Problem Chosen
D

November 15, 2021
ShuWei Cup
Summary Sheet

Team Control Number
202111056924

Summary

As an important indicator of a movie's success, it is important to predict the box office before its release. However, the box office is affected by many factors and there is a lack of particularly well-developed models to accomplish this task. Therefore, it is significant to build a model that can predict the box office of a movie and deal with "network navy" and unexpected events at the same time.

As to question 1, we first performed feature quantification and feature engineering dimensionality reduction on the data we collected, after which we divided the movies into four categories based on box office by K-means clustering algorithm, and finally input the filtered features to construct a random forest model to achieve classification of movies. The accuracy of the finally constructed random forest model was 0.813, which proved the effectiveness of its classification.

As to question 2, we decided to build a BP neural network using the PyTorch framework. We selected 5 input layers using five valid features from the feature engineering analysis of the first question as input, 11 hidden layers based on empirical formulas and 1 output layer based on the task of predicting box office. We chose the Adam optimization algorithm to optimize our neural network and briefly explained its principle. We randomly selected non-overlapping training and test sets to validate our model and found that the prediction results were perfectly acceptable. Further, we performed box office prediction statistics for each category of the first question classification.

As to question 3, we proved that there is a greater correlation between online public opinion and box office by constructing a nonlinear equation. We established a corresponding equation to reflect the degree of its influence. For the identification of "network navy", we construct an NLP model and find that "network navy" mainly have a greater degree of correlation with actors by analyzing word frequency information. The accuracy of the model is 0.848, which proved that it is feasible for recognizing "network navy" in movie reviews.

As to question 4, we first analyzed the negative impact of the attendance restriction policy on box office under the COVID-19 epidemic. We also analyzed the problems that might be encountered in constructing the model and revised the BP neural network model in Problem 2. After the reconfiguration, we discussed the results of several models and used 9 movies during the epidemic to verify its feasibility. For the movies under different attendance restriction policies, we predicted their box office after the epidemic stabilized and compared them with the actual box office, and found a significant difference between the predicted and actual values for the cases where the attendance restriction was below 30%.

Key word: K-means, Random Forest, BPNN, Nonlinear regression, NLP

Content

1. Introduction

1.1 Background

With the increase of China's cultural consumption demand, the movie industry is booming, and the movie box office is also increasing year by year; as an important index to measure the success of a movie, the movie box office is influenced by various factors such as the quality of the movie itself, the director, the actors, and the screening time; however, due to the relatively little research on the prediction of movie box office in China, there are not a few movies that end badly after the premiere; at the same time, nowadays, the network However, there are few studies on box office prediction in China, and at the same time, the impact of comments and unexpected events on box office is also huge. Therefore, it is especially important to build a model that can predict movie box office and cope with the impact of online network navy and unexpected events.

1.2 Restatement of the Problem

1. Filter the feature labels of movies, build clustering and classification models to classify movies respectively, and verify the validity of the classification.
2. Based on the classification results of Task 1, build a prediction model to predict the box office of each category and the total box office of all movies separately.
3. Collect public opinion evaluation, analyze the correlation between online public opinion and box office as well as the degree of influence on box office, and establish a model to identify online network navy.
4. Analyze the impact of different attendance rates (30%, 50%, 75%) on movie box office considering the case of epidemic emergencies and predict the corresponding results.

1.3 Our Work

For the first question, we first collected numerous features such as movie box office as well as directors and actors, then performed feature quantification and feature engineering processing, then used K-means clustering, and finally constructed a random forest model to classify the movies and verified the effectiveness of the classification.

For the second question, we constructed a BPNN neural network for predicting the box office of a movie, trained and predicted based on the classification results of the first question.

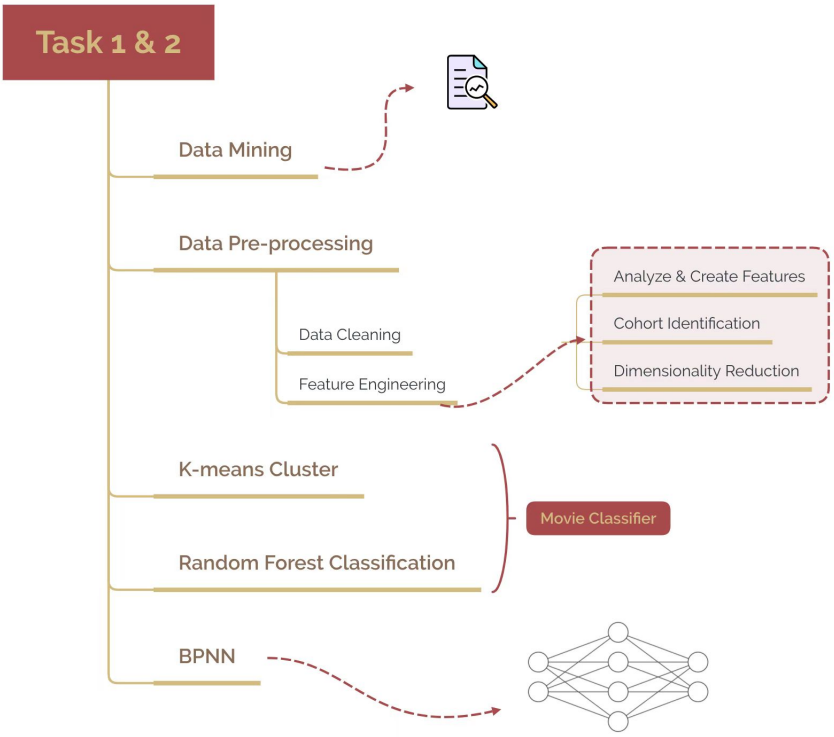

Figure 1 Box office classification and forecasting flow chart

For the third question, we first prove the correlation between public opinion and movie box office using a nonlinear regression equation, and then construct an NLP model to identify online Network navy.

For the fourth question, we added the effect of the epidemic to the neural network in the second question, and finally came up with the effect of different attendance rates on movie box office prediction.

2. Assumptions and Justifications

To simplify the problem, we make the following basic assumptions, each of which is properly justified.

- **Assumption 1**: Assuming that all the movie data we collected are authentic and credible.
  Justification: Since our movie data comes from professional data platforms such as Douban and Maoyan, we can assume that the data is true and reliable.

- **Assumption 2**: Assume that in the same movie, the director and the lead actor have the same weight of influence on the box office.
  Justification: Since the influence of the director and the lead actor on the box office shows a positive correlation, we can simplify to assume that their joint influence is the same in

the same movie.

- **Assumption 3**: Assuming that the films do not interfere with each other.
  Justification: Although in real life two movies with colliding schedules may have an impact on their respective box office, since we mainly analyze the impact of the movie's own characteristics and external contingencies on the movie's box office in this time. Therefore we can directly ignore the impact generated between movies.

- **Assumption 4**: Ignore the effect of inflation on movie box office in different years.
  Justification: Since movies are released in different years, there is bound to be some inflation, but for the model of this question, considering inflation would make the model bloated and we would not have access to accurate inflation information, so we choose to ignore this influence.

3. Notations

Table 1 Symbols and its Description

Symbols	Description
Dir_i	The influence of the i-th director
LR_i	The influence of the i-th actor
Cor_i	The influence of the ith company
E	Overall error in clustering algorithms
g_t	Gradient
s_t	state vector
v_t	leaky average
x_1	Normalized to [-1,1] Douban's rating
x_2	Normalized to [-1,1] Maoyan's rating
Com	Measuring public opinion for better or worse

4. Model Preparations

Since the data of the movies are not directly given in the title, we first need to collect data of relevant movies from platforms such as Douban and Maoyan; our team collected data of 140 movie box office, each movie has 16 feature tags, and the meanings of the specific tags are shown

in the following table.

Table 2 Movie tag meaning explained

Tags	Meaning
id	Serial number
name	Movie Title
director	Director Name
actor	Actor Name
time	Film duration(in minutes)
type	Film genre
format	Film Format
sequel	Whether the movie is a sequel
display_time	Movie release time
production	Publishing company of the film
publisher	Distribution company of the film
douban	Douban rating of the movie
maoyan	Maoyan rating of the movie
ticket_price	Average ticket price of movies
first_week_box_office	The first week of the film's box office
box_offcie	Total box office of the film

After that, we will use the relevant data we have collected to classify and predict the movies.

5. Task 1: Movie Classifier

5.1 Feature metrics

Before classifying and predicting movies, we observed that the data had directors, actors, companies and release dates, among others, in a non-numerical format that did not facilitate our subsequent processing, so we needed to perform feature metrics to quantify these important features.

Firstly, the four indicators of directors, actors, publishing companies and distribution companies. Since each of these four indicators contains Chinese names and is more scattered, we cannot translate them directly, so we can calculate the influence corresponding to directors, actors and companies and quantify them with influence values.

Since our whole classification model needs to be classified based on box office, we can take box office as a significant indicator of influence, and in addition, with reference to the number of

appearances, social influence and other factors, we can get the formula for influence as follows:

Calculating the impact of directors:

$$Dir_i = \sum_{j=1}^{m} b_j / m \tag{1}$$

Calculating the impact of actors:

$$LR_i = \sum_{j=1}^{m} b_j / n_j m \tag{2}$$

Calculating the impact of company:

$$Cor_i = \sum_{j=1}^{m} b_j / n_j m \tag{3}$$

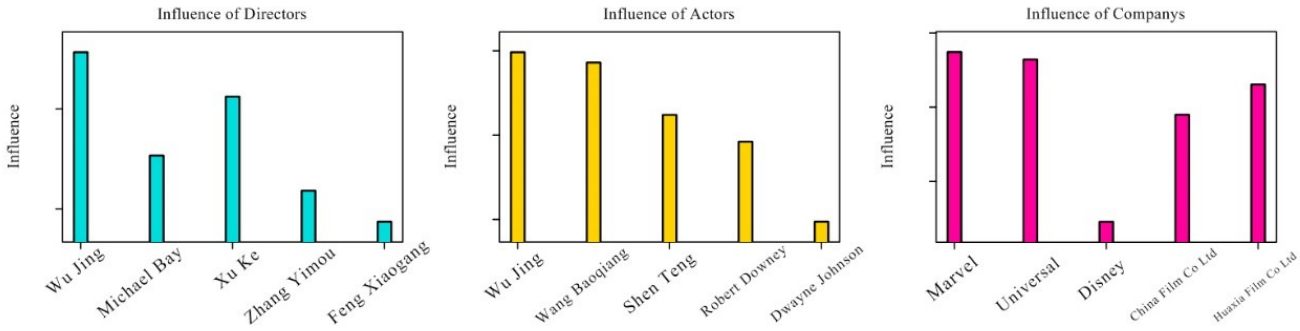

Figure 2 Director, actor, company influence

Nextly, the release date of the movie. If we simply look at the date we cannot observe the relationship between it and the box office; however, it is obvious that the box office during the holidays, especially during the Chinese New Year, is much higher than the box office during normal times, so we have reviewed the following date characterization table.[1]

Table 3 Release date characterization table

Display	Value
Spring Festival	1
Labor Day	2
Summer Vacation	3
National Day	4
Others	5

With the above feature metric, we have roughly transformed the features we need into values that we can easily process; parsing comes to the point where we need to further process our features through feature engineering.

5.2 Feature engineering

We observe that for the classification of movies, we already have 14 features (excluding the serial number and movie name), too many features will give us large errors in the classification results, so we need to perform feature dimensionality reduction using feature engineering and delete the less important features.

First of all, the influence characteristics of director, actor, publishing company and distribution company, because the influence of these four characteristics on the box office is almost all positive, so we can combine these four labels into "famous_degree", and because the difference between the influence values is large, so we need to carry out the normalization operation, and all the values are grouped into [0, 10]; finally we get the formula of "famous_degree" as follows.

$$famous\_degree = \frac{(influence_i - influence_{min})}{(influence_{max} - influence_{min})} + 0.01 \tag{4}$$

Next is the type of movie, which is divided into nine types: action, comedy, drama, fantasy, science fiction, animation, romance, disaster and war, but we found that by checking movie-related information. The four types of movies, namely fantasy, science fiction, disaster and war, are mostly special effects blockbusters, so we can directly classify them into one category, and finally our type characterization table is shown below.

Table 4 Film genre characterization table

Type	Value
Action	1
Comedy	2
Drama	3
Fantasy	4
Science Fiction	4
Animation	5
Romance	6
Disaster	4
War	4

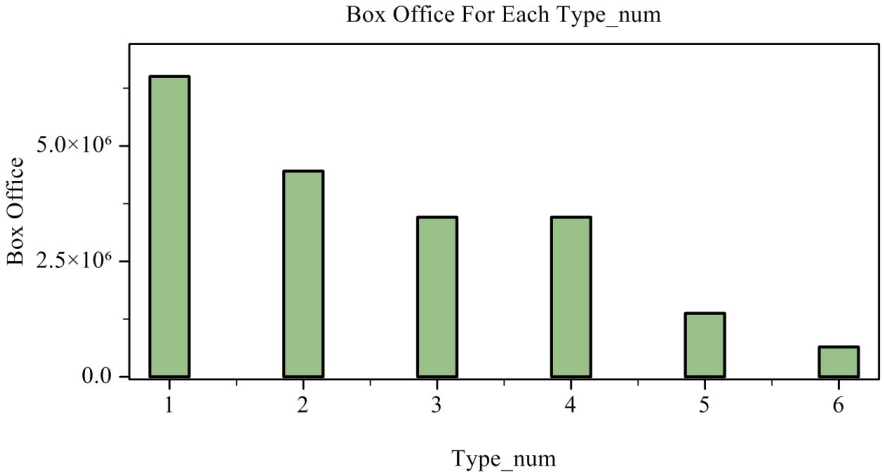

Figure 3 Box Office for Each Type_num

Finally, we have the feature of whether the movie is a sequel or not. Since the box office and reputation of the child movie are largely influenced by the parent movie, we also have to extract features for the label of whether the movie is a sequel or not.

Table 5 Sequel to Whether

Sequel	Whether
Yes	1
No	0

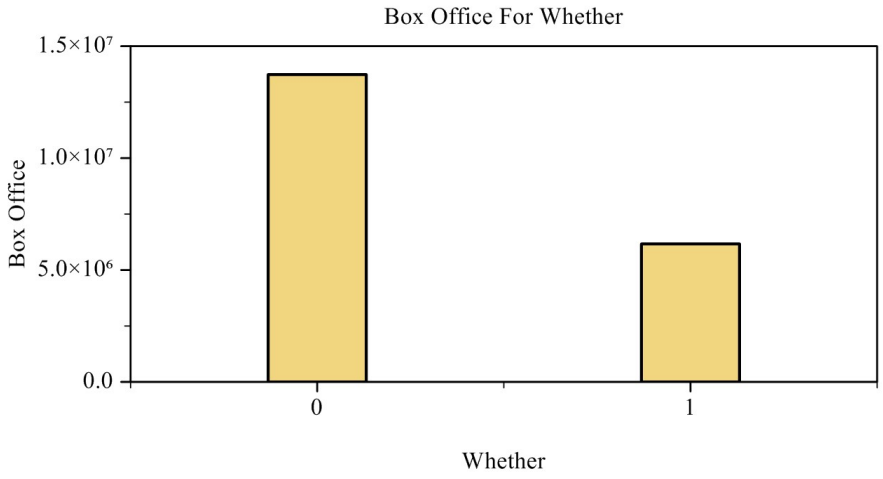

Figure 4 Box Office for Whether

After constructing the feature engineering tags, we also need to check the correlation between these tags and the box office, and here we visualize the correlation by drawing a heat map.

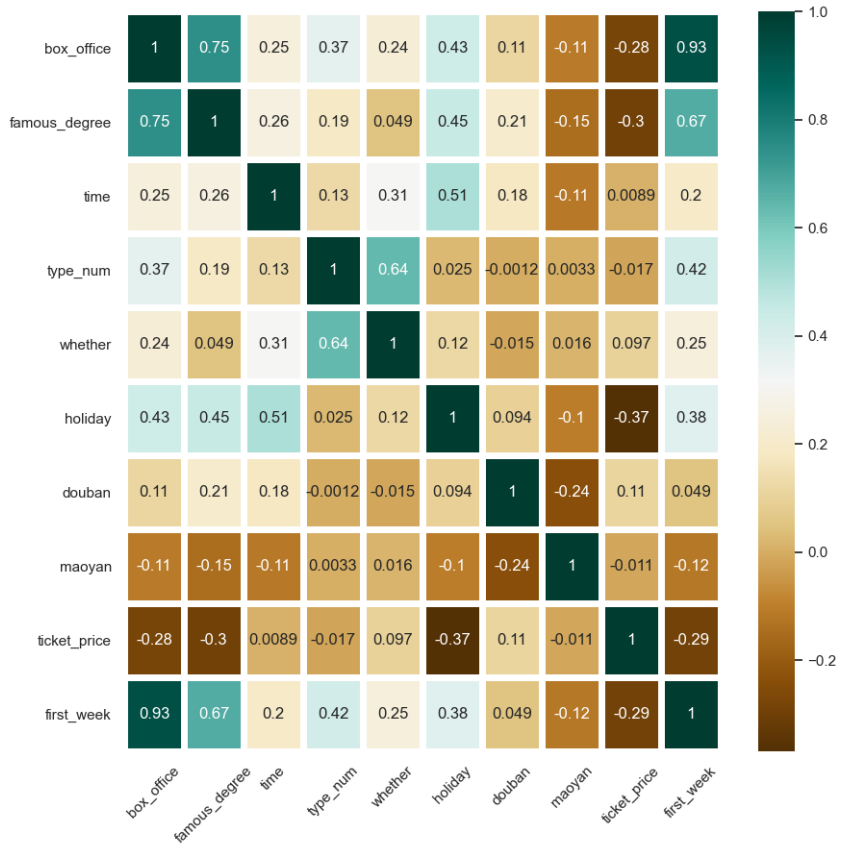

Figure 5 Correlation Heat Map

It can be seen that the four labels of famous_degree, type_num, other and holiday have a higher correlation degree with box office, and finally we also leave these four features as the classification features of movies.

The correlation degree of the ticket_price tag exceeds 0.9, and the value is too large to cause a single variable to affect the global situation, and the correlation degree of the remaining features is also low, so they can be discarded.

5.3 K-means cluster analysis

K-means algorithm (i.e., K-means clustering algorithm) is a well-known divisional clustering partitioning method. The basic idea of the division method is that given a data set with N tuples or records, the splitting method will construct K groupings, each of which represents a cluster, $K < N$,and these K groupings satisfy the following conditions:

1. Each grouping contains at least one data record.
2. Each data record belongs to one and only one group.

For a given K, the algorithm first gives an initial grouping method, and later changes the grouping by iterative iterations so that the records in the same grouping are closer and the records in different groupings are further apart. The general steps of the K-means algorithm are as follows:

1. Choose any k objects from the n data objects as the initial clustering centers.
2. Based on the mean value of each clustered object (central object), the distance of each object from these central objects is calculated and the corresponding objects are re-classified according to the minimum distance.
3. The mean value (central object) of each cluster is recalculated until the cluster center no longer changes. This division minimizes the following equation.

$$E = \sum_{j=1}^{k} \sum_{x_i \in \omega_i} \left\| x_i - m_j \right\|^2 \tag{5}$$

Finally, we divided the movies into four categories according to their box office by K-means clustering, and replaced them with A, B, C and D. Category A represents the group of movies with the highest box office, while category D is the lowest category.

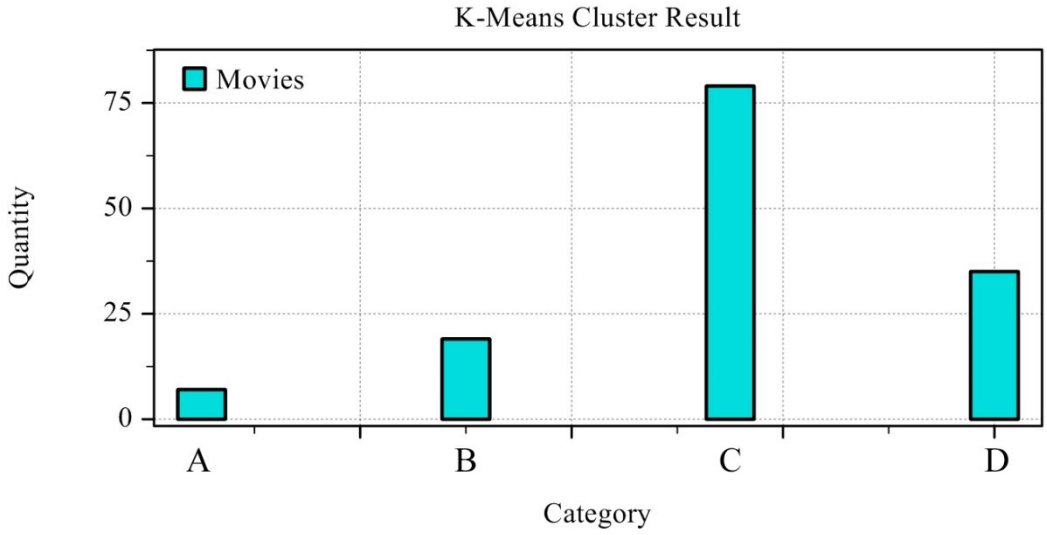

Figure 6 K-means Cluster Result

5.4 Random Forest Movie Classification Model

Random forest is a product of integrated learning idea, which integrates multiple decision trees to integrate into a forest to solve the limitations and inherent defects due to individual decision trees in order to obtain more reasonable and accurate prediction results.

Random forest is a kind of sampling method with put-back, and the sampling strategy is simple random sampling. The principle is straightforward, just put multiple base models together

and average them at the end. Here the decision tree can be taken as the base model with the following formula.

$$f(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x) \tag{6}$$

According to what is described in 1.2, we input four features, "famous_degree", "type_num", "whether", other, and holiday, into the random forest and construct the model with the classification categories as the output.

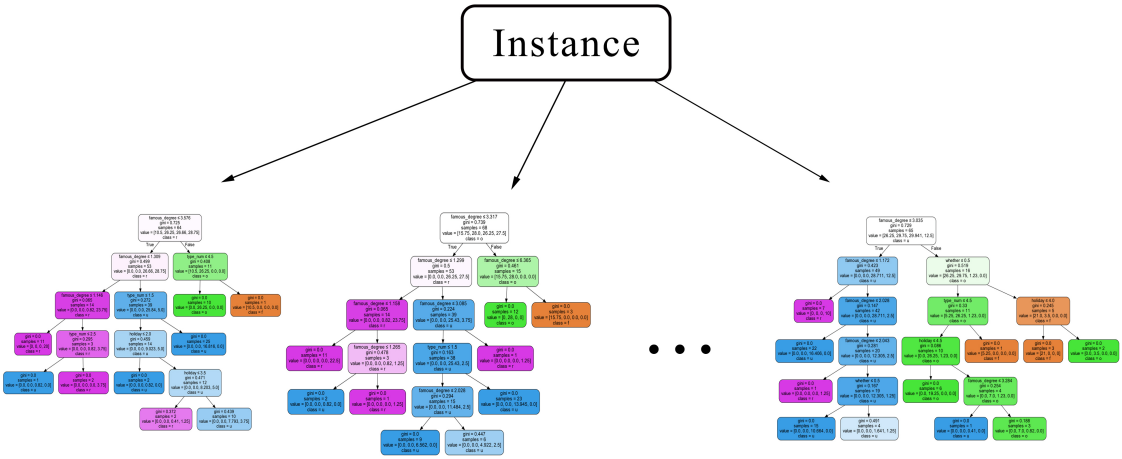

Figure 7

The final prediction results are shown in the following table.

Table 6 Random forest prediction results

Actual	Predicted
B	B
B	C
A	A
C	C
B	B
⋮	⋮

test accuracy: 0.813

6. Task 2: BPNN Forecast Movie Box Office

This question requires the construction of a box office prediction model. Common box office prediction models include multiple regression analysis, neural networks, etc. Then, to be more specific about the method, logistic regression, decision tree, random forest, and BPNN can be used. Here we tried the construction of BPNN. According to the effective prediction features selected in feature engineering, five features are used along this question: type, famous_degree, other, holiday, and word.

6.1 BPNN

BP neural network is the most basic neural networks with forward propagation of output results and backward propagation of errors. BP neural network is a supervised learning method and requires human specification of labels in the training set. The standard BP algorithm can approximate a nonlinear function with arbitrary accuracy using only three layers of feedforward nets.[2]

6.2 Adam Optimization Algorithm

The Adam optimization algorithm is an extension of stochastic gradient descent and has recently been widely used in computer vision and natural language processing for deep learning applications.[3]

There are many benefits of using Adam for non-convex optimization problems, such as simple algorithms, higher computational efficiency, smaller memory requirements, and ideal approach to data problems.

One of the key components of Adam is that it uses exponential weighted moving averages (also known as leaky averaging) to obtain an estimate of both the momentum and also the second moment of the gradient. That is, it uses the state variables.

$$\begin{aligned} \mathbf{v}_t &\leftarrow \beta_1 \mathbf{v}_{t-1} + (1-\beta_1)\mathbf{g}_t, \\ \mathbf{s}_t &\leftarrow \beta_2 \mathbf{s}_{t-1} + (1-\beta_2)\mathbf{g}_t^2 \end{aligned} \tag{7}$$

The common choices are $\beta_1 = 0.9$ and $\beta_2 = 0.999$, which are the parameters we set in this problem.

And the normalized state variables are given by[4]

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta_1^t} \text{ and } \hat{\mathbf{s}}_t = \frac{\mathbf{s}_t}{1-\beta_2^t} \tag{8}$$

Armed with the proper estimates we can now write out the update equations. First, we rescale the gradient in a manner very much akin to that of RMSProp to obtain

$$\mathbf{g}_t' = \frac{\eta \hat{\mathbf{v}}_t}{\sqrt{\hat{\mathbf{s}}_t} + \epsilon} \tag{9}$$

Now we have all the pieces in place to compute updates. This is slightly anticlimactic and we have a simple update of the form

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} - \mathbf{g}'_t \tag{10}$$

6.3 Box office forecast

When we build the neural network, we only need to take into account the 5 filtered effective feature factors as the output, according to the empirical formula[5]

$$n = \frac{N_{test}}{\sqrt{N_i + N_o} + a} \tag{11}$$

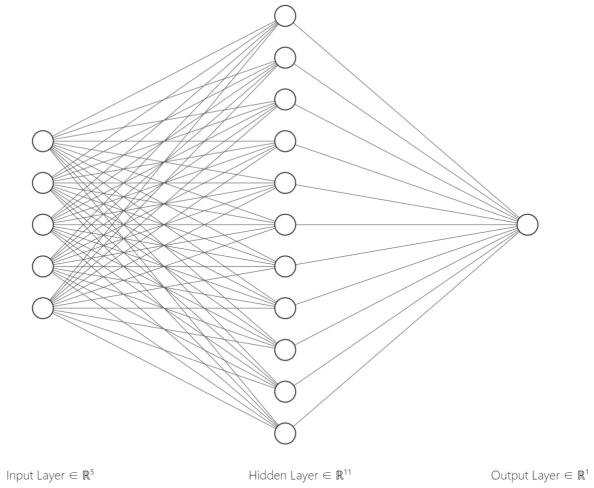

Figure 8 Neural network architecture

We set the hidden layers to 11 and output layer to 1 based on the task of predicting box office. For the overall data set, we select 100 items as the training set and 40 items as the test set to verify the ability of the model to predict box office.

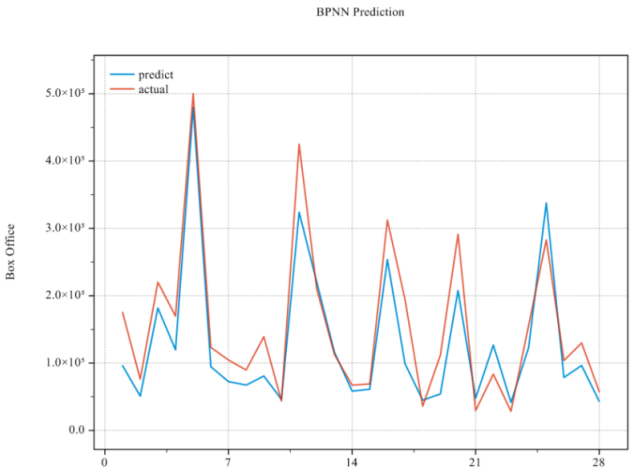

Figure 9 Predict and Actual

For the classification result of the first question, the box office prediction is also verified in each category using this model.

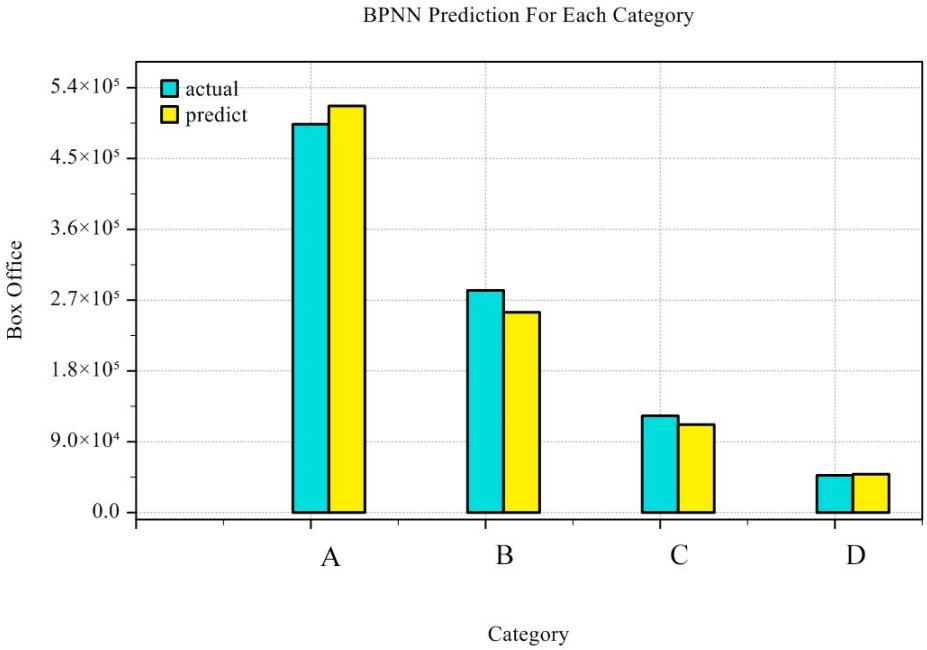

Figure 10 BPNN Prediction for Each Category

7. Task 3:

7.1 Nonlinear regression

First of all, we need to analyze the correlation between public opinion and box office, here we use Douban rating and Maoyan rating to roughly simulate public opinion evaluation; nowadays, people will use the scoring system of Douban, Maoyan and other movie-watching software to respond to the degree of personal love or approval of a movie, so we define the public opinion coefficient as

$$Com = \alpha x_1 + \beta x_2 \tag{12}$$

Where, α, β are the weighting coefficients of Douban and Maoyan respectively, Com is the measure of public opinion, the closer to 1 means the better the evaluation, the closer to -1 means the worse the evaluation.

Through the analysis, we can see that the rating of Maoyan is always slightly higher than that of Douban, so we take

$$\alpha = 0.6, \beta = 0.4$$

To describe the impact of public opinion, i.e., Douban and Cat's Eye ratings, on box office, we adopt a nonlinear regression approach for fitting.

A nonlinear regression is one in which the dependent variable y is nonlinear with respect to the regression coefficient $\beta_1, \ldots, \beta_m$ (and not the independent variable). This is where we need to look at the image to determine the nonlinear regression curve.

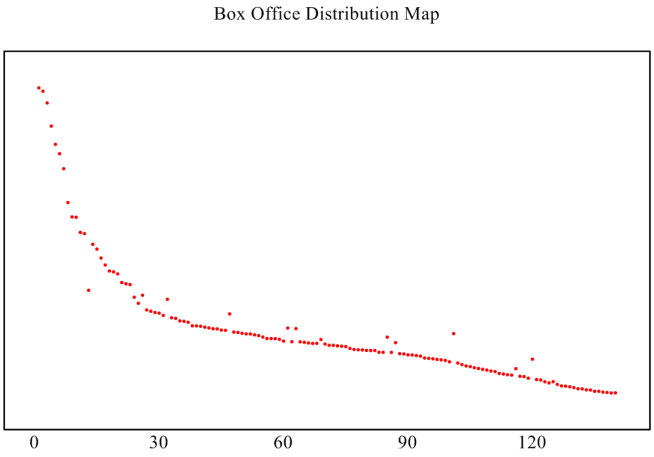

Figure 11 Total box office distribution

By looking at the images, we can determine that the regression equation should take the form

$$y = \frac{\beta_3 x_1 x_2}{1 + \beta_1 x_1 + \beta_2 x_2} \tag{13}$$

Finally, we use the computer you and get the value of the parameter should be

$$\begin{aligned} \beta_1 &= 5.648354 \\ \beta_2 &= 0.699052 \\ \beta_3 &= 2.235352 \end{aligned} \tag{14}$$

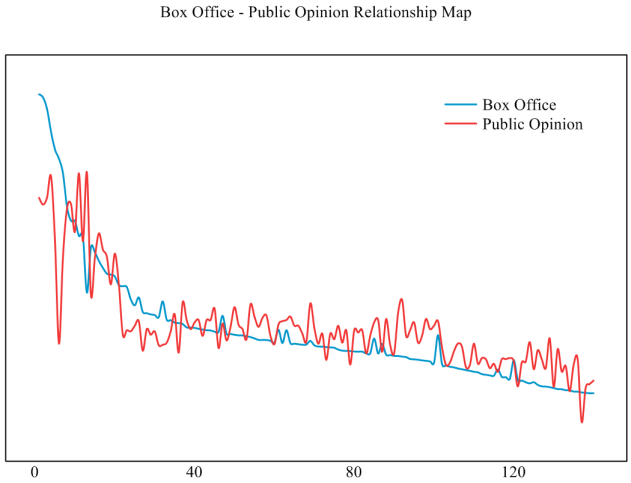

Figure 12 Fitting the curve

Two valid data are kept, so we finally get the relationship between the total box office and the ratings of Douban and Cat's Eye as follows

$$y = \frac{2.24x_1x_2}{1 + 5.65x_1 + 0.70x_2} \tag{15}$$

The images show that our results by fitting are similar to the actual trend and the average error $\bar{R}^2 = 0.3225$ is small, so we think that the re-fitting curve is relatively close and can prove that public opinion is correlated with movie box office.

7.2 NLP Network Navy Force Identification Model

Since online network navy is usually for a certain movie, we selected the more representative one in the data, "Douluo Continent", as the object of our analysis; the main actor of the movie is Xiao Zhan, we crawled from Douban, Maoyan, Weibo and other platforms to crawl the comments of the corresponding fans, and we filtered out the highest repetition book in the comments for visualization.

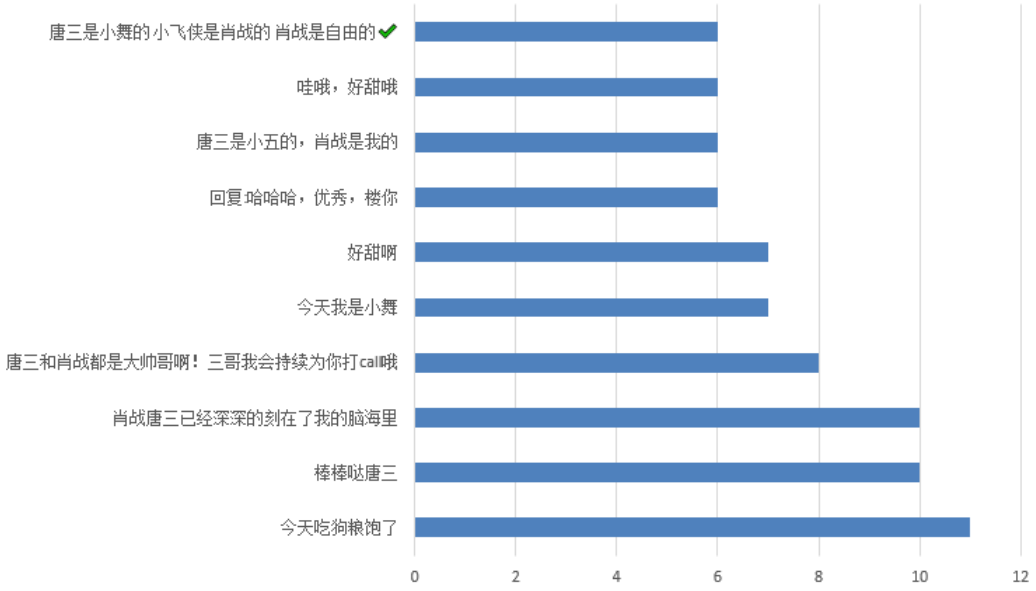

Figure 13 Frequency of statements

It can be seen that most of the comments are related to Xiao Zhan, and then referring to the comments of several movies, we can observe that most of the online comments, if they present the phenomenon of network navy, are related to the actors.

Therefore, we can use nlp statistics word frequency to build a relevant model to identify whether it is a network navy comment; we divide the comment into word frequency to draw a word cloud map.

Figure 14 Word cloud map

Finally we put the model we built into the different movie reviews we got, and the recognition results are shown in the following figure

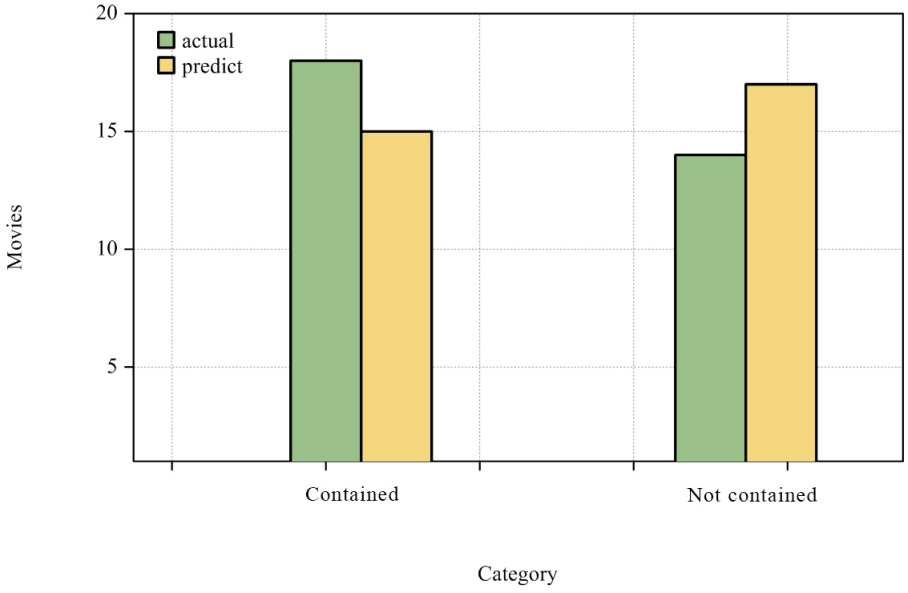

Figure 15 Network Navy Prediction

Thus we can obtain the confusion matrix of the model as

$$\begin{bmatrix} 15 & 3 \\ 2 & 13 \end{bmatrix} \tag{16}$$

The final calculated recognition accuracy of the model is 84.8%. The feasibility of the model is proved.

8. Task 4: Movie box office under the epidemic

8.1 Status of cinema attendance

After the peak of the COVID-19, the government lifted cinema viewing restrictions in the summer of 2020 and audiences were allowed to purchase tickets for offline viewing. However, cinema attendance was strictly limited to maintain a safe social density. In times of severe epidemic fluctuations, theater attendance may be limited to 50% or even less than 30%, resulting in a negative impact on the box office of movies during the epidemic.

This problem is explained and illustrated using the BP neural network model designed and trained in Title 2. We further argues for the realistic impact and future prediction of the epidemic on box office under different attendance rates.

8.2 Model Reconfiguration

There are several significant problems in model building as follows: the number of movies since the lifting of restrictions is not large enough to form a convincing training set; the repeated fluctuations of the epidemic make it difficult to collect box office data for a specific time period; the local nature of the epidemic leads to significant differences in attendance restriction policies in different places, and the collected box office data basically comes from national data. Therefore, we decided to conduct only a small batch of test validation.

For the BPNN model established in Title 2, we removed the "FamousDegree" feature and chose the first-week box office feature instead. This is because the "FamousDegree" of the original model is highly correlated with the box office, and training on small batches of data will result in serious overfitting. The network architecture of BPNN remains the same with 5 input layers, 11 hidden layers and 1 output layer. Ideally, the predicted value of the model should be significantly higher than the actual value (which reflects the negative impact of the epidemic on the box office).

Table 7 Model Comparison

Model	Prediction	Test Error
Original BPNN(epochs=2000)	nearly	0.0018
Original BPNN(epochs=500)	nearly	0.0629
BPNN without "FamousDegree"	less	0.5324
BPNN Fixed	more	0.3833

On the most original BPNN model, the prediction result shows about the same as the actual box office, which is not what we want. The prediction of the model after removing this feature is even lower than the actual box office, which is contrary to the reality and incorrect. The error rate of the model with "first-week box office" added as a feature is acceptable to a certain degree, and the overall box office prediction is higher than the actual box office, which indicates that the model is correct overall.

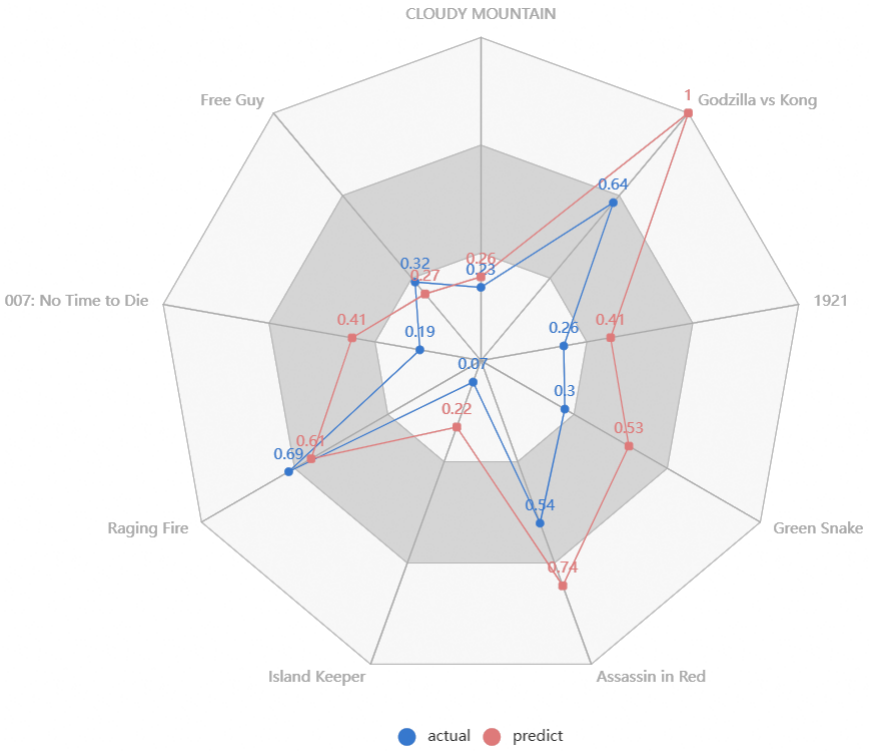

Figure 16 Model Radar Chart

The figure shows the predictions for a small batch (9 movies) in the new model. It can be observed that the predicted box office is generally larger than the actual box office, which reflects the realistic impact of attendance restrictions on box office under the epidemic.

For box office projections under three different attendance limits (30%, 50%, 75%) policies, the following relationships are given:

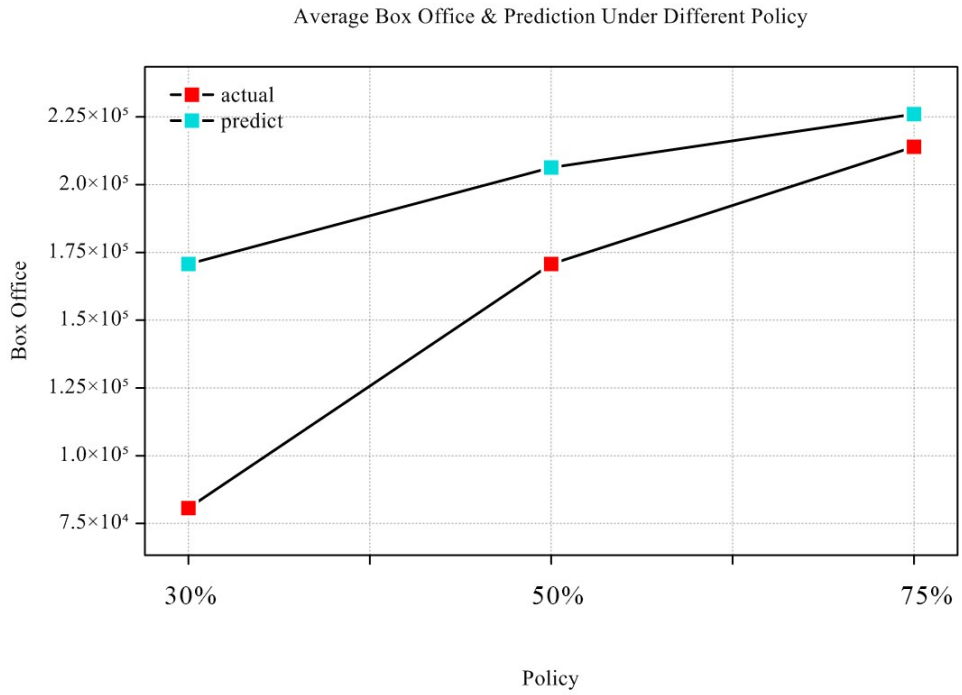

Figure 17 Average Box office & Prediction Under Different Policy

It can be observed that the model predicts that the difference in box office under the three attendance policies is significantly smaller than the difference between the actual values. In the most restrictive case (30%), the actual box office is severely affected, while the gradual relaxation of restrictions gradually approaches the predicted value.

9. Sensitivity Analysis

For the relationship between the total box office and the ratings of Douban and Maoyan:

$$y = \frac{2.24x_1x_2}{1 + 5.65x_1 + 0.70x_2} \tag{17}$$

We remember sensitivity

$$\begin{aligned} S_1(y, x_1) &= \frac{\partial y}{\partial x_1} \cdot \frac{x_1}{y} = \frac{2.24x_1x_2 + 1.568x_1x_2^2}{y(1 + 5.65x_1 + 0.70x_2)^2} \\ S_2(y, x_2) &= \frac{\partial y}{\partial x_2} \cdot \frac{x_2}{y} = \frac{2.24x_1x_2 + 12.656x_1^2x_2}{y(1 + 5.65x_1 + 0.70x_2)^2} \end{aligned} \tag{18}$$

Substituting a point $(x_1, x_2, x_3) = (-0.3179, -0.8424, -0.2384)$ in the sample, we get

$$S_1 = -0.5376 \qquad S_2 = 1.0431$$

It can be seen that when x_1 increases by 1%, y only decreases by 0.5376%, and when x_2 increases by 1%, y only increases by 1.0431%, so the change model is more stable and reliable.

10. Conclusion

Before formally solving the problem in the stem, we obtained some basic information about the movie through magazines and the Internet, and then we pre-processed this information to make it more convenient and reasonable for us to use later.

In problem 1, we first filtered the important influencing factors through feature engineering, omitted the unimportant factors to make our model simpler and with stronger generalization ability, and finally classified the filtered and processed influencing factors through random forest;

In problem 2 is a typical multi-input, single-output problem, so we decided to use BP neural network to build the model, and the final fitting;

In problem 3, we mapped the relationship between public opinion and box office to the relationship between Douban rating and Maoyan rating and box office, and fitted a reasonable result through non-linear fitting, while for the judgment and identification of online network navy, we constructed the model by counting the high-frequency words in the reviews, so as to improve the identification rate of network navy;

In problem 4, due to the epidemic limiting the maximum In order to estimate the degree of impact, we used the BPNN_fix model to simulate and finally concluded that when the country gradually increases the maximum attendance value, the more the total box office of these movies solves the predicted result of our model II, which is in line with our common sense.

References

[1] Han ZM, Yuan BH, Chen Y, et al. An effective GBRT-based early movie box office prediction model[J]. Computer Application Research, 2018.

[2] https://github.com/ccc-hhh/BPNN-regression-and-classify

[3] Kingma, D. P., & Ba, J. (2014). Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[4] https://d2l.ai/chapter_optimization/adam.html

[5] Cilimkovic M. Neural networks and back propagation algorithm[J]. Institute of Technology Blanchardstown, Blanchardstown Road North Dublin, 2015, 15: 1-12.

Appendix

Listing 1: Impact Calculation

```
Dir_name = "";
for i = 1:length(Dir)
    sp_Dir = strsplit(Dir(i),'/');
    for j = 1:length(sp_Dir)
        for v = 1:length(Dir_name)
            if strcmp(sp_Dir(j),Dir_name(v))
                Dir_Pfb(v) = Dir_Pfb(v) + A_OFBX(i);
                Dir_count(v) = Dir_count(v)+1;
                v = -1;
                break
            end
        end
        if v ~= -1
            Dir_name(v+1) = sp_Dir(j);
            Dir_Pfb(v+1) = A_OFBX(i);
            Dir_count(v+1) = 1;
        end
    end
end
Dir_name = Dir_name(2:end);
Dir_Pfb = Dir_Pfb(2:end);
Dir_count = Dir_count(2:end);
Dir_inf = Dir_Pfb./Dir_count;
Dir_eff = [Dir_name',Dir_count',Dir_Pfb',Dir_inf'];

LR_name = "";
for i = 1:length(Lead_role)
    sp_LR = strsplit(Lead_role(i),'/');
    for j = 1:length(sp_LR)
        for v = 1:length(LR_name)
            if strcmp(sp_LR(j),LR_name(v))
                LR_Pfb(v) = LR_Pfb(v) + A_OFBX(i);
                LR_count(v) = LR_count(v)+1;
                v = -1;
                break
            end
```

```
        end
        if v ~= -1
            LR_name(v+1) = sp_LR(j);
            LR_Pfb(v+1) = A_OFBX(i);
            LR_count(v+1) = 1;
        end
    end
end
LR_name = LR_name(2:end);
LR_Pfb = LR_Pfb(2:end);
LR_count = LR_count(2:end);
LR_inf = LR_Pfb./LR_count;
LR_eff = [LR_name',LR_count',LR_Pfb',LR_inf'];

Cor_name = "";
for i = 1:length(Corpre)
    sp_Cor = [strsplit(Corpre(i,1),'/'),strsplit(Corpre(i,2),'/')];
    for j = 1:length(sp_Cor)
        if sp_Cor(j) == ""
            continue
        end
        for v = 1:length(Cor_name)
            if strcmp(sp_Cor(j),Cor_name(v))
                Cor_Pfb(v) = Cor_Pfb(v) + A_OFBX(i);
                Cor_count(v) = Cor_count(v)+1;
                v = -1;
                break
            end
        end
        if v ~= -1
            Cor_name(v+1) = sp_Cor(j);
            Cor_Pfb(v+1) = A_OFBX(i);
            Cor_count(v+1) = 1;
        end
    end
end
Cor_name = Cor_name(2:end);
Cor_Pfb = Cor_Pfb(2:end);
Cor_count = Cor_count(2:end);
```

```
Cor_inf = Cor_Pfb./Cor_count;
Cor_eff = [Cor_name',Cor_count',Cor_Pfb',Cor_inf'];
```

Listing 2: Date feature quantification

```
for i = 1:length(Data)
    sp_Data = strsplit(Data(i),'-');
    month = str2num(sp_Data(2));
    day = str2num(sp_Data(3));
    if month <= 2 && day <= 13
        tips(i) = 1;
    elseif (month == 4 && day >= 23) || (month == 5 && day == 1)
        tips(i) = 2;
    elseif (month == 7 && day >= 15) || (month == 8 && day <= 15)
        tips(i) = 3;
    elseif (month == 9 && day >= 23) || (month == 10 && day == 1)
        tips(i) = 4;
    else
        tips(i) = 5;
    end
end
tips = tips';
```

Listing 3: K-means

```
L = length(A_FOBX);
Y=ones(1,L);
N = L;

K = 4;
n = 4;
m = 1;
eps = 1e-7;

u1 = [A_FOBX(1),1];
u2 = [A_FOBX(25),1];
u3 = [A_FOBX(57),1];
u4 = [A_FOBX(77),1];

U1 = zeros(2,100);
```

```
U2 = zeros(2,100);
U3 = zeros(2,100);
U4 = zeros(2,100);

U1(:,2) = u1;
U2(:,2) = u2;
U3(:,2) = u3;
U4(:,2) = u4;
E = zeros(n,N);

while(abs(U1(1,m) - U1(1,m+1)) > eps || abs(U2(1,m) - U2(1,m+1)) > eps ||
    abs(U3(1,m) - U3(1,m+1)) > eps || abs(U4(1,m) - U4(1,m+1)) > eps)
    m = m +1;

for i = 1 : N
    E(1,i) = abs(A_FOBX(i) - U1(1,m));
end
for i = 1 : N
    E(2,i) = abs(A_FOBX(i) - U2(1,m));
end
for i = 1 : N
    E(3,i) = abs(A_FOBX(i) - U3(1,m));
end
for i = 1 : N
    E(4,i) = abs(A_FOBX(i) - U4(1,m));
end

A = zeros(2,N);
B = zeros(2,N);
C = zeros(2,N);
D = zeros(2,N);

for k = 1: N
    [MIN,index] = min(E(:,k));
    if index == 1
        A(1,k) = A_FOBX(k);
        A(2,k) = Y(k);
    elseif  index == 2
        B(1,k) = A_FOBX(k);
```

```
        B(2,k) = Y(k);
    elseif  index == 3
        C(1,k) = A_FOBX(k);
        C(2,k) = Y(k);
    else
        D(1,k) = A_FOBX(k);
        D(2,k) = Y(k);
    end
end
indexA = find(A(1,:) ~= 0);
indexB = find(B(1,:) ~= 0);
indexC = find(C(1,:) ~= 0);
indexD = find(D(1,:) ~= 0);


U1(1,m+1) = mean(A(1,indexA));
U1(2,m+1) = mean(A(2,indexA));
U2(1,m+1) = mean(B(1,indexB));
U2(2,m+1) = mean(B(2,indexB));
U3(1,m+1) = mean(C(1,indexC));
U3(2,m+1) = mean(C(2,indexC));
U4(1,m+1) = mean(D(1,indexA));
U4(2,m+1) = mean(D(2,indexA));


juleizhongxin=[U1(1,m+1),U2(1,m+1),U3(1,m+1)];

end
```

Listing 4: The lingo source code

```
y = A_FOBX;
% plot(y,'r.');
x = [B_score,CE_score];
xy = [x,y];
xyn = zscore(xy);
x = xyn(:,1:2);
y = xyn(:,3);

targe = @(beta,x) (beta(3)*x(:,1).*x(:,2))./(1+beta(1)*x(:,1)+beta(2)*x(:,2));
```

```
beta0 = [1,1,1]';
[beta,r,j] = nlinfit(x,y,targe,beta0);
yy = (1+beta(1)*x(:,1).*x(:,2))./(beta(2)+beta(3)*x(:,1));
```

Listing 5: BPNN

```
import numpy as np
from sklearn.model_selection import train_test_split

path = "/Users/keyoncheung/Desktop/"
datafile = "data.csv"
train_num = 140

def DataSet():
    data = np.genfromtxt(path + datafile, delimiter=",")

    X = data[:, 1:]
    X = X[1:, ]

    y = data[:, 0]
    y = y[1:]

    x_train = X[:train_num]
    x_test = X[train_num:]
    y_train = y[:train_num]
    y_test = y[train_num:]

    return x_train, x_test, y_train, y_test

def normalized(x_data, y_data):
    e = 1e-7
    for i in range(x_data.shape[1]):
        max_num = np.max(x_data[:, i])
        min_num = np.min(x_data[:, i])
        x_data[:, i] = (x_data[:, i] - min_num + e) / (max_num - min_num + e)
    y_data = (y_data - np.min(y_data) + e) / (np.max(y_data) - np.min(y_data)
        + e)
    return x_data, y_data
```

```
def inverse_normalized(output, y_test):
    output = output * (np.max(y_test) - np.min(y_test)) + np.min(y_test)
    return output


if __name__ == "__main__":
    x_train, x_test, y_train, y_test = DataSet()
    print(y_test)

%%
import torch
from movies_data import *
import matplotlib.pyplot as plt
import os
import torch.nn.functional as F

learning_rate = 0.01
betas = (0.9, 0.999)
alpha = 0.9

path = "/Users/keyoncheung/Desktop/"

class Net(torch.nn.Module):
    def __init__(self, n_feature, n_hidden, n_output):
        super(Net, self).__init__()

        self.hidden = torch.nn.Linear(n_feature, n_hidden)
        self.predict = torch.nn.Linear(n_hidden, n_output)

    def forward(self, x_train_normalized):

        x_train_normalized = torch.tanh(self.hidden(x_train_normalized))
        x_train_normalized = self.predict(x_train_normalized)
        return x_train_normalized


def train(model, epochs, x_train, y_train, path):

    optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate,
```

```
        betas=betas)
    loss_func = torch.nn.MSELoss()

    plt.ion()
    plt.show()
    for i in range(epochs):
        model.train()
        prediction = model(x_train)
        loss = loss_func(prediction, y_train)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if i % 50 == 0:
            plt.cla()
            plt.title("Training Process")
            X = np.linspace(0, len(np.array(y_train)), len(np.array(y_train)))
            plt.plot(X, y_train, marker='.', label="origin data")
            plt.xticks([])
            plt.plot(X, prediction.detach().numpy(), 'r-', marker='.',
                label="train", lw=1)
            plt.xticks([])
            plt.legend(loc="upper right")
            plt.pause(0.1)
    state = {'model': model.state_dict(), 'optimizer': optimizer.state_dict(),
        'epoch': epochs}
    torch.save(state, path)

def test(model, x_test, y_test, path):
    checkpoint = torch.load(path)
    model.load_state_dict(checkpoint['model'])
    optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate,
        betas=betas)
    optimizer.load_state_dict(checkpoint['optimizer'])

    output = model(x_test)
    loss = torch.nn.MSELoss()
    error = loss(output, y_test)
    deviation_mean = torch.mean(torch.abs(torch.sub(y_test, output)))
```

```
    plt.title("Testing Process")
    x = np.linspace(0, len(y_test), len(y_test))
    plt.plot(x, y_test.detach().numpy(), color='blue', marker='.',
        label="actual")
    plt.plot(x, output.detach().numpy(), color='red', marker='.',
        label="predict")
    plt.xticks([])
    plt.text(0, 0.97, 'Error=%.4f' % error, fontdict={'size': 15, 'color':
        'red'})
    plt.legend(loc="upper right")
    plt.show()

    output = output.detach().numpy()
    output = np.squeeze(output)
    x_train, x_test, y_train, y_test = DataSet()
    output = inverse_normalized(output, y_test)
    print("predict: ", output)
    print("actual: ", y_test)

def main():
    path_adam = path + "BPNN_adam.pt"
    x_train, x_test, y_train, y_test = DataSet()
    x_train_normalized, y_train_normalized = normalized(x_train, y_train)
    x_test_normalized, y_test_normalized = normalized(x_test, y_test)

    x_train_normalized =
        torch.from_numpy(x_train_normalized).clone().detach().float()
    y_train_normalized =
        torch.from_numpy(y_train_normalized).unsqueeze(1).clone().detach().float()

    x_test_normalized =
        torch.from_numpy(x_test_normalized).clone().detach().float()
    y_test_normalized =
        torch.from_numpy(y_test_normalized).unsqueeze(1).clone().detach().float()

    net = Net(n_feature=5, n_hidden=11, n_output=1)

    train(net, epochs=500, x_train=x_train_normalized,
        y_train=y_train_normalized, path=path_adam)
```

```
    test(net, x_test_normalized, y_test_normalized, path_adam)


if __name__ == "__main__":
    main()
```