Team Number :	0968

Problem Chosen :	**C**

2016 APMCM summary sheet

The star and drama questions are based on Large Data. To deal with these questions, it's significant to search reliable data from the Internet, filter away the bad value, determine the weights and give the prediction.

First, regarding the TV drama ranking, we select the TV drama scores, the critics number of each drama and the number of TV drama sets as top 3 significant indexes to judge the final ranking. To find to the most reasonable weights of the 3 indexes, we use TOPSIS to calculate the best weights. At last, We give the rank of dramas based on Gray Relational and Single-level Comprehensive Evaluation when the relationship of indexes is unclear if calculated directly.

Second, we consider that the indexes judging the star popularity are rich and unofficial, so Apriori algorithm can be used to filter the unimportant indexes and only retain the high weight indexes. By traversing multiple data collected from the Internet, we attain final frequent n-item-sets, in which are the most significant indexes. Then using PCA to get the weight of relevant indexes. Besides that, the special cases should be considered as well, such as the fierce changes caused by tidbits in short time. Finally, we compare the rank based on our indexes with the rank existed in official websites, finding a approximate equal.

Third, aiming to build a new team to create new products, we can use the data searched by crawler from the Internet, such as click rate, critics, starring stars, production team and so on. To filter away the unimportant indexes, Stepwise Regression method can be used, then we can get the regression equation after standardization. Through this equation, every indexes will correspond to a weight, which measures the contribution to the final index. Then, compare the derived ranking with the official ranking to get trustworthiness and judge that the index is acceptable. Based on final index, describe an ideal production team.

Last, for getting the most fitting recommendation from the audience's browsing history and rating scores on each channel. Here using LDA algorithm, to find the main trend and main types the history. The assignment mainly relies on the probability that the history data is of different types, then, by the Cosine Calculation, finding the data with highest fitting degree. Using the data after standardization, we prove that the credibility is up to 93.2%. What's more, when depending on ratings, this model is reliable as well.

Every model has been tested by data from reality, which is from the Internet searched by crawler in Python.

Key words：Dram Rank, TOPSIS, GRA, PCA, Stepwise Regression, LDA, Weight

Contents

1 Restatement of the Problem

The current Chinese television market is highly competitive and full of different styles and topics. Though the annual TV drama production remains high, but TV stations do not have much desire to buy these dramas, which leads to the oversupply of TV dramas. Quantity has never been a problem for TV dramas, yet quality has always been the problem we need to solve. How to lower cost and get rid of the hasty and crude TV investment hold the future development for TV dramas.

In the meantime, using Big Data from 2014 as an analysis instrument to test TV market has been quite successful. Though Big Data cannot create the script, it can analyze data and forecast pretty precisely. This could be applied to script writing, TV rating forecast, outcome of TV commercials, and TV drama purchase. It is possible to reduce TV investment risk, improve script quality, and forecast audience response to ensure maximum benefits.

In film and television drama market, how to evaluate and customize the film and television drama and other issues has always been the center of focus during a production. Now please try to use mathematical modeling methods to solve the following problems.

- Rank the TV dramas based on the ranking index and name your top 10.
- Please collect and use related data as foundation, design a star popularity index, and try to prove the attainability of your index by giving a real-life example from this year.
- Describe an ideal production team, including the producer and actors. Try to prove your point with a real-life example.
- By viewing history and the ratings of programs, find the script content most suitable for audience and each local TV station. Collect related data, provide a solution by using mathematical modeling methods, and use a real-life example to prove your point.

2 Assumptions and Justifications

1) **Data can reflect the popularity correctly.** There is no internet mercenaries to maliciously improve popularity.
2) **All the forums named by star or drama names are talking about the theme.** There are all related posts in the forums
3) **There is no commercial speculation on the rankings from the well-known websites.** All the rankings rely on reality, and must be objective.
4) **The models have universality.** Because the data from the Internet cannot contain all the stars and dramas. Considering by Calculating large enough scale of data. The models derived can work on all the stars and dramas. And the error is too small that can be organized.

3 Notations

Table 1: Constants

$\boldsymbol{X}$	The influence indexes matrix of drama
$\boldsymbol{E_j}$	Entropy of each significant index
$\boldsymbol{h_j}$	Weights of the significant indexes in the measure
$\boldsymbol{\gamma_i}$	Correlation between to indexes
$\boldsymbol{M_n}$	Frequent n-sets
$\boldsymbol{P}$	The probability of the parameters
$\boldsymbol{R}$	Relation matrix

P.S. The other symbols will be given in the notes

4 Models of Solution

4.1 Comprehensive ranking model by Gray Relational TOPSIS

At present, the Chinese TV market is increasingly fierce; a variety of film and television works vary. For the audience, a reliable TV drama ranking, the demand for the audience trend is particularly important. In this paper, according to the television data in Annex 1 and Annex 2, the TV drama scores, the critics number of each drama and the number of TV drama sets are selected as the three indicators of the TV drama ranking, and the comprehensive score ranking of the films is obtained based on TOPSIS.

In the paper of 429 TV shows, the number of first i-th TV critics is the ratio of the total critics and the number of sets：

$$s_i = \frac{C_i}{m_i}(1 \le i \le 429) \tag{1}$$

Where, s_i the first i-th TV critics number, the C_i is the total critics number, the m_i is the number of sets.

Because the TV drama scores, the critics number of each drama and the number of TV drama sets are not independent of each other, so the simple linear addition is not desirable. Based on grey relational analysis model to quantify between selected indicators unclear grey relation, and through the TOPSIS method[1] to optimal ranking.

4.1.1 Determining the weight by TOPSIS

First, extract the Annex 1 and Annex 2 in the TV drama scores, the total critics and number of TV drama sets, through the ratio method to determine the critics of each episode, the establishment of film critics, film critics and film and television sets the number of initial matrix.

The Entropy method is used to determine the weight of the indicator according to the amount of information contained in the above three indexes. The smaller the entropy is, the greater the degree of variation of the index is, the more information provided has, the greater the weight occupied in the comprehensive evaluation. The main steps are as follows:

1) Constructing the evaluation matrix of the three indicators of all TV drama

2) objects in the Annex:

$$\boldsymbol{X} = (x_{ij})_{429\times3}，（i = 1,2, \dots 429；j = 1,2,3） \tag{2}$$

3) Finding the standardization index:

$$\boldsymbol{P_{ij}} = \frac{x_{ij}}{\sum_{i=1}^{429} x_{ij}}，（\boldsymbol{i = 1,2, \dots 429；j = 1,2,3}） \tag{3}$$

4) Finding the entropy of each index:

$$\boldsymbol{E_j} = \frac{\sum_{i=1}^{429} P_{ij}\, ln\, P_{ij}}{ln\, 429} \tag{4}$$

5) Finding the weights of each indicator:

$$\boldsymbol{h_j} = \frac{1-E_j}{\sum_{i=1}^{3}(1-E_j)}，（\boldsymbol{j = 1,2,3}） \tag{5}$$

The entropy method can judge the information content of three evaluation indexes, such as the TV drama scores, critics and television drama sets, so as to get the weight distribution of these three indexes in the course of comprehensive evaluation.

4.1.2 Determining the ranking by Gray Relational

Because of the gray relation between the selected indexes, the gray relational analysis method can deal with the gray system which is not completely clear. The gray relational analysis method is suitable for the problem of small sample irregularity evaluation like this comprehensive score ranking. As the evaluation level is relatively simple, the gray relational analysis [2] is introduced into the single-level comprehensive evaluation.

For the selected objects in the Annex, the establishment of three indicators of the gray relational single-level comprehensive evaluation. The data of the evaluation index are standardization, and there is $x_1, x_2, x_3，x_i = [x_i(1), x_i(2), \dots, x_i(429)，i = 1,2,3]$.And the ideal plan is $x_0 = \{1,1, \dots 1,\}$,showing that the maximum value of relevance about each item is 1,and between the items and k-th element of the correlation coefficient are:

$$\boldsymbol{\sigma_i(k)} = \frac{\Delta min+\rho\Delta max}{\Delta_i(k)+\rho\Delta max}，（\boldsymbol{i = 1,2, \dots, 429；k = 1,2,3}） \tag{6}$$

Where,

$\Delta\text{min} = min_i min_k|x_0(k) - x_i(k)|$； $\Delta\text{max} = max_i max_k|x_0(k) - x_i(k)|$, ρ is the resolution factor, ρ is 0.5. So the correlation about i-th plan of ranking and ideal plan is:

$$\boldsymbol{\gamma_i} = \sum_{k=1}^{3} \boldsymbol{W_k \sigma_i(k)} \tag{7}$$

By formula (7), we can get that when the degree of association is higher, the closer the scheme is to the ideal scheme, the higher the scheme ranking is.

4.1.3 Solution of ranking scheme

The scores of TV drama, the critics number of each drama and TV sets are assigned to the weights. The influence factors of the comprehensive score of the selected objects according to the weights are the scores, the number of critics and the number of sets, 0.7108, 0.2761, 0.0130. According to the gray relational to get the final comprehensive score, and the top ten scores is the top ten in the ranking.

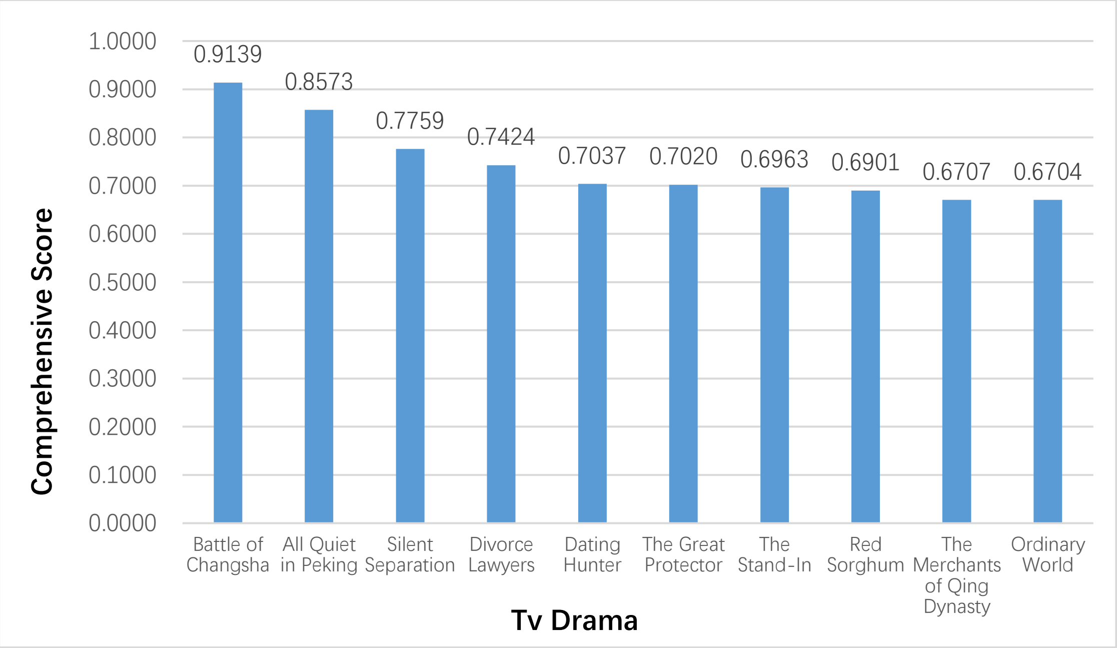

Figure 1: Top 10 in the final ranking according Gray Relational

4.2 Building the star popularity index by Apriori-PCA

4.2.1 Determining the measure by Apriori

For the measure of the star popularity index is very rich. Using Python's crawler function, we can obtain the ranking of the popular website index and index reference. Finding the most of the sites are based on their own search engine search quantity to rank, so that the measure of the stars of the target are up to 32 indicators. However, in these indicators, not all of the indicators on the overall popularity ranking have a greater impact; the introduction of subsequent calculations will increase the redundancy of the algorithm. We can introduce frequent item-sets mining Apriori algorithm[3], filter all the indicators, and finally we can retain the ranking of a number of popular indicators.

First, all the ranking data are traversed, and the number of times of single index is counted, and the indexes of less than certain threshold are filtered to get frequent item-sets.

1) Frequent item-setsM_1:

$$\boldsymbol{M_1} = \{\boldsymbol{M_i} \geq \boldsymbol{\epsilon}\}, \ (\boldsymbol{i} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{m}) \tag{8}$$

In formula (8), M_i is all the metrics collected, ϵ is minimum threshold, m is the total number of search metrics.

And then traversal, two combinations of indicators, if an indicator of the two indicators are in a single set, the statistical indicators of the number of occurrences, or skip. Filter out the minimum threshold to get frequent binomial sets.

2) Frequent Binomial-sets M_2:

$$\boldsymbol{M_2} = \{\boldsymbol{M_i}, \ \boldsymbol{M_{i'}}\}, \ \left(\boldsymbol{M_i}, \ \boldsymbol{M_{i'}} \in \boldsymbol{M_1}; \ \boldsymbol{i} = \mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{m}\right) \tag{9}$$

3) Analogy to Frequent n-item-sets M_n:

$$\boldsymbol{M_n} = \{M_i, \ M_{i'} \ldots M_x\}, \ (M_i, \ M_{i'}, \ M_x \in \boldsymbol{M_{n-1}}; \ i = 1,2, \ldots, m; \ x \leq m) \tag{10}$$

Using Apriori algorithm to get the 32 indicators for frequent filtering, and finally get 14 of the popular evaluation index.

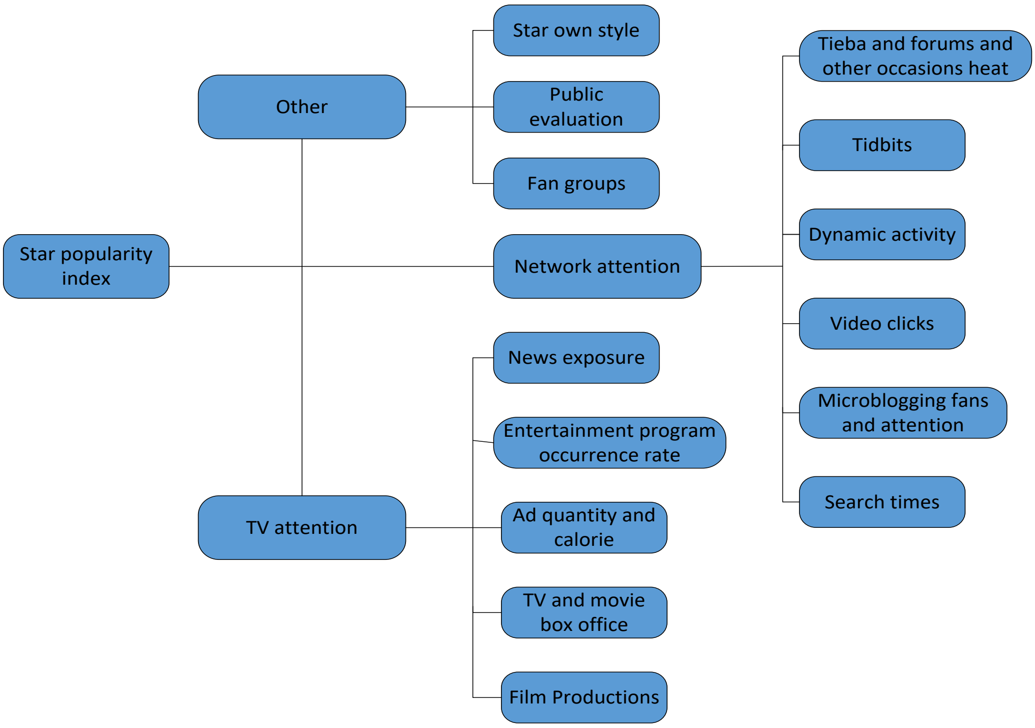

Figure 2: The possible popular evaluation index

The popularity of the above evaluation indicators, although some popularity index on the popularity index contributed significantly, but with other indicators will be repeated or even be replaced. So we can use the PCA to analyze the index to obtain a number of indicators of a larger contribution, and the final indicators are not or have little coherence.

4.2.2 Determining the weight PCA

For the data obtained in section 4.2.1, although each indicator is genetically different, but the correlation between different indicators tend to produce a factor on the star popularity index plays a dominant role. In order to find the correlation and distribution weight, this paper introduces principal component analysis[4] to solve the problem. Proceed as follows:

1) Eliminate the influence of dimension, carry on standardization processing:

$$\widetilde{S_{ij}} = \frac{S_{ij}-\bar{S_i}}{s_i} \tag{11}$$

Where S_i and s_i are the mean and standard deviation of the i-th indicator. We can eliminate the impact of different dimensions; in the normalization of the transformation, it will not change the correlation coefficient between the variables.

2) Calculate the correlation coefficient matrix of the normalized data and find the eigenvalues and eigenvectors:

$$r_{ii'} = \frac{\sum_{k=1}^{14} \widetilde{S_{ik}}\widetilde{S_{i'k}}}{14-1}, \quad (i, i' = 1,2,\dots,14) \tag{12}$$

And the correlation coefficient matrix is $R = (r_{ii'})_{14\times 14}$, $r_{ii} = 1$, $r_{ii'} = r_{i'i}$. Where the eigenvalues is $\lambda_i (i = 1,2,\dots,14)$, the eigenvectors is $L_i (i = 1,2,\dots,14)$.

3) Calculate the contribution rate $T_k = \frac{\lambda_i}{\sum_{i'=1}^{14} \lambda_{i'}}$ and the cumulative contribution rate $D_k = \sum_{i'=1}^{k} T_{i'}$. Select the eigenvalues $D_k \geq 85\%$ of $\lambda_1, \lambda_2, \dots, \lambda_x$, $(x < 14)$.

4) Obtain the weight of each index on the popularity index:

Let the calculate the contribution rate D_x about the eigenvalues of x-th main ingredient is 1, calculate that $T_1, T_2, \dots, T_x$ to new $T'_1, T'_2, \dots, T'_x$, this is the weight value of principal component index.

4.2.3 Coming to the final popularity index algorithm

From the section 4.2.2, the main impacts of the popularity index of several important index are the Tieba and Microblogging heat, films and television, network search quantity, fans heat and dynamic activities. The above five indicators can be used as a long-term measure, but according to the reality of the situation, the star tidbits in a short time can change their attention, thus affecting the popular index greatly.

Using Matlab can get that popular index and the five long-term index are in positive correlation in a short time. Because of tidbits. The influence formula on trend heat given by tidbits is:

$$y = Ke^{-d\beta}, \quad \beta > 0 \tag{13}$$

Where K is the proportional coefficient, d is the days after the tidbits, β is the attenuation factor.

According to microblogging hot search history can be obtained: after happening the tidbits, attention to the general is an attenuation in the main trend. Using reptile technology to get more data of the stars. When the attenuation factor $\beta \approx 0.041$, here is the best fit of the attention heat. The tidbits formula is:

$$y = Ke^{-0.041d} \tag{14}$$

So the final star popularity index equation is:

$$P = k_1m_1 + k_2m_2 + k_3m_3 + k_4m_4 + k_5m_5 + k_6Ke^{-0.041d} \tag{15}$$

Where the k_i are the weights. And m_1 is the Tieba and Microblogging heat, m_2 is the films and TV shows, m_3 is the network search quantity, m_4 is the fans heat, m_5 is the dynamic activities, m_6 is the tidbits.

Using the Matlab to process the data got by crawler program and the existing popularity index to establish the relationship between stars.

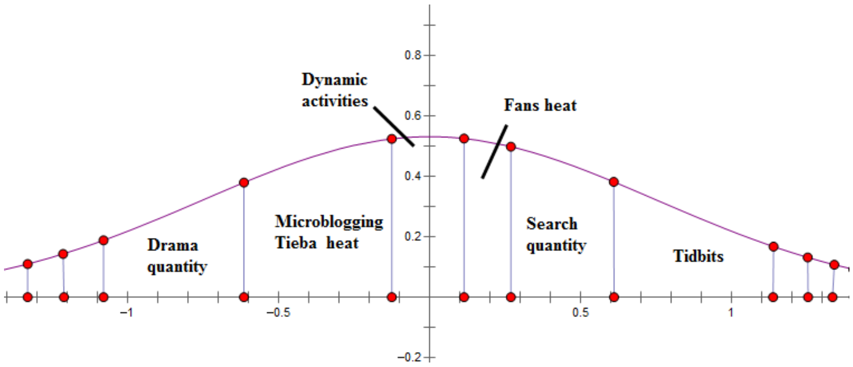

Figure 3: The Normal distribution of indexes influencing popularity

Here is the star popularity index equation with true weights:

$$P = 0.2806m_1 + 0.2231m_2 + 0.1804m_3 + 0.0407m_4 + 0.0942m_5 + 0.2010e^{-0.041d} \tag{16}$$

4.2.4 Proving of the index

Using the crawler program in Python, we can get the current daily-changed star popularity ranking on the Internet. Meanwhile, the daily-changed significant indexes shown in section 4.2.3 are also obtained. Put the indexes and the popularity index into the formula (16), the result we get are very closed to the official data. Using Matlab, we can get the mean square error is between 17 to 23 in the top 50 ranking, and between 33 to 50 in the top 100 ranking. Finally, the star popularity index equation given by section 4.2.3 is acceptable.

Table 2:Star official ranking(Top 26)

	Names		TV drama
1	Liying Zhao	14	Jay Chou
2	Poetry	15	Jackie Chan
3	Yifeng Li	16	Hanliang Zhong
4	Bingbing Fan	17	Yang Mi
5	Hu Ge	18	Liu Tao
6	Tang Yan	19	Andy Lau
7	Yifan Wu	20	Deng Chao
8	Zhao Wei	21	Yang Yang

9	Xiaoming Huang	22	Li Chen
10	Lu Han	23	Sun Li
11	Jianhua Huo	24	Nicky Wu
12	Yifei Liu	25	Yuanyuan Gao
13	Shuang Zheng	26	Yixing Zhang

Table 3:Star ranking from formula(Top 26)

	Names		TV drama
1	Liying Zhao	14	Zhao Wei
2	Yifeng Li	15	Jackie Chan
3	Poetry	16	Hanliang Zhong
4	Bingbing Fan	17	Yang Mi
5	Hu Ge	18	Yang Yang
6	Tang Yan	19	Jianhua Huo
7	Yifan Wu	20	Deng Chao
8	Jay Chou	21	Andy Lau
9	Xiaoming Huang	22	Sun Li
10	Lu Han	23	Li Chen
11	Liu Tao	24	Yixing Zhang
12	Yifei Liu	25	Yuanyuan Gao
13	Shuang Zheng	26	Nicky Wu

4.3 Prediction on best team based on Stepwise Regression

4.3.1 Calculation on indexes weights by Stepwise Regression method[5]

Through the provided data and searched data, we can see that there are many indexes affecting the popularity of TV drama, such as: starring star, TV type, production team, production costs, broadcasting time and broadcasting channel and other indexes.

Table 4: Factors influencing the popularity of drama

Starring star	TV type
Production team	Production costs
Broadcasting time	Broadcasting channel
Geographical factors	Theme
Reflection on life	Propaganda power
Technological content in production	People group
Ad implantation	……

Because there are many indexes affecting the popularity of TV drama, stepwise regression method can be used to choose the most significant indexes. By screening the independent variables, the larger the number of independent variables is, the larger the regression square sum is, the smaller the residual square sum is, and the higher the quality of regression analysis is, which can effectively improve the accuracy of the regression model analysis.

1) Renumber the indexes and standardize the different dimensions:

Let $y_\alpha = x_{\alpha k}$, the number of the indexes is $k-1$, so its mathematical model is:

$$x_{\alpha k} = \beta_0 + \beta_1 x_{\alpha 1} + \beta_2 x_{\alpha 2} + \beta_3 x_{\alpha 3} + \cdots + \beta_{k-1} x_{\alpha k-1} \tag{17}$$

$\alpha = 1,2,\ldots,n$, (n is the number of sample)

Where, $S = \sum(x_{\alpha k} - \overline{x_k})^2$, $S_Q = S - S_U = \sum(x_{\alpha k} - \widehat{x_k})^2$,

What's more, the partial regression square sum of x_j is :

$$S'_U = \frac{b_j}{c_{jj}} \tag{18}$$

Where $\overline{x_k}$ is the arithmetic mean of $x_{\alpha k}$, b_j is the Partial regression coeffic

of x_j,, c_{jj} is the diagonal elements of the inverse matrix L^{-1}.

2) The regression model with the new index as the parameter is:

$$\widehat{x_k} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_{k-1} x_{k-1} \tag{19}$$

The initial data to the standardized regression mathematical model, the solution is:

$$z_{\alpha j} = \frac{x_{\alpha j} - \overline{x_j}}{S_j} \tag{20}$$

Where,

$$\overline{x_j} = \frac{1}{n}\sum_{\alpha=1}^{n} x_{\alpha j} \tag{21}$$

3) Here we can get:

$$S_j = \sqrt{l_{jj}} = \sqrt{\sum(x_{\alpha j} - \overline{x_j})^2} \tag{22}$$

Where, l_{jj} is the deviation sum of squares, $\sqrt{l_{jj}}$ is the root of the deviation sum

of squares, S_j^2 is the variance, S_j is the standard deviation.

4) Here we can get the initial regression curve model and correlation coefficient. Establish the correlation coefficient matrix:

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1\ k-1} \\ r_{21} & r_{22} & \cdots & r_{2\ k-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{k-1\ 1} & r_{k-1\ 2} & \cdots & r_{k-1\ k-1} \end{pmatrix}$$

Based on R, $B = [r_{1\ k} r_{2\ k} \ldots r_{k-1\ k}]^T$ could be calculated, through which we can calculate the regression model after standardized.

$$\widehat{z_k} = d_1 z_1 + d_2 z_2 + \cdots + d_{k-1} z_{k-1} \tag{23}$$

Where, $\widehat{z_k}$ is the beat fitting curve of the popularity of drama, d_{k-1} are the correlation coefficients of best indexes， z_{k-1} are the best indexes.

By formula (23), we can get a number of significant indexes of the TV drama and the corresponding weights. These indexes are considered as parameters and are put into the formula after being standardized, so that the size of unity can be avoided and the systematic error can be minimized. The weights of the indexes are in positive correlation with the coefficients in the formula (23), which means that assuming that the total weight is 1 and the weights are in multiple relationship; all the weights sum together is 1 and in the final formula of judging popularity, the corresponding weights can be calculated easily.

The final formula of judging TV drama popularity is：

$$P = 0.3634x_1 + 0.2184x_2 + 0.1723x_3 + 0.1018x_4 + 0.0102x_5 + 0.0778x_6 \tag{24}$$

Where x_1 is the star popularity, x_2 is the relevant element of types of the drama, x_3 is the relevant element of broadcasting channel, x_4 is the special effects level, x_5 is the broadcasting time, and x_6 is the propaganda power.

The star popularity is the mean popularity indexes of the starring stars. The x_2 is the numbers of the type. Through the data, we can calculate the popularity of different types, and show the in the formula in the form of numbers. The broadcasting channel and time, special effects level and propaganda power can be calculated in the same way. ***P.S.*** The star popularity indexes can be got in section 4.2. And in the formula (24), the total weight is not 1, because some indexes' contribution are so small to the whole index. Here we can ignore those indexes.

4.3.2 Validation of the index

Through the crawler program, collect a number of drama clicks quantity, scores, starring stars, team, channel and time, and other data. Put the standardization of these indexes into section 4.3.1 in the evaluation equation (24). Through this equation (24) to get a set of evaluation scores, and through the high and low scores can be arranged the TV drama rankings. In the ranking, the data are from the official network. We can compare the official data with the calculated data to calculate the variance of the size of the two data to determine the credibility.

Table 5: The official ranking of some dramas

	TV drama	Ratings(%)		TV drama	Ratings(%)
1	Beautiful secret	2.8	16	Begonia still	1.33
2	Dear translator	2.65	17	Father's identity	1.23
3	Monthly biography	2.41	18	Husband and wife	1.19
4	There is love on earth	2.25	19	Wonderful town	1.18
5	Stepfather home	1.85	20	Army One	1.14
6	Take the wrong car	1.82	21	Storm years	1.09
7	Distant engagement	1.82	22	Dad is a dragon	1.06
8	Valkyrie Zhao Zilong	1.81	23	A road to the north	1.06
9	Marshal Peng Dehuai	1.74	24	Princess	1.06
10	Decryption	1.71	25	Fall in love with you	1.06
11	Because love is happy	1.66	26	I am waiting for you	1.04

12	Small husband	1.63	27	Legend of dispensers	1.01
13	Hot neighbors	1.54	28	My husband husband	1
14	Happy together	1.45	29	You are my eyes	1
15	Woman’s sky	1.36			

Table 6: Ranking based on the index

	TV drama		TV drama
1	Monthly biography	16	Begonia still
2	Beautiful secret	17	Husband and wife
3	Dear translator	18	Wonderful town
4	Stepfather home	19	Father's identity
5	Take the wrong car	20	Army One
6	There is love on earth	21	Storm years
7	Distant engagement	22	Princess
8	Small husband	23	A road to the north
9	Marshal Peng Dehuai	24	Dad is a dragon
10	Decryption	25	Legend of dispensers
11	Valkyrie Zhao Zilong	26	Fall in love with you
12	Because love is happy	27	I am waiting for you
13	Hotneighbors	28	You are my eyes
14	Woman’s sky	29	My husband husband
15	Happy together		

Through the Table 3 and Table 4, it is easy to find that: totally speaking, the official rankings of the TV series are roughly the same as result from the formula; but the derived ranking differs from the official ranking in detail. The confidence of the results can be determined by calculating their mean and mean square error. Where the mean is $\bar{x} = 1.76$, while the mean square error $x_D = 0.67$. Where the mean represents the average errors of ranking for each drama compared with the official rankings, and the mean-square error represents the degree which the data deviates from the mean.

This error can be accepted by combining the two data. Because of the incomplete objectivity and incomplete accuracy of the network data. Then, the index can measure the popularity of TV drama.

4.3.3 The ideal group of the TV drama

Aiming to get the list of ideal production team, we must consider the collocation between each kind index. Because the ideal team is not the simple group of the most popular stars, the most popular types and other most popular indexes. For example, one star will not fit in all types of drama, and one team usually do well in specific types. Based on formula (24), a new matrix should be built up to describe the relation of collocation.

$$C_i = [c_{i1}\ \ c_{i2}\ \ c_{i3}\ \ \ldots\ \ c_{in}] \tag{25}$$

Where C_i is the i-th star's collocation matrix, which contains degree of collocation with all kinds of drama that the star has been in. And n is the number of drama that the star has been in.

When matching with the types of drama, the matching coefficient $\delta = C_i \times D_i$. D_i is the feature matrix of the drama type. The bigger the matching coefficient is, the better the couple match. Meanwhile, the production team is to type what the star is to type. Every index could be expressed by a specific matrix. Combined with formula (24) , we can get the ideal team as follows：

Table 7: The ideal team based on formula

Director	Sheng Kong
Starring	Liying Zhao, Yifeng Li, Yan Tang, Ge Hu
Types	Contemporary Urban
Screenwriter	Liping Wang, Yan Hai
Organization	Shandong Film and Television Media group

The most popular type of drama is contemporary urban, which can be considered as a basic index. Based on the basic index, combined with formula (24) and (25), we can find the most matched stars, director and screenwriter. Maybe they are not the top ones in their rankings, but the team working together will make a difference.

The most significant matter is resolving the collocation problem and distribute the weights to different indexes.

4.4 History searching and prediction model based on LDA

Aiming to figure out the most suitable drama for the audience and the drama that the audience most interested in according to the frequency the audience watching. This is based on probabilistic analysis of some samples in large data sets. The program rating of a television station is like to the viewing history of the audience, and the higher the rating score is, the higher the frequency of viewing is; the lower the rating score is, the lower the viewing frequency is. Followed by the basic analysis of the total number of home data, we will make predictive analysis.

Obtain the viewing history of the audience, by analyzing the frequency of the audience watching on certain types of drama, and we can infer the audience for some degree of interest in the drama. As the original data, the LDA algorithm[6] is introduced, and LDA is generated for the viewing records of the audience. The history records are classified according to the different types of frequencies, and the algorithm is self-learning to derive a reasonable list of push.

4.4.1 The principle of LDA algorithm

View history of all viewers is set D, topic is T, view history of each people is d_m, corresponding entry is $w_{m,n}$, corresponding topic is $z_{m,n}$, topic can be seen as a hidden variable. Topic-entries distribution, document-topics distribution, are the correlation statistics between $w_{m,n}$ and $z_{m,n}$, which can be considered as the state variables of Markov chain.

1) Probability φ_{ik} about w_i and z_k; θ_{km} about d_m and z_k:

$$\varphi_{ik} = \mathrm{P}(w_i|z_k) = \frac{C_{ik}^{VK}+\beta}{\sum_{i=1}^{V} C_{ik}^{VK}+V\beta} \tag{26}$$

$$\theta_{km} = \mathrm{P}(z_k|d_m) = \frac{C_{mk}^{MK}+\alpha}{\sum_{k=1}^{K} C_{mk}^{MK}+K\alpha} \tag{27}$$

Where C_{ik}^{VK} is the number w_i gives z_k, and $\sum_{i=1}^{V} C_{ik}^{VK}$ is the number of all the words giving to z_k, C_{mk}^{MK} is the number d_m gives z_k, $\sum_{k=1}^{K} C_{mk}^{MK}$ is the d_m gives to z_k which is the total number of words in d_m.

2) The probability of z_k showing in D:

$$\theta_k = \mathrm{P}(z_k) = \frac{\sum_{m=1}^{M} C_{mk}^{MK}+\alpha}{\sum_{k=1}^{K}\sum_{m=1}^{M} C_{mk}^{MK}+K\alpha} \tag{28}$$

Unlike other LDA model, the model in this paper don't need to be given the topic and it can be adaptive to all the topics searching. Based on Minimum of average topic similarity, the model will calculate the number of topics and catch what the audience are interested in.

3) Based on LDA model, calculated the value of primitive search $Q(u)$:

$$P(z_{kQ}) = P(Q(u)|z_k) = \prod_{i=1}^{L} P\,(w_i|z_k) = \prod_{i=1}^{L} \varphi_{ik} \tag{29}$$

$$P(z_{kQ}) = P(z_k|Q(u)) = \sum_{i=1}^{L} P(z_k|w_i)P(w_i|Q(u)) \propto \sum_{i=1}^{L} \frac{\varphi_{ik}\theta_k}{\sum_{k=1}^{K}\varphi_{ik}\theta_k} \tag{30}$$

In Formula (30), the probability of each word w_i showing in $Q(u)$ is $P(w_i|Q(u))$, which can be considered as $1\,/\,L$.

4) Relevance between expanded candidate and intention of searching:

Obtain the initial data and the interested topics, and calculated the relevance between them. After catching the intention of the audience, random word w_i is considered as the expanded candidate of initial search $Q(u)$, and the formula is:

$$P(w_i|Q(u)) = \sum_{k=1}^{K} (w_i|z_k) \times \lambda_k \times P(z_{kQ}) = \sum_{k=1}^{K} \varphi_{jk} \times \lambda_k \times P(z_{kQ}) \tag{31}$$

Where the λ_k is the weight of z_k.

5) Determining the weights of z_k:

Based on Cosine Calculation, we can calculate relevance between the other topics and the most possible topics, let it be λ_k. Put each topic as vector of words, and the probability that the word shows in the topics can be considered as weights. The relevance between the topics can be got from the vectors of the words, using the relevance between the vectors. In this paper, the Cosine Calculation can be the tool to find the correlation.

The most possible topic is z_t and the similarity with others topics z_k is:

$$\lambda_k = \frac{\sum_{i=1}^{V} P(w_i|z_t) \times P(w_i|z_k)}{\sqrt{\sum_{i=1}^{V} P(w_i|z_t)^2 \times \sqrt{\sum_{i=1}^{V} (w_i|z_k)^2}}} \tag{32}$$

$$\lambda_k = \frac{\sum_{i=1}^{V} \varphi_{it} \times \varphi_{ik}}{\sqrt{\sum_{i=1}^{V} {\varphi_{it}}^2 \times \sqrt{\sum_{i=1}^{V} {\varphi_{ik}}^2}}} \tag{33}$$

If the most probability are more than one, then we should calculate the degree of similarity between z_k and others. The maximum is best.

Finally, put the best λ_k into the formula (31), here we get $P(w_i|Q(u))$, and based on $P(w_i|Q(u))$ we can get a ranking of types. The top N types can be recommended to the audience.

For the model where the data from rating score, we can use the probability φ_{ik} and θ_{km} to reflect the average scores. And it works when put into the formula (33), which means this model meet the requirement of this question. And the verification will be given in next section.

4.4.2 Verification of the model

Aiming to prove the credibility of the model, build a new model 2 to measure the index in the model after work. The main process is that by calculating the mean and standard deviation of the error. About the error, we calculate the result from the index and search how many bad value which the type doesn't match with the ones interested well. Get the number of bad value, sum up the number of bad value after the model runs several and compare with the whole data printed. Then calculate the mean and standard deviation.

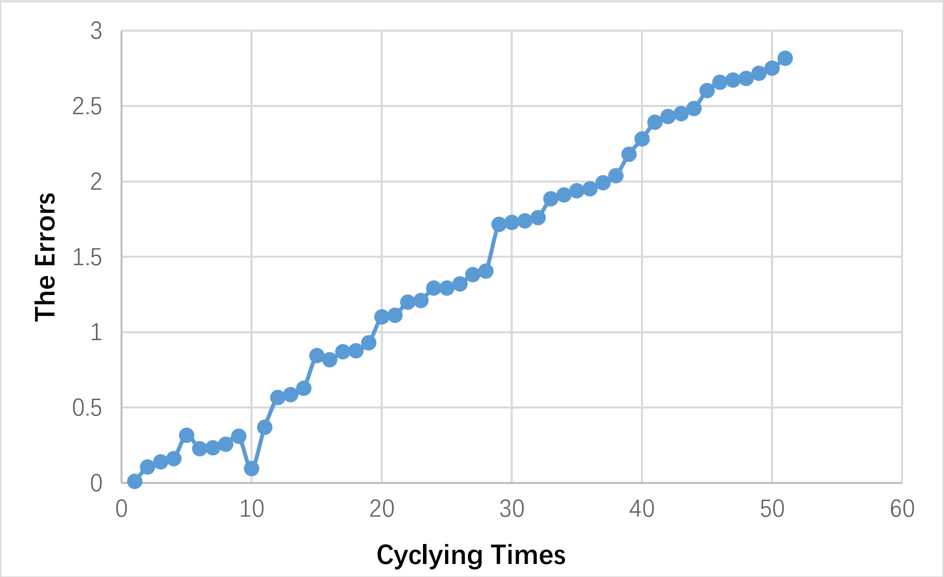

Figure 4: The errors with increase of the working times

By model 2, the mean and standard deviation are in an acceptable area. Prove the credibility is high enough to meet the requirement.

$$\bar{x}=3.4, \quad x_D = 2.01$$

Where the $\bar{x}$ is number of errors in 50 total data, and the x_D is the deviation value from the mean value. Finally, the credibility is between 89.28% and 97.22%, the average credibility is 93.2%, which means this model works well.

5 Weakness and Strength

Like any model, our model has its own strengths and weaknesses. Some of major points are presented below.

5.1 Strength

- Our models are based on the data from Internet; it is objective and have high universality. By testing on the data from reality, the results given by the models can fit well with the ranking from the official websites. According to the verification of the models, we can see that the error rates are under the acceptable areas, which means that the models can work well.
- The data are rich and consider questions from comprehensive ways, which means that if some factors change fiercely or some situations special, the models will be stable and still for a period.
- Simplifying the calculation by simplifying the models. In the LDA, using Cosine Calculation, we can easily get the best values and don't have to put long and big formula to find the answers.

5.2 Weakness

- For simplifying the models, we organized some unimportant factors. Those factors contribute little to the results which we have tested. But maybe they are important in other are.
- There is no completely official data, which may influence the result to some degree.

References

[1]Zhou Ya, The TOPSIS in the Multiple Attribute Decision Making, Wuhan University of Technology,2009

[2]Liu Sifeng, Cai Hua, Yang Yingjie, Cao Yin, The research progress of GRA, Institute of Gray System, Nanjing University of Aeronautics and Astronautics, 2013,8

[3]http://baike.baidu.com/link?url=9ddYSSeoySdQnQXoBi_vrUIIzSDidO6lMCCANuEWr3w9RUHTrAyfM14rHFIJ4CqefN7KMO7m6BX-HqeZGOlNp_

[4] http://www.cnblogs.com/haore147/p/3630002.html

[5]Hongping Zhao, Usage of Stepwise Regression based on Different econometrics software, School of Economics, Law and Politics, Nanjing Xiaozhuang University, 2007, 09

[6] http://blog.csdn.net/huagong_adu/article/details/7937616

Annex

- **Getting the data from Internet**

```
#coding:utf-8
import requests
import re,json

def get_stars(url):

    strs = '''<span class="rank_left_name" person-id="529">Liyin Zhao</span><span clas'''\
          '''s="rank_left_value"><b class="rlv_gray">9.0814</b>'''
    req = requests.get(url).text
    pattern = r"<span.*?person-id=\"\d*?\">(.*?)</span>.*?\">([\d,\.]*?)</b>"
    out = re.findall(pattern,req)
    for i in out:
        print i[0] + "," + i[1]  # print stars so that wo can covert the file to a csv format.
    return out                # return the list of stars.


def get_rank():

    url_1 = "http://www.xunyee.cn/rank-person-index-3.html"
    get_stars(url_1)
    length = []
    for i in range(2,35):
        url_2 = "http://www.xunyee.cn/rank-person-index-3-page-%d.html"%(i)
        length.append(get_stars(url_2))

    return length


# the stars list
stars = [u'Zhao Liying', u'Li Yifeng', u'Lay', u'Yang Zi', u'Ma Tianyu', u'Yang Yang', u'Hu Ge', u'William Chan',
    u'Liu Tao', u'Yang Mi', u'Victoria', u'Zheng Shuang', u'Wang Kai', u'Tang Yan', u'Ruby Lin', u'Liu Shishi',
    u'Guan Xiaotong', u'Wang Ziwen', u'Wallace Huo', u'Zhang Yishan', u'Zhangruoyun', u"Zhang Tian'ai", u'Di Ali Gerba',
    u'Joker', u'Cheney Chen', u'Fan Bingbing', u'Maggie Jiang', u'Zhang Han', u'Joe Chen', u'Gulnazar', u'Honglei Sun',
u'Jiang Xin', u'Wu Lei', u'Zhang Meng', u'Hawick Lau', u'Mark', u'Qin
Junjie', u'Juen-Kai Wang', u'Angela Baby',
    u'Tansongyun', u'Chenhe', u'Liu Yifei', u'YoonA', u'Song Joong
Ki', u'Yuan Wang', u'Tangyixin', u'Wu You', u'William Feng',
    u'Jiangjinfu', u'Through', u'Jin Dong', u'Liuhaoran', u'Li
Zhongshuo', u'Dongyu Zhou', u'Jackson Yi', u'Zhong Hanliang',
    u'Kan Kiyoko', u'Deng Chao', u'Luyi Zhang', u'Li Chen', u'Sun
Li', u'Guo Degang', u'Liu Yan', u'Lu Yi', u'Huang Lei',
    u'Zhangmingen', u'Luhan', u'Ju Jingyi', u'Cheng Yi', u'Ji Chang
Wook', u'Xiaozhan', u'Zheng Kai', u'Mao Zijun',
    u'Huang Xiaoming', u'Yu Hewei', u'Hai Qing', u'Luo Jin', u'Qi
Wei', u'Huang Bo', u'Li Qin', u'Wu Xiubo',
    u'Xinyi Zhang', u'Qing Jia', u'Huang Haibing', u'Yuan Shanshan',
u'Jia Nailiang', u'Du Chun', u'Cary Woodworth',
    u'Zu Feng', u'Baishu', u'Qiao xin2', u'Zhao Wei', u'Liyan Tong',
u'Yuan Hong', u'Chen Xiao', u'Maoxiaotong',
    u'Qiao Zhenyu', u'Ady Ann', u'Gao Yuanyuan', u'Yang Shuo', u'Chen
Xiang', u'Zheng Yin', u'Hye gyo Song', u'Nicky Wu',
    u'Wujiacheng', u'Chen yao1', u'Lee Jun-ki', u'Xiao Che', u'Zhang
Yi', u'Huyunhao', u'Joe Cheng', u'Gilbert air',
    u'Baoqiang Wang', u'Janine Chang', u'Jin Chen', u'For the',
u'Eddie Peng', u'Sheenah', u'Hongchen', u'Wang Ou',
    u'Faye Yu', u'Sun Yi Chau', u'Pets Ceng', u'Fuchengpeng', u'Jing
Bairan', u'Qiao Renliang', u'Show Luo', u'Wu Jing',
    u'Zhe Han Zhang', u'Handongjun', u'Liyitong', u'Alec Su',
u'Loura', u'Zhang Danfeng', u'Yan Ni', u'krystal',
    u'The white buildings', u'Guozifan', u'Houmengsha', u'Louis Koo',
u'Hubingqing', u'Park Shin Hye', u'Andy',
    u'Jimmy Lin', u'Pengchuyue', u'Rong Yang', u'Zifeng Zhhang',
u'Shuyaxin', u'Zhang Xinyu', u'Kris', u'Yangle',
    u'Yuanbingyan', u'Zhu Yawen', u'Maidina', u'Zhangxueying', u'Ng
Cheuk Hai', u'Kelsey', u'Kyle Cui', u'Xuhaiqiao',
    u'Happy', u'Qian Wu', u'Jay Chou', u'Wang Xiaochen', u'Li
Xiaoran', u'Liu Ye', u'Zhao Lei', u'Xu Doudou', u'Jiro Wang',
    u'Yanzidong', u'Ouyang Nana', u'Gao Yixiang', u'Benny Chan',
u'Song Jia', u'Jordan Chan', u'Bea Hayden', u'Michelle Chen',
    u'Yan Yi wide', u'Stephen Chow', u'Alyssa Chia', u'Ying Er',
u'Raymond Lam', u'Bosco Wong', u'Xiong Naijin', u'Hu Bing',
    u'Bing Shao', u'Angela Chang', u'Anita Yuen', u'Baijingting',
u'Vincent Chiao', u'Gillian Chung', u'JJ Lin', u'iu', u'Xu',
    u'Kenny', u'Charmaine Sheh', u'Angie Chiu', u'Tsung-Han Lee',
u'Kim Su Hyon', u'Zihan Chen', u'Yu-chi Chen', u'Ariel Lin',
u'Wang Yuexin', u'Du Haitao', u'Jiangzile', u'Chenruoxuan', u'Ma
Sichun', u'Pubaojian', u'Niujunfeng', u'Peter Ho',
    u'Gujiacheng',
]

def getFansAndPosts():

    pattern = r"<span
class=\"card_menNum\">([\d,\,]*?)</span>[\w\W]*?<span
class=\"card_infoNum\">([\d,\,]*?)</span>"
    for i in stars:
        url = "http://tieba.baidu.com/f?kw=%s"%(i)
        # print url
        req = requests.get(url).text
        result = re.findall(pattern,req)[0]
        # print results so that wo can covert the file to a csv
format.
        print result[0].replace(',','')+','+result[1].replace(',','')

def calc(ll):
    out = 0;
    for i in ll:
        out += int(i)
    return out/len(ll)

def getIndexAndMedia():

    for i in stars:
        try:

            get_media_url =
"http://index.so.com/index.php?a=soMediaJson&q=%s"%i
            media =
json.loads(requests.get(get_media_url).text)['data']['media'].values(
)[0].split('|')[-300:-1]
            get_index_url =
"http://index.so.com/index.php?a=soIndexJson&q=%s"%i
            index =
json.loads(requests.get(get_index_url).text)['data']['index'].values(
)[0].split('|')[-300:-1]
            # calculate the average num of Media Focus
            avg_media = calc(media)
            # calculate the average num of Index.
            avg_index = calc(index)


          # print stars so that wo can covert the file to a csv format.
            print str(avg_index) + "," + str(avg_media)
        except Exception,e:
            print i
            exit(0)


if __name__ == '__main__':

    getIndexAndMedia()
```

- **Getting the TV drama data**

```
#coding:utf-8

import requests
import re
import httplib
import md5
import urllib
import random
import json

def translate(q):
    appid = '20151113000005349'
    secretKey = 'osubCEzlGjzvw8qdQc41'


    httpClient = None
    myurl = '/api/trans/vip/translate'
    fromLang = 'zh'
    toLang = 'en'
    salt = random.randint(32768, 65536)

    sign = appid+q+str(salt)+secretKey
    m1 = md5.new()
    m1.update(sign)
    sign = m1.hexdigest()
    myurl =
myurl+'?appid='+appid+'&q='+urllib.quote(q)+'&from='+fromLang+'&to='+
toLang+'&salt='+str(salt)+'&sign='+sign
```

```
 try:
        httpClient = httplib.HTTPConnection('api.fanyi.baidu.com')
        httpClient.request('GET', myurl)

        #response HTTPResponse
        response = httpClient.getresponse()
        return json.loads(response.read())['trans_result'][0]['dst']
    except Exception, e:
        print e
    finally:
        if httpClient:
            httpClient.close()


tags = ['love','comedy','city','Suspense''Costume','idol','crime','history','war','Martial arts','Police bandit','Science Fiction']

def get_page(tag):
    url = "http://v.sogou.com/teleplay/list/style-%s+zone-内地.html"%(tag)
    con = requests.get(url).text
    return con

def find_vedio(context):
    # print context
    pattern = r'target=\"_blank\">(.*?)<\/a><\/div>'
    return re.findall(pattern, context)

def get_data(tags):
    out = []
    for tag in tags:
        data = {}
        vedios = find_vedio(get_page(tag))
        data[tag] = vedios
        out.append(data)
    return out

if __name__ == '__main__':
    data = get_data(tags)
    for tag in data:
        key = tag.keys()[0]
        # print key
```

```
vedios = tag.values()[0]
    for vedio in vedios:
        # print data so that can be covertd to csv format.
        print translate(key)+','+translate(vedio.encode('utf-8'))
```

- **The drama data from Internet (Part)**

urban	Frog Prince
Crime	Conquer the first
gangster	Police hat
History	Jinxiu Le Weiyang
Suspense	Fire blue blade
science fiction	I love
Crime	Mekong major
Love	Lan Ling Princess
science fiction	Micro power
science fiction	Ultra juvenile password Wang Junkai cut
Love	Love shuttle between the 2 of the moon under the moonlight
Martial arts	Shushan olgame JX legend
Crime	Flower bloom
comedy	I’m not a monster.
Ancient costume	Qingyun documentary records
urban	It is hard to serve two masters
History	Mi months pass
History	Legend of Lu Zhen
Love	Let’s fall in love.
science fiction	Qingyun documentary records
gangster	Pioneer of law 3
science fiction	The last second seasons
Crime	Anti Terror team
Suspense	Decrypt
Love	If the snail has love
Love	Custom happiness
Suspense	Exploration spirit file
Love	Basketball fire

Crime	Police hat
urban	Shanshan coming
idol	A return of 1
urban	weaning
urban	A wife's lies
idol	Meet Wang Lichuan
Suspense	A left eye.
Suspense	If the snail has love
gangster	Pioneer of law 1
Ancient costume	The Legendary Swordsman Jackie Lui version
Martial arts	The The Heaven Sword and Dragon Saber version of Steve Ma
urban	The new era of bestie
Ancient costume	The lower part of the new living Buddha Ji Gong
gangster	The Mysteries of Love
Martial arts	Bad people draw the rivers and lakes
Crime	tyrannical official
comedy	Hospital laughter 2
Ancient costume	Lan Ling Princess
urban	So Young
Suspense	Steep cliff
Martial arts	Heroes day
comedy	Love apartment 2
Crime	Criminal Police
Ancient costume	Happy spy
gangster	Every Move You Make
gangster	silent
urban	A Beautiful Daughter-in-law Era
urban	Best couple
comedy	Love apartment 3
Martial arts	The legend of the Condor Heroes Hu Ge version
Martial arts	Zhu Xian Zhi Yun
Ancient costume	The King of Yesterday and Tomorrow
science fiction	Schrodinger's Cat
Suspense	Ten deadly sins