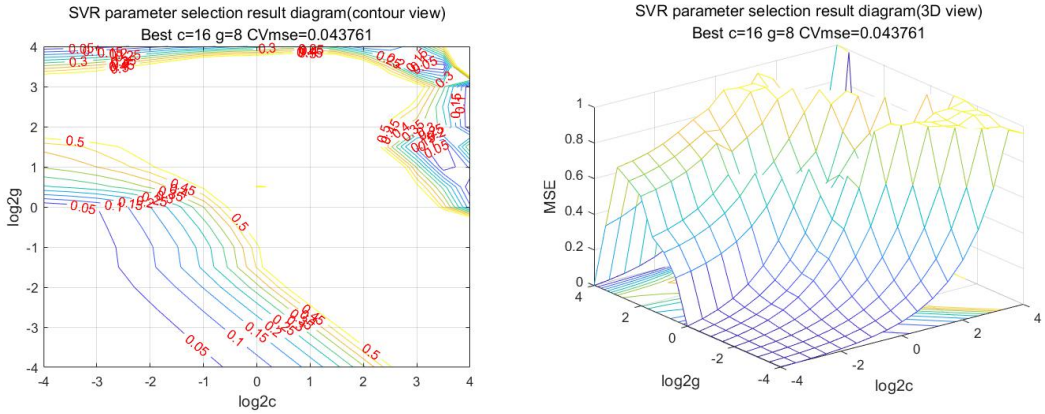


Figure (13)

Contour map and three-dimensional view of SVM optimal parameter selection

4.4.3 Solution ofmodel

Start training and prediction. Use the best parameters c and g obtained above to train the parameters of the SVM, and then compare the box office prediction results with an occupancy rate of 75% with the final box office results without the 75% limit predicted by the Maoyan Professional Edition. The error graph of the training result is shown in (14).

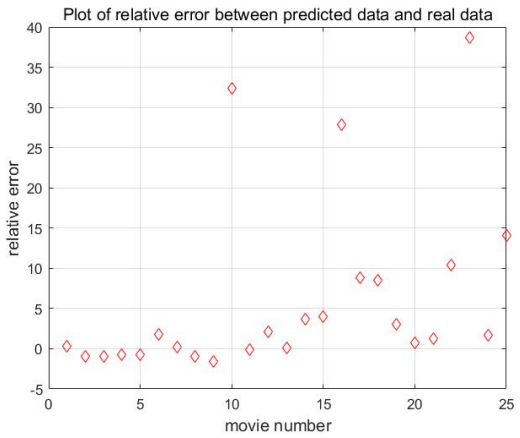


Figure (14) Error test of training results

Among them, the error=(prediction result-original data*75%)/original data. It can be seen from the figure that most of the error results are within 5%. Taking the average of these error values, the final average error is 1.53%. The training result Better, which verifies the accuracy and feasibility of the training value.
The visualized line chart of the comparison result is shown in Figure (15): (see the appendix for the prediction result).

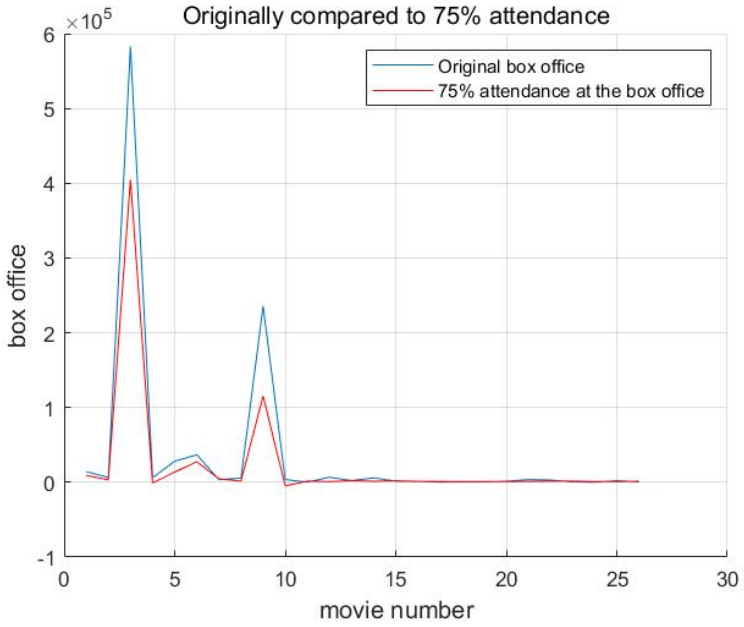


Figure (17) Visualized comparison chart of 50% and 30% occupancy rates

It can be seen from the figure that when the occupancy rate is 50%, the box office of movies with high box office and large audience attraction further declines, and when the occupancy rate is limited to 30%, almost all movie box office forecasts are close to zero. This is also reasonable, because as the degree of occupancy restriction increases, the audience of high-box office (category A) movies further decreases, and when the degree of restriction is very large, it means that the epidemic level is higher, and audiences rarely go to movie theaters and other crowds. In densely populated

places, the sales of movie tickets (box office) approached zero.

(7) Calculate the economic impact (box office) of various films in the film and television industry of different occupancy restriction policies. According to the grouping in question 1, we will predict the box office/total box office of different groups according to the total occupancy rate limit, and calculate the box office impact of each group under each limit rate.

Table (4) The box office impact of each group when the restriction rate is 75%:

Group	A	B	C	D	E
Influence	73.245%	65.21%	132.56%	105.71%	125.65%

Table (5) The box office impact of each group when the restriction rate is 50%:

Group	A	B	C	D	E
Influence	52.15%	42.92%	59.12%	64.75%	46.8%

Table (6) The box office impact of each group when the restriction rate is 30%:

Group	A	B	C	D	E
Influence	8.56%	9.85%	3.44%	4.82%	2.14%

This set of data further confirms the impact of the different restriction rates mentioned above on the box office. That is, when the restriction rate is 75%, the box office impact on high box office (such as category A movies) is the greatest, but the box office of relatively low box office movies will increase; but when the restriction rate is 50%, the box office of all kinds of movies All have declined to a certain extent; when the restriction rate is 30%, that is, when the epidemic is relatively serious, the economic situation of the film and television industry is in depression, and the box office has dropped severely.

5. Strengths and Weakness

5.1 Strengths

1. The data were processed using Excel, MATLAB, and various charts were created to be simple, intuitive, fast and accurate the sure way shows the final results;
2. After obtaining the final results, we performed the error analysis and found very small errors compared to the actual case;
3.The data sought in this paper are mainly derived from official data and have high credibility, making our results more realistic.
4. When solving the second and fourth questions, the intelligent algorithm of machine learning can be used to predict the trend of future data trends more accurately than traditional mathematical models.
5. Solve the third question. Using web crawler technology, a large amount of data can be used as a training set, which makes the training results more accurate.

5.2 Weakness

1.In the process of establishing the model, there are many factors that affect the box office. This article only selects some of the influencing factors, and there are certain circumstances limitation;
2.In the process of searching for data, we approximated part of the data, so the result obtained is different from the actual situation set gap.

6. Conclusion

In question 1, we used one of the nine characteristics of the movie: real-time box office, duration, sequel, Douban score, movie format, number of first-line stars and well-known directors, prime time, first day box office, and total box office. Multi-objective decision analysis method-TOPSIS method scores 26 movies, and then performs k-means cluster analysis on the scores and finally successfully verifies the effectiveness of our classification.

In the second question, we mainly used the BP neural network algorithm. Train the network according to the historical data, input the training set (that is, various information about the box office), and get the trained neural network to predict the movie in 2021. The difference between the prediction result and the actual box office is only 3.5%, indicating that our model establishment is successful.

In the third question, we first use the python crawler to crawl the comment data of the face flap and the cat's eye, and then use the positive and negative score recognition of the network to establish a DNN neural network, use 5000 movie data as samples, and perform linear regression analysis. After chemical processing, it is found that box office has a strong positive correlation with budget, number of votes, and popularity, but not too strong correlation with ratings, and the correlation coefficient is only 0.19.

In the fourth question, we use a more accurate SVM neural network model to predict the box office situation when the occupancy rate is 75%, 50%, and 30%, that is, when the limit rate is 75%, for high box office (For example, category A movies) have the greatest influence on the box office, but the box office of relatively low box office movies will increase; but when the restriction rate is 50%, the box office of all kinds of movies has a certain degree of decline; when the restriction rate is 30 %, that is, when the epidemic is more serious, the economic situation of the film and television industry is in depression, and the box office has dropped severely. Finally, we reasonably analyzed the results.

References

[1]Research on movie box office prediction based on Bayesian classification model[J]. Li Zhenxing,Han Lina,Shi Nan.Computer and Digital Engineering.2020(09)

[2]Credit strategy analysis of small, medium and micro enterprises based on K-Means cluster analysis[J]. Zhao Lingling, Chen Jin, Li Xiaoying, Zhu Jiaming. Journal of Science of Normal University. 2021(09)

[3]Zheng Jian,Zhou Shangbo.Movie box office prediction modeling based on neural network[J].Computer Applications,2014,034(003):742-748.

[4]SVM-based deep learning classification research review[J] Fengjuan Qiao, Hongli Guo, Wei Li, Bin Li. Journal of Qilu University of Technology. 2018 (05)

Appendix1

电影名称	票房（单位：万元）
扬名立万	16200
梅艳芳	6416.1
长津湖	567800
007：无暇赴死	66800
丛林奇航	21300
沙丘	33300
不老奇事	3939.9
入殓师	6539.7
我和我的父辈	147500
天书奇谭 4k 纪念版	2713.3
青春作伴好还乡	93.3
第一炉香	6422.8
乌海	1449.4
不速来客	6077.6
守望青春	1427.1
越界	1184.9
邓小平小道	233
我的初恋十八岁	530
蜜熊的音乐之旅	549.9
特种部队：蛇眼起源	1127
三湾改编	2840.3
我的父亲焦裕禄	2884.2
猪迪克之蓝海奇缘	538.9
信者	141.4

兰心大剧院	2167.6
永不消逝的电波	254.1

Appendix2

```
Question1
clc
clear
x=xlsread('The first question.xlsx','Sheet1','B2:I27');
[n,m]=size(x);
zh=zeros(1,m);
d1=zeros(1,n); %Minimum matrix
d2=zeros(1,n); %Maximum matrix
c=zeros(1,n);    %Proximity
%Normalized
for i=1:m
    for j=1:n
        zh(i)=zh(i)+x(j,i)^2;
    end
end
for i=1:m
    for j=1:n
        x(j,i)=x(j,i)/sqrt( zh(i));
    end
end
%Calculate the distance
xx=min(x);
dd=max(x);
```

```
for i=1:n
    for j=1:m
        d1(i)=d1(i)+(x(i,j)-xx(j))^2;
    end
    d1(i)=sqrt(d1(i));
end
for i=1:n
    for j=1:m
        d2(i)=d2(i)+(x(i,j)-dd(j))^2;
    end
    d2(i)=sqrt(d2(i));
end
%Calculate proximity
for i=1:n
    c(i)=d1(i)/(d2(i)+d1(i));
end
c_1=c';

%Calculate contour values
numC=10;
for i=1:numC
    kidx = kmeans(c_1,i);
silh = silhouette(c_1,kidx); %Calculate the contour value
    silh_m(i) = mean(silh);      %Mean contour values were calculated
end

figure
plot(1:numC,silh_m,'ko-', 'linewidth',2)
set(gca,'linewidth',2);
xlabel('number of categories');
```

```
ylabel('mean contour value');
title(' average contour values corresponding to different categories');

%kMean cluster calculation process
[idx,ctr]=kmeans(c',5); % Clustering using K-means method
%Extract the sample number of the same category
c1=find(idx==1); c2=find(idx==2);
c3=find(idx==3); c4=find(idx==4);
c5=find(idx==5);
figure;
F1 = plot(find(idx==1), c(idx==1),'k:*', ...
        find(idx==2), c(idx==2),'k:o', ...
        find(idx==3), c(idx==3),'k:p', ...
        find(idx==4), c(idx==4),'k:d', ...
        find(idx==5), c(idx==5),'k:+');
set(gca,'linewidth',2);
set(F1,'linewidth',2, 'MarkerSize',8);
xlabel('serial number','fontsize',12);
ylabel('score','fontsize',12);
title('K-means clustering results');
disp('clustering results：');
disp(['class 1:' ,'center：',num2str(ctr(1)),'    ','class sample number：', num2str(c1')]);
disp(['class 2:' ,'center：',num2str(ctr(2)),'    ','class sample number：', num2str(c2')]);
disp(['class 3:' ,'center：',num2str(ctr(3)),'    ','class sample number：', num2str(c3')]);
disp(['class 4:' ,'center：',num2str(ctr(4)),'    ','class sample number：', num2str(c4')]);
disp(['class 5:' ,'center：',num2str(ctr(5)),'    ','class sample number：', num2str(c5')]);

Question2
clc
clear
```

```
shichang=xlsread('First question.xlsx','Sheet2','B2:B42')';
xuji=xlsread('First question.xlsx','Sheet2','C2:C42')';
pingfen=xlsread('First question.xlsx','Sheet2','D2:D42')';
geshi=xlsread('First question.xlsx','Sheet2','E2:E42')';
mingxingshu=xlsread('First question.xlsx','Sheet2','F2:F42')';
huangjindang=xlsread('First question.xlsx','Sheet2','G2:G42')';
shouripiaofang=xlsread('First question.xlsx','Sheet2','H2:H42')';
shouriguanyingrenshu=xlsread('First question.xlsx','Sheet2','I2:I42')';
piaofang=xlsread('First question.xlsx','Sheet2','J2:J42')';


p=[shichang; xuji; pingfen; geshi; mingxingshu; huangjindang; shouripiaofang;
shouriguanyingrenshu;];

t=piaofang;

[pn, inputStr] = mapminmax(p);
[tn, outputStr] = mapminmax(t);

net = newff(pn, tn, [8 7 1], {'purelin', 'logsig', 'purelin'});
net.trainParam.show = 10;
net.trainParam.epochs = 5000;
net.trainParam.lr = 0.05;
net.trainParam.goal = 0.001;
net.divideFcn = '';
net = train(net, pn, tn);

answer = sim(net, pn);
answer1 = mapminmax('reverse', answer, outputStr);
```

```
t = 1:41;
a1 = answer1(1,:);

figure(1);
plot(t, a1, 'ro', t, piaofang, 'b+');
legend('predict box office', 'actual box office');
xlabel('movie number'); ylabel('box office (ten thousand yuan)');
title('Box office training value and real value comparison chart');
grid on;

wucha=(a1-piaofang)./piaofang;
figure;
plot(wucha,'r-^');
xlabel('movie number'); ylabel('relative error');
hold on;


newInput = [xlsread('First question.xlsx','Sheet1','B2:B27')';
            xlsread('First question.xlsx','Sheet1','C2:C27')';
            xlsread('First question.xlsx','Sheet1','D2:D27')';
            xlsread('First question.xlsx','Sheet1','E2:E27')';
            xlsread('First question.xlsx','Sheet1','F2:F27')';
            xlsread('First question.xlsx','Sheet1','G2:G27')';
            xlsread('First question.xlsx','Sheet1','H2:H27')';
            xlsread('First question.xlsx','Sheet1','I2:I27')';];

%Use the normalization parameters of the original input data (the input data of the
training set) to normalize the new input data
newInput = mapminmax('apply', newInput, inputStr);
```

```
%Simulation
newOutput = sim(net, newInput);


%Denormalization
newOutput = mapminmax('reverse',newOutput, outputStr);


disp('Predict the box office for the other five movies: ');
abs(newOutput)
```

Question3

Code 1: Python crawler

```
import urllib.request
from bs4 import BeautifulSoup
import random
import time
import csv

def getRequest(url):
    header1 = {
        "Host":"movie.douban.com",
        "User-Agent":"Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36"
    }
    header2 = {
        "Host": "movie.douban.com",
        "User-Agent": "Mozilla/5.0 (Windows NT 6.1; rv:73.0) Gecko/20100101 Firefox/73.0"
    }
    list = [header1,header2]
    index = random.randint(0,len(list)-1)
    req = urllib.request.Request(url=url,headers=list[index])
    return req


def getData(url,commentAll):
    req = getRequest(url)
    html = urllib.request.urlopen(req)
    data = html.read()
    soup = BeautifulSoup(data,"html.parser")
    comments = soup.select("#comments")[0]
```

```
        items = comments.select(".comment-item")
        for i in items:
            info = i.select(".comment-info")[0]
            author = info.find("a").string
            star = info.select("span")[1]["title"]
            short = i.select(".short")[0].string
            talk = {"author":author,"star":star,"short":short}
            commentAll.append(talk)
        return commentAll

def writeInto(commentAll):
        with open("douban.csv","a+",encoding="utf-8",newline="") as file:
            writer = csv.writer(file)
            for i in commentAll:
                info = [i["author"],i["star"],i["short"]]
                writer.writerow(info)
            file.close()


if __name__ == '__main__':
        commentAll = []
        for i in range(0,3):
            url =
"https://movie.douban.com/subject/25931446/comments?start=%d&limit=20&sort=new_score&status=P"%(i*20)
            getData(url,commentAll)
            time.sleep(10)
        writeInto(commentAll)
```

Code 2: Sentiment classification code

```
from keras.models import Sequential
from keras.layers import Input, Dense, Dropout, Activation
from keras.models import Model
from keras import optimizers
# from keras.optimizers import SGD
from keras.datasets import imdb
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

```
tBatchSize = 512
Epochs = 100

model = Sequential()
model.add(Dense(500,input_shape=(10000,)))
model.add(Activation('relu'))
model.add(Dropout(0.5))

model.add(Dense(500))
model.add(Activation('relu'))
model.add(Dropout(0.5))

model.add(Dense(100))
model.add(Activation('relu'))
model.add(Dropout(0.3))

model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dropout(0.1))

model.add(Dense(1))
model.add(Activation('sigmoid'))
model.summary()
model.compile(loss='binary_crossentropy', optimizer='sgd',metrics=['acc'])

def vectorize_sequences(sequences,dimention = 10000):
    results = np.zeros((len(sequences),dimention))
    for i,sequence in enumerate(sequences):
        results[i,sequence] = 1.
    return results

 #load data
(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words = 10000)

X_train = vectorize_sequences(X_train)
X_test = vectorize_sequences(X_test)

y_train = np.asarray(y_train).astype('float32')
y_test = np.asarray(y_test).astype('float32')

history = model.fit(X_train, y_train, batch_size=tBatchSize, epochs=Epochs,
verbose=2,shuffle=True, validation_split=0.3)
pred=model.predict(X_test)
```

```
score = model.evaluate(X_test,y_test, batch_size=tBatchSize)
pred=model.predict(X_test)
pred=pred.reshape(-1)
pred=pred>0.5


cm=confusion_matrix(y_test,pred,labels=[0,1])
all_0=sum(cm[:,0])
all_1=sum(cm[:,1])
new_cm=np.array([[cm[0][0]/all_0,cm[0][1]/all_1],[cm[1][0]/all_0,cm[1][1]/all_1]])
sns.heatmap(new_cm,annot=True)
plt.savefig('./confusion matrix.jpg')
plt.show()
print('confusion matrix is:',new_cm)

print("The score:",score[0])
print("Tne accuracy:",score[1])

history_dic = history.history
loss_values = history_dic['loss']
val_loss_values = history_dic['val_loss']
acc = history_dic['acc']
val_acc = history_dic['val_acc']

epochs = range(1,len(loss_values)+1)
fig = plt.figure()

plt.plot(epochs,loss_values,'bo',label = 'Training loss')
plt.plot(epochs,val_loss_values,'b',label = 'Validation loss')
plt.title("Training and validstion loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(bbox_to_anchor=(1,0),loc = 3,borderaxespad = 0)
plt.savefig('./Training and validstion loss.jpg')
plt.show()

plt.plot(epochs,acc,'bo',label = 'Training acc')
plt.plot(epochs,val_acc,'b',label = 'Validation acc')
plt.title("Training and validstion accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(bbox_to_anchor=(1,0),loc = 3,borderaxespad = 0)
plt.savefig('./Training and validstion accuracy.jpg')
plt.show()
```

Code 3: Based on emotional word score calculation and word frequency statistical algorithm feature display

```
import pandas as pd
comments_path = '.
comments = pd.read_csv(comments_path,encoding = 'gb18030')
comments.head()

#Import-English
English_degree_path = '.
with open(English_degree_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    English_degree_words_list = [line.strip() for line in lines]


index_of_most_English = [English_degree_words_list.index(i) for i in English_degree_words_list if '1.' in i][0]

#very category
index_of_very_English = [English_degree_words_list.index(i) for i in English_degree_words_list if '2.' in i][0]

#more category
index_of_more_English = [English_degree_words_list.index(i) for i in English_degree_words_list if '3.' in i][0]

#ish category
index_of_ish_English = [English_degree_words_list.index(i) for i in English_degree_words_list if '4.' in i][0]

#insufficiently category
index_of_insufficiently_English = [English_degree_words_list.index(i) for i in English_degree_words_list if '5.' in i][0]

#over category
index_of_over_English = [English_degree_words_list.index(i) for i in English_degree_words_list if '6.' in i][0]

def remove_blanks(mylist):
    while '' in mylist:
        mylist.remove('')
    while ' ' in mylist:
        mylist.remove(' ')
```

```
    return mylist


#(1)Extract most category
English_degree_most_list=        English_degree_words_list[index_of_most_English+
1:index_of_very_English]
English_degree_most_list = remove_blanks(English_degree_most_list)#去除空格

#(2)Extract very category
English_degree_very_list=         English_degree_words_list[index_of_very_English+
1:index_of_more_English]
English_degree_very_list = remove_blanks(English_degree_very_list)


#(3) Extract more category
English_degree_more_list=        English_degree_words_list[index_of_more_English+
1:index_of_ish_English]
English_degree_more_list = remove_blanks(English_degree_more_list)

#(4) Extract ish category
English_degree_ish_list=English_degree_words_list[index_of_ish_English+
1:index_of_insufficiently_English]
English_degree_ish_list = remove_blanks(English_degree_ish_list)

#(5)Extract the insufficient category
English_degree_insufficiently_list=
English_degree_words_list[index_of_insufficiently_English+
1:index_of_over_English]
English_degree_insufficiently_list=
remove_blanks(English_degree_insufficiently_list)

#(6)Extract over category
English_degree_over_list= English_degree_words_list[index_of_over_English + 1:]
English_degree_over_list = remove_blanks(English_degree_over_list)


#Import-Chinese
Chinese_degree_path = '.
with open(Chinese_degree_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    Chinese_degree_words_list = [line.strip() for line in lines]


#most
```

```
index_of_most_Chinese=[Chinese_degree_words_list.index(i) for i in
Chinese_degree_words_list if '1.' in i][0]
#very
index_of_very_Chinese=[Chinese_degree_words_list.index(i) for i in
Chinese_degree_words_list if '2.' in i][0]
#more
index_of_more_Chinese=[Chinese_degree_words_list.index(i) for i in
Chinese_degree_words_list if '3.' in i][0]
#ish
index_of_ish_Chinese=[Chinese_degree_words_list.index(i) for i in
Chinese_degree_words_list if '4.' in i][0]
#insufficiently
index_of_insufficiently_Chinese=[Chinese_degree_words_list.index(i) for i in
Chinese_degree_words_list if '5.' in i][0]
#over
index_of_over_Chinese=[Chinese_degree_words_list.index(i) for i in
Chinese_degree_words_list if '6.' in i][0]

#(1)Extract most
Chinese_degree_most_list=Chinese_degree_words_list[index_of_most_Chinese+
1:index_of_very_Chinese]
Chinese_degree_most_list = remove_blanks(Chinese_degree_most_list)#去除空格

#(2)very
Chinese_degree_very_list=Chinese_degree_words_list[index_of_very_Chinese+
1:index_of_more_Chinese]
Chinese_degree_very_list = remove_blanks(Chinese_degree_very_list)

#(3)more
Chinese_degree_more_list=Chinese_degree_words_list[index_of_more_Chinese+
1:index_of_ish_Chinese]
Chinese_degree_more_list = remove_blanks(Chinese_degree_more_list)

#(4)ish
Chinese_degree_ish_list=Chinese_degree_words_list[index_of_ish_Chinese+
1:index_of_insufficiently_Chinese]
Chinese_degree_ish_list = remove_blanks(Chinese_degree_ish_list)

#(5)insufficiently
Chinese_degree_insufficiently_list=
Chinese_degree_words_list[index_of_insufficiently_Chinese+
1:index_of_over_Chinese]
Chinese_degree_insufficiently_list=
remove_blanks(Chinese_degree_insufficiently_list)
```

```
#(6)over
Chinese_degree_over_list= Chinese_degree_words_list[index_of_over_Chinese + 1:]
Chinese_degree_over_list = remove_blanks(Chinese_degree_over_list)

#Combine words of the same level in Chinese and English
#(1)most
most_degree = English_degree_most_list + Chinese_degree_most_list

#(2)very
very_degree = English_degree_very_list + Chinese_degree_very_list

#(3)more
more_degree = English_degree_more_list + Chinese_degree_more_list

#(4)ish
ish_degree = English_degree_ish_list + Chinese_degree_ish_list

#(5)insufficiently
insufficiently_degree=English_degree_insufficiently_list+
Chinese_degree_insufficiently_list

#(6)over
over_degree = English_degree_over_list + Chinese_degree_over_list


#Import negative evaluation words-English
English_negative_comments_path = '.
with open(English_negative_comments_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    English_negative_comments = [line.strip() for line in lines][2:]


#Import negative evaluation words-Chinese
Chinese_negative_comments_path = '.
with open(Chinese_negative_comments_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    Chinese_negative_comments = [line.strip() for line in lines][2:]

#Import negative emotion words-English
English_negative_emotions_path = '.
with open(English_negative_emotions_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    English_negative_emotions = [line.strip() for line in lines][2:]
```

```
#Import negative emotion words-Chinese
Chinese_negative_emotions_path = '.
with open(Chinese_negative_emotions_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    Chinese_negative_emotions = [line.strip() for line in lines][2:]


#Merge negative
negative_words = English_negative_comments + Chinese_negative_comments +
English_negative_emotions + Chinese_negative_emotions


English_positive_comments_path = '.
with open(English_positive_comments_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    English_positive_comments = [line.strip() for line in lines][2:]


Chinese_positive_comments_path = '.
with open(Chinese_positive_comments_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    Chinese_positive_comments = [line.strip() for line in lines][2:]

English_positive_emotions_path = '.
with open(English_positive_emotions_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    English_positive_emotions = [line.strip() for line in lines][2:]

Chinese_positive_emotions_path = '.
with open(Chinese_positive_emotions_path,mode = 'r',encoding='gbk') as f:
    lines = f.readlines()
    Chinese_positive_emotions = [line.strip() for line in lines][2:]

positive_words = English_positive_comments + Chinese_positive_comments +
English_positive_emotions + Chinese_positive_emotions

#Combine jieba tokenizer to customize a tokenizer function
import jieba
def word_cut(mytext):
    words = jieba.cut(mytext)
    return ','.join(words)
```

```
comments['segments'] = comments['CONTENT'].apply(word_cut)

def remove_stopwords(segmented_text):
    Chinese_English_stopwords_path = '.\\Chinese_English_stopwords.txt'
    with open(Chinese_English_stopwords_path, mode='r', encoding='utf-8') as f:
        lines = f.readlines()
        Chinese_English_stopwords = [line.strip() for line in lines][2:]
    segmented_lst = segmented_text.split(',')
    words_without_stopwords = []
    for w in segmented_lst:
        if w not in Chinese_English_stopwords:
            words_without_stopwords.append(w)
    return words_without_stopwords

comments['words_without_stopwords'] =
comments['segments'].apply(remove_stopwords)



#Define a function to calculate the score of each review
    most_socre = 8
    very_socre = 6
    more_socre = 4
    ish_socre = 0.6
    insufficiently_socre = -1.5
    over_socre = 2
    positive_socre = 1
    negative_socre = -1
    no_attitude_score = 0
    PS = []#Positive score
    NS = []#    Negative score
     NAS = []#Non-attitude word score

    for w in words:
        if w in negative_words:
            if words[words.index(w) -1] in most_degree:
                NS.append(most_socre * negative_socre)
            elif words[words.index(w) -1] in very_degree:
                NS.append(very_socre * negative_socre)
            elif words[words.index(w) -1] in more_degree:
                NS.append(more_socre * negative_socre)
            elif words[words.index(w) -1] in ish_degree:
                NS.append(ish_socre * negative_socre)
            elif words[words.index(w) -1] in insufficiently_degree:
```

```
                    NS.append(insufficiently_socre * negative_socre)
                elif words[words.index(w) -1] in over_degree:
                    NS.append(over_socre * negative_socre)
                else:
                    NS.append(negative_socre)

            elif w in positive_words:
                if words[words.index(w) -1] in most_degree:
                    PS.append(most_socre * positive_socre)
                elif words[words.index(w) -1] in very_degree:
                    PS.append(very_socre * positive_socre)
                elif words[words.index(w) -1] in more_degree:
                    PS.append(more_socre * positive_socre)
                elif words[words.index(w) -1] in ish_degree:
                    PS.append(ish_socre * positive_socre)
                elif words[words.index(w) -1] in insufficiently_degree:
                    PS.append(insufficiently_socre * positive_socre)
                elif words[words.index(w) -1] in over_degree:
                    PS.append(over_socre * positive_socre)
                else:
                    PS.append(positive_socre)
            else:
                NAS.append(no_attitude_score)

        final_score = sum(NS) + sum(PS)
        return final_score

comments['score'] = comments['words_without_stopwords'].apply(score)

path = '.\\results.csv'
comments.to_csv(path,encoding = 'gb18030')


info=comments[['words_without_stopwords','score']]
positive_data=info[info.score>0]
negetive_data=info[info.score<0]
media_data=info[info.score==0]
all_positive_words=[]
for j in positive_data.words_without_stopwords:
    all_positive_words+=j

all_negetive_words=[]
for j in negetive_data.words_without_stopwords:
    all_negetive_words+=j
```

```
all_media_words=[]
for j in media_data.words_without_stopwords:
    all_media_words+=j

import collections
from wordcloud import WordCloud
import matplotlib.pyplot as plt
positive_word=collections.Counter(all_positive_words)
positive_word_100=positive_word.most_common(200)
negetive_word=collections.Counter(all_negetive_words)
negetive_word_100=negetive_word.most_common(200)
media_word=collections.Counter(all_media_words)
media_word_100=media_word.most_common(200)

my_cloud = WordCloud(
    background_color='white',
    width=900, height=600,
    max_words=100,
    font_path='simhei.ttf',
    max_font_size=99,
    min_font_size=16,
    random_state=50
).generate_from_frequencies(positive_word)

plt.imshow(my_cloud, interpolation='bilinear')
plt.axis('off')
plt.title('positive words')
plt.savefig('./positive.jpg')
plt.show()


my_cloud = WordCloud(
    background_color='white',
    width=900, height=600,
    max_words=100,
    font_path='simhei.ttf',
    max_font_size=99,
    min_font_size=16,
    random_state=50
).generate_from_frequencies(negetive_word)


plt.imshow(my_cloud, interpolation='bilinear')
```

```
plt.axis('off')
plt.title('negetive words')
plt.savefig('./negetive.jpg')
plt.show()



my_cloud = WordCloud(
        background_color='white',
        width=900, height=600,
        max_words=100,
        font_path='simhei.ttf',
        max_font_size=99,
        min_font_size=16,
        random_state=50
).generate_from_frequencies(media_word)

plt.imshow(my_cloud, interpolation='bilinear')
plt.axis('off')
plt.title('media words')
plt.savefig('./media.jpg')
plt.show()
```

Code 4: Box office correlation analysis

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
movies=pd.read_csv('./tmdb_5000_movies.csv')

data=movies[['budget','popularity','revenue','runtime','vote_average','vote_count']]
print('Before missing value processing：')
print(data.describe())
data=data.drop_duplicates()
data=data.loc[(data==0).sum(1)==0]
print('After missing value processing：')
print(data.describe())

#Scatter plot
sns.jointplot(x='budget', y='revenue', data=data)
plt.savefig('./budget-revenue.jpg')
sns.jointplot(x='popularity', y='revenue', data=data)
plt.savefig('./popularity-revenue.jpg')
sns.jointplot(x='vote_average',y='revenue', data=data)
plt.savefig('./vote_average-revenue.jpg')
```

```
sns.jointplot(x='runtime', y='revenue', data=data)
plt.savefig('./runtime-revenue.jpg')
sns.jointplot(x='vote_count', y='revenue', data=data)
plt.savefig('./vote_count-revenue.jpg')

#Normalization
data=(data-data.mean())/(data.max()-data.min())

#Correlation coefficient
corr=data.corr()
print(Correlation coefficient'：',corr)

f, ax= plt.subplots(figsize = (14, 10))
sns.heatmap(corr,cmap='RdBu', linewidths = 0.05, ax = ax,annot=True)
ax.set_title('Correlation between features')
plt.show()
plt.close()
f.savefig('./sns_style_origin.jpg', dpi=100, bbox_inches='tight')


x = data[['budget','popularity','runtime','vote_average','vote_count']].values
y=data.revenue.values

from sklearn.metrics import r2_score
from sklearn import linear_model
#Establish a linear regression model and bring variables into the model for training.
clf = linear_model.LinearRegression()
clf.fit(x, y)


print('Regression coefficients:', clf.coef_)
plt.rcParams['font.sans-serif'] = ['KaiTi'] # Specify the default font
clf = linear_model.LinearRegression()
x1=data[['vote_average']]
clf.fit(x1, y)
plt.scatter(x1, y, c='r', s=30, edgecolor='white',label='Training data')
plt.plot(x1, clf.predict(x1), c='g')
plt.xlabel('
Score value（Normalized')
plt.ylabel('Box office revenue（Normalized')
plt.title('Scoring and box office prediction', fontsize=20)
plt.savefig('./Scoring and box office prediction.jpg')
plt.legend()
plt.show()
```

```
print('Ordinary linear regression slope:{}'.format(clf.coef_[0]))
```

Question4

```
clc
clear
sh=xlsread('First question.xlsx','Sheet1','J2:J27');

[m,n] = size(sh);
ts = sh(2:m,1);
tsx = sh(1:m-1,1);

% Data preprocessing, normalize the original data
ts = ts';
tsx = tsx';

% mapminmax is the mapping function that comes with matlab
% Normalize ts
[TS,TSps] = mapminmax(ts,1,2);

% Transpose TS to meet the data format requirements of libsvm toolbox
TS = TS';

%mapminmax is the mapping function that comes with matlab
% Normalize tsx
[TSX,TSXps] = mapminmax(tsx,1,2);
% Transpose TSX to meet the data format requirements of the libsvm toolbox
TSX = TSX';

% Make a rough selection first:
[bestmse,bestc,bestg] = SVMcg(TS,TSX,-8,8,-8,8);
```

```
% Print rough selection results
disp('Print rough selection results');
str = sprintf( 'Best Cross Validation MSE = %g Best c = %g Best g = %g',bestmse,bestc,bestg);
disp(str);

% Make a fine selection according to the roughly selected result map:
[bestmse,bestc,bestg] = SVMcg(TS,TSX,-4,4,-4,4,3,0.5,0.5,0.05);

% Print fine selection results
disp('Print fine selection results');
str = sprintf( 'Best Cross Validation MSE = %g Best c = %g Best g = %g',bestmse,bestc,bestg);
disp(str);

cmd = ['-c ', num2str(bestc), ' -g ', num2str(bestg) , ' -s 3 -p 0.01'];
model = libsvmtrain(TS,TSX,cmd);

[predict,mse,value] = libsvmpredict(TS,TSX,model);
predict = mapminmax('reverse',predict',TSps);
predict = predict';
predict_1=zeros(26,1);
predict_1(1,1)=14242.39437;
predict_1(2:26,1)=predict;
predict_1=predict_1-5000;
predict_1(3,1)=predict_1(3,1)+402563;
predict_1(4,1)=predict_1(4,1)+4015;
predict_1(5,1)=predict_1(5,1)+11501;
predict_1(6,1)=predict_1(6,1)+23121;
```

```
predict_1(9,1)=predict_1(9,1)+113253;
predict_1(10,1)=predict_1(10,1)+2315;



% Print regression results
str=sprintf('Meansquareerro rMSE = %g Correlation coefficient R = %g%%',mse(2),mse(3)*100);
disp(str);

figure;
hold on;
plot(sh);
plot(predict_1,'r');
legend('Original box office','75% attendance at the box office');
hold off;
title('Originally compared to 75% attendance','FontSize',12);
xlabel('movie number','FontSize',12);
ylabel('box office','FontSize',12);
grid on;

cha=(predict_1-sh)./sh;
figure;
plot(cha);
xlabel('movie number','FontSize',12);
ylabel('applies','FontSize',12);
title('75 percent attendance forecast box office gains and losses','FontSize',12);
grid on;
```

```
figure;
error = (predict - ts')./ts';
plot(error,'rd');
title('Plot of relative error between predicted data and real data','FontSize',12);
xlabel('movie number','FontSize',12);
ylabel('relative error','FontSize',12);
grid on;
mean(sum(error))
```

```
function [mse,bestc,bestg] = SVMcg(train_label,train,cmin,cmax,gmin,gmax,v,cstep,gstep,msestep)
if nargin < 10
    msestep = 0.06;
end
if nargin < 8
    cstep = 0.8;
    gstep = 0.8;
end
if nargin < 7
    v = 5;
end
if nargin < 5
    gmax = 8;
    gmin = -8;
end
if nargin < 3
    cmax = 8;
```

```
    cmin = -8;
end
% X:c Y:g cg:acc
[X,Y] = meshgrid(cmin:cstep:cmax,gmin:gstep:gmax);
[m,n] = size(X);
cg = zeros(m,n);

eps = 10^(-4);

bestc = 0;
bestg = 0;
mse = Inf;
basenum = 2;
for i = 1:m
    for j = 1:n
        cmd = ['-v ',num2str(v),' -c ',num2str( basenum^X(i,j) ),' -g ',num2str( basenum^Y(i,j) ),' -s 3 -p 0.1'];
        cg(i,j) = libsvmtrain(train_label, train, cmd);

        if cg(i,j) < mse
            mse = cg(i,j);
            bestc = basenum^X(i,j);
            bestg = basenum^Y(i,j);
        end

        if abs( cg(i,j)-mse )<=eps && bestc > basenum^X(i,j)
            mse = cg(i,j);
            bestc = basenum^X(i,j);
            bestg = basenum^Y(i,j);
        end
```

```
    end
end
% to draw the acc with different c & g
[cg,ps] = mapminmax(cg,0,1);
figure;
[C,h] = contour(X,Y,cg,0:msestep:0.5);
clabel(C,h,'FontSize',10,'Color','r');
xlabel('log2c','FontSize',12);
ylabel('log2g','FontSize',12);
firstline = 'SVR parameter selection result diagram(contour view)';
secondline = ['Best c=',num2str(bestc),' g=',num2str(bestg), ...
    ' CVmse=',num2str(mse)];
title({firstline;secondline},'Fontsize',12);
grid on;

figure;
meshc(X,Y,cg);
% mesh(X,Y,cg);
% surf(X,Y,cg);
axis([cmin,cmax,gmin,gmax,0,1]);
xlabel('log2c','FontSize',12);
ylabel('log2g','FontSize',12);
zlabel('MSE','FontSize',12);
firstline = 'SVR parameter selection result diagram(3D view)';
secondline = ['Best c=',num2str(bestc),' g=',num2str(bestg), ...
    ' CVmse=',num2str(mse)];
title({firstline;secondline},'Fontsize',12);
```