Credit Scoring Specification

V 2.1

1 Overview

Credit Score Algorithm (CS) is a system for quantitative assessment of personal credit risk. It obtains credit scores through quantitative calculation of personal information, reflecting the credit status of individual customers. Credit score systems are widely used in financial services such as personal credit, credit cards, insurance claims, etc., to provide quantitative support for the definition, analysis, evaluation, and optimization of credit policies .

2 Architecture

The product builds different models based on the personal information loaded by the user or other validated resources. It analyzes user's data, calculates scores for multiple dimensions and gives user a total score through mathematical statistics and other algorithms. The general process of scoring is shown in the figures below:

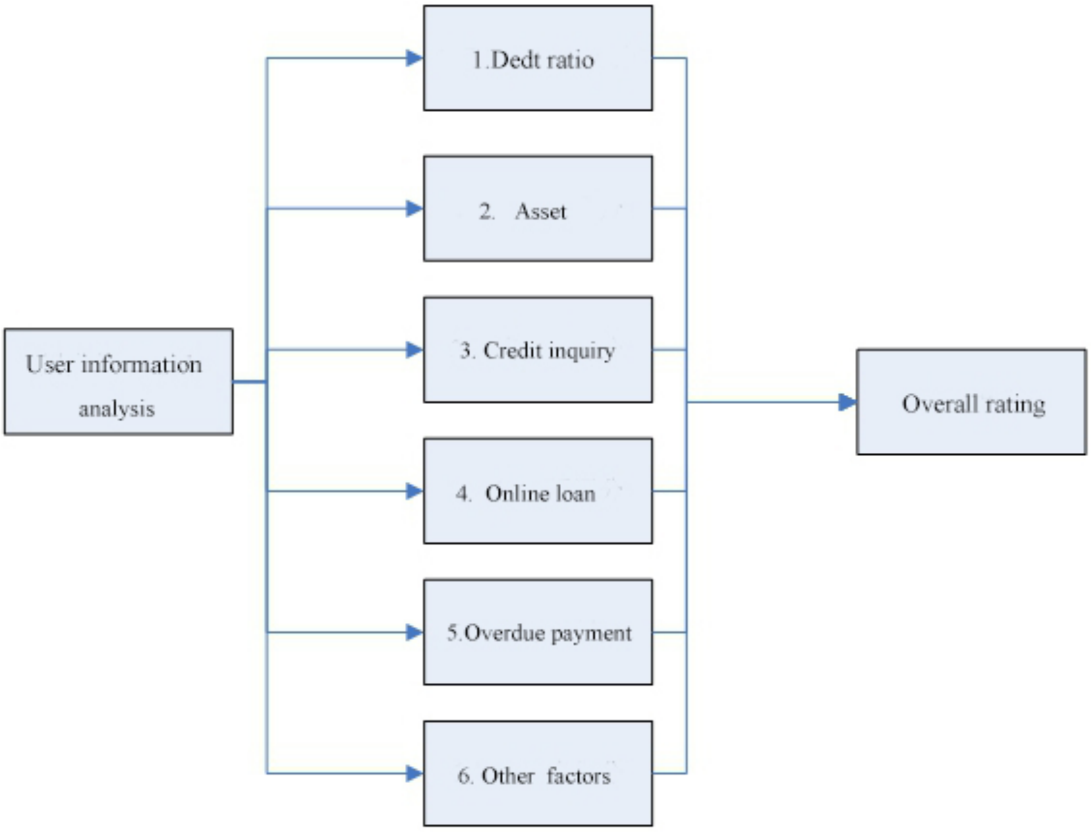

Figure1. synthetical scoring process of personal credit qualification.

The evaluation module of personal credit qualification includes six modules, Debt ratio, Asset, Credit inquiry, Online loan, Overdue payment and other factors. CS will give an overall rating by having calculated each individual module score.

Personal credit score reflects the individual's credit status in a score range, generally speaking the higher the score , the lower the risk or the better the credit. The overall rating could provide a reference to the decision makers.

3 Specification

3.1 User Info

The basis of the CS algorithm is the user's basic information.

Table 1. User Information

Prarameter	**Type**	**Description**	**Mandatory**	**Remarks**
TotalCredit	F loat	Credit card used limit	Yes	Debt ratio
TotalR epayment	Float	Monthly repayment amount of all non-mortgage loans	Yes, default 0	
PublicFund	Float	Provident Fund	Yes, default 0	
Salary	Float	Reported income	Yes, default 0	
HaveHouse	Bool	House property	Yes	Asset
HaveCar	Bool	Vehicle property	Yes	
HaveID	Bool	Life insurance	Yes	
HaveSS	Bool	Social security	Yes	
MonQueryNumber	int	Number of queries in a month	Yes	Credit inquiry
ThMon QueryNumber	int	Number of queries in three months	Yes	
SixMonQueryNumber	int	Number of queries in half a year	Yes	
Net L oan Number	I nt	Number of online loans	Yes, default 0	Online loan
CardInfo	**A rray**	**All card information arrays, the contents of the arrays please refer to Table 2;**	Yes	Overdue payment
DeadAccount	Bool	Are there bad or frozen debts	Yes, default false	Other facets
Warrantor	Bool	If the guarantor compensates	Yes, default false	
Asset	Bool	If there is asset disposal	Yes, default false	
PublicInfo	Bool	Is there any public information abnormal	Yes, default false	
CreditRecord	Bool	Are there any abnormalities in non-credit overdue records?	Yes, default false	

Table 2: Contents of Card Information Array

parameter	Type	Description	Mandatory
TotallAccOverdueNumber	Bool	Currently overdue (yes/no)	Yes
Six MonOverdueNumber	Bool	Accumulated 6 months overdue (yes/ no)	Yes
ThMonOverdueNumber	Bool	Overdue for 3 consecutive months (yes/no)	Yes
TwoYearOverdueNumber1	Bool	Less than 3 overdue times within 2 years	Yes
TwoYearOverdueNumber2	Bool	The number of past dues within 2 years is greater than or equal to 3	Yes

3.2 Single Module Score

The overall rating is a synthetic score of all 6 single module scores and each single module score is calculated as the followings

3.2.1 Debt ratio score

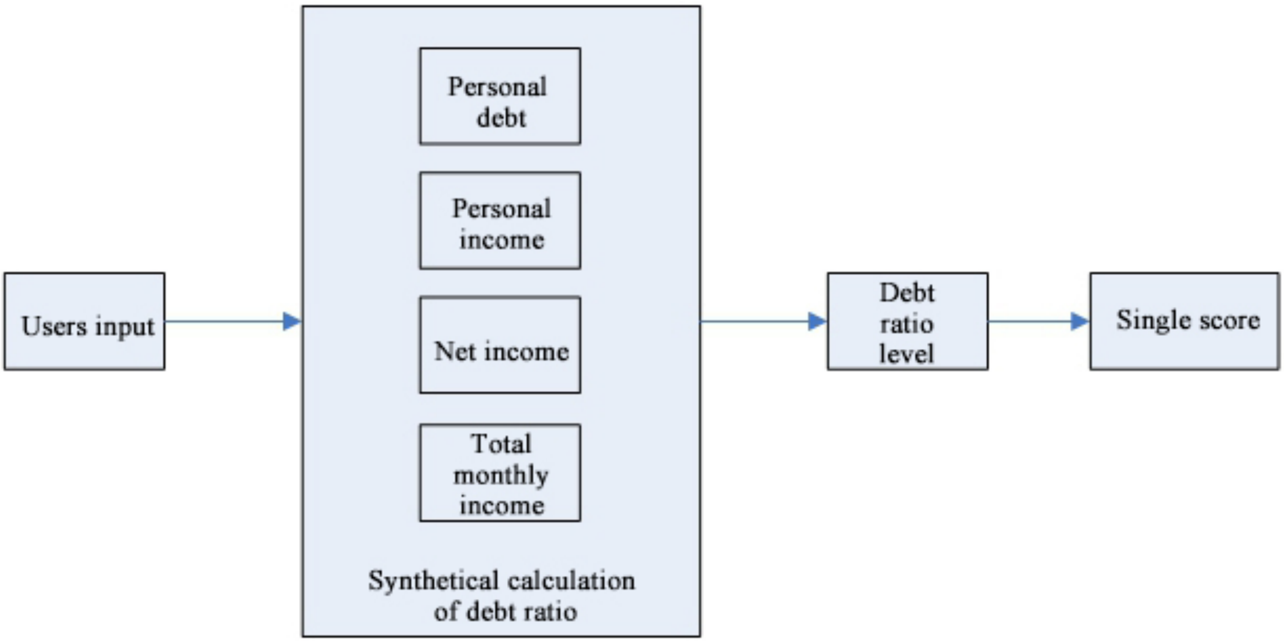

Figure 2. Debt ratio score calculation model

As shown in the figure above, the input parameters of the debt ratio model includes: personal debt, personal income, net income and total monthly income.The user's debt score is given out according to the sectors above.

*Formula : Debt ratio= personal debt / personal income *100%*

** Personal Debt = used credit card limit * 10 % + monthly repayment amount of all non-mortgage loans ;*

** Personal income = income 1 > income 2 ? Income 1; (select the maximum income)*

** Check-in income = Provident Fund / 0.14;*

** Reported income = fill in by the user;*

Debt ratio score output: {5, 10 }

Debt ratio~[0~100%]:   Output: 10;

Debt ratio ~ (100%~200%] : Output: 8;

Debt ratio ~ (200%~300%] : Output: 6;

Debt ratio ~ [300% or more] : Output: 5;

3.2.2 Asset score

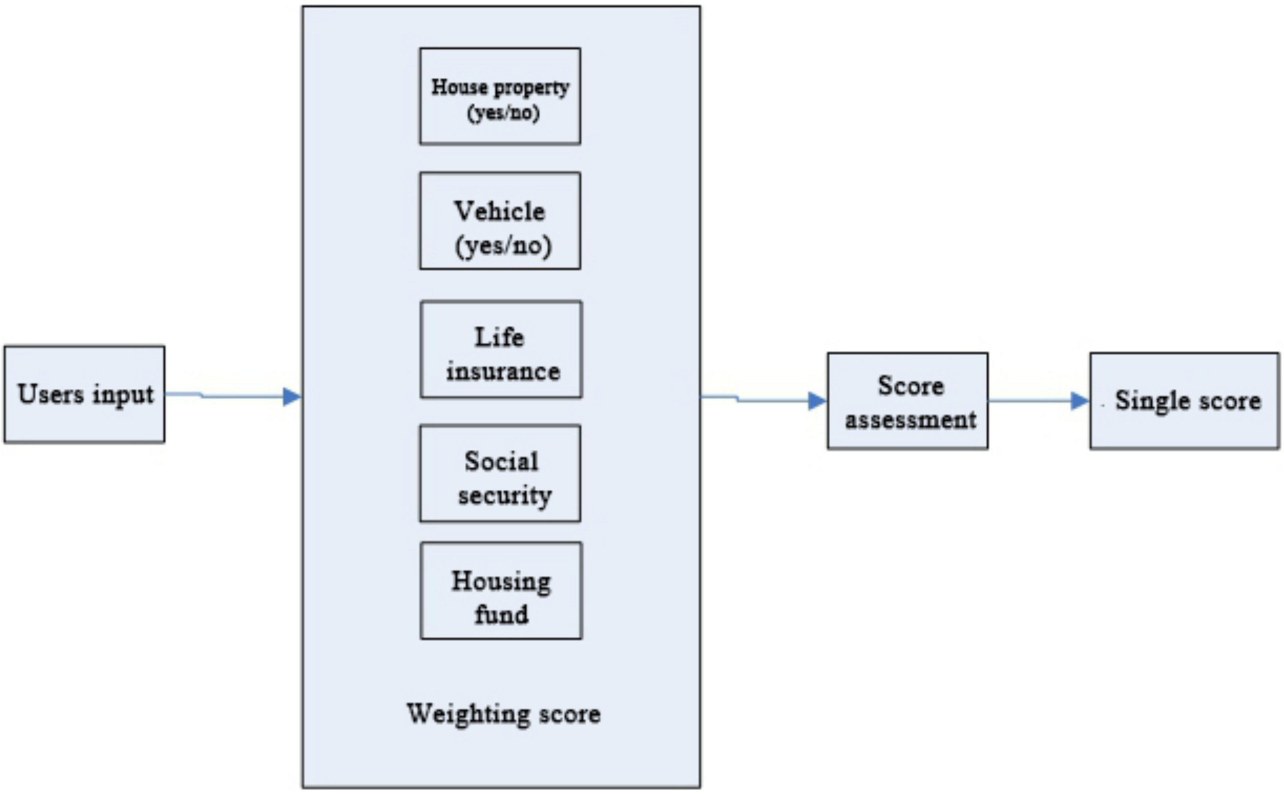

Figure 3. Asset score calculation model

As shown in the figure, the asset score will be calculated based on the user's property information.

House; => Output: 2;

Vehicle; => Output: 2;

Life insurance; => Output: 2;

Reported income > 0; => Output: 1;

Social Security; => Output: 1;

Provident fund> 0; => Output: 2;

Asset score output : {0, 10 }

Output = House + Vehicle + Life insurance + Reported income+ Social security + provident fund;

3.2.3 Credit inquiry score

The Credit inquiry model calculates the score based on the parameters entered by the user, including the number of queries within 1 month, the number of queries within 3 months, and the number of queries within 6 months. The calculation model is shown in the following figure:

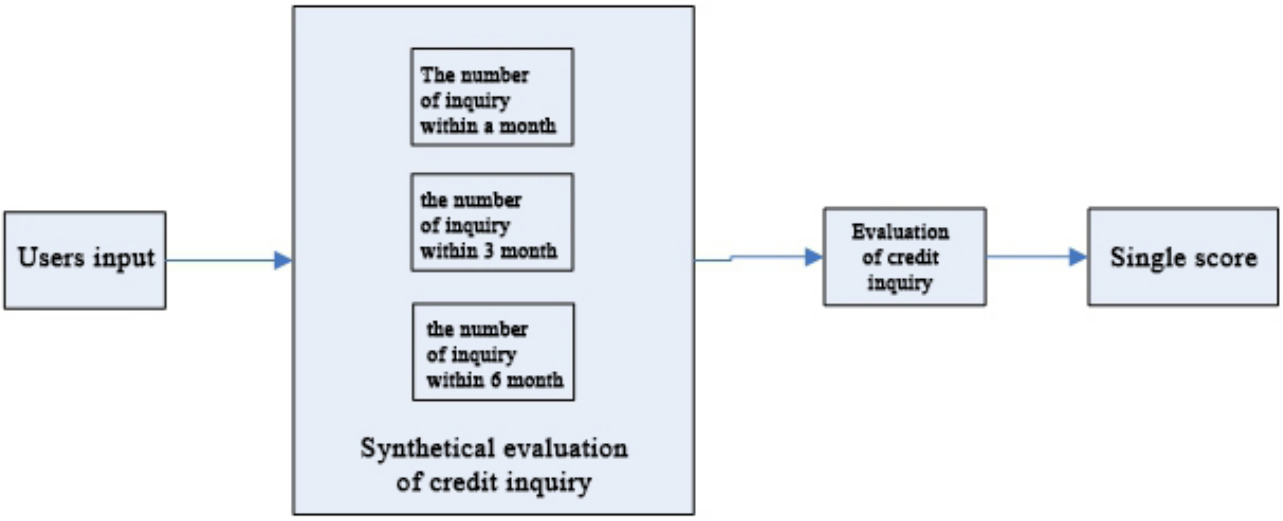
Users input
The number of inquiry within a month
the number of inquiry within 3 month
the number of inquiry within 6 month
Synthetical evaluation of credit inquiry
Evaluation of credit inquiry
Single score

Figure 4. Query analysis model

Input parameters:

X = the number of queries in 1 month;

Y = number of queries in 3 months;

Z = Number of queries in 6 months;

(X <4) && (Y<8) && (Z<15); => Output: 10:

(X <4) && (Y<8); => Output: 4;

(Y<8) && (Z<15); => Output: 2;

(X <4) && (Z<15); => Output: 3:

Else => Output: 0;

3.2.4 Online loan score

In addition to the user's fixed assets, it is also necessary to assess user's online loan information. The online loan assessment is relatively simple.

Number of online loans: Net Loan

NetLoan = 0; => Output: 10;

0 < NetLoan <4; => Output: 8;

4 < = NetLoan < = 6; => Output: 5;

NetLoan >= 7; => Output: 3 ;

3.2.5 Overdue payment score

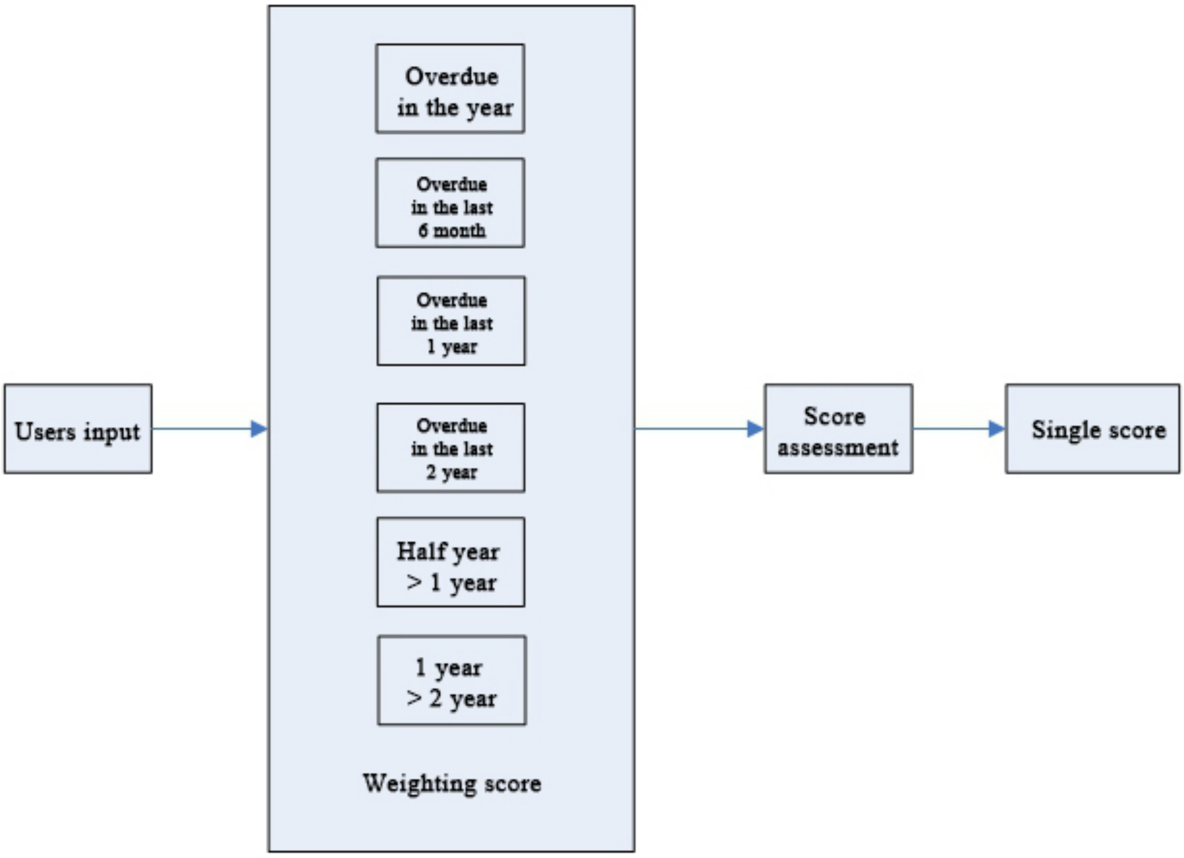

Figure 5. Overdue analysis model

As shown in the figure above, 2 rounds of overdue scoring are required, and then the results of each round are valued twice to output the final scores.

After 2 rounds of scoring for overdue payments, select the lower score as output;

1st round

Input:

Currently overdue; 1 /2(3 , 4 , 5……)/0

Recent 6 months; 1/2(3 , 4 , 5……)/0

Recent 12 months; 1/2(3 , 4 , 5……)/0

Recent 24 months; 1/2(3 , 4 , 5……)/0

(6, 12] months; 1/2(3 , 4 , 5……)/0

(12, 24] months; 1/2(3 , 4 , 5……))/0

Current overdue > = 2; => Output: 50;

Half a year overdue > 2; => Output: 50;

Overdue within one year >= 2; => Output: 50;

Overdue within two years >= 4; => Output: 50;

Current overdue = 1; => Output: 4;

Overdue within half a year = 2; => Output: 0;

Overdue within half a year = 1; => Output: 4;

Overdue (6, 12] months; <= 2; => Output: 3;

Overdue (12, 24] months; <= 3; => Output: 2;

2nd round :

Input:

Total number of overdue

Statistics table number

Current overdue

Total number of overdue > 6; => Output: 65;

Statistics table number >= 3; => Output: 70;

Current overdue = 1; => Output: 4;

Current overdue > 1; => Output: 50;

(Statistics table number > = 3) && (currently overdue = 1); => Output: 60;

(Total number of overdue > 6) && (current overdue = 1); => Output: 60;

(Statistics table number > = 3) && (total number of overdue > 6); => Output: 55;

V 2.0

Conditions:

A: Currently overdue (yes/no)

C: Accumulated overdue in 6 months (yes/no)

D : Overdue in 3 consecutive months (yes/no)

E: The number of overdue within 2 years < 3

F: The number of overdue within 2 years >= 3

! A && ! C && ! D && ! E && ! F; => Output: 10;

A && ! C && ! D && ! E && ! F; => Output: 4;

A && C; => Output: 60;

A && D; => Output: 60;

A && E; => Output: 2;

A && F: ; => Output: 0;

! A && C && ! D && ! E && ! F; => Output: 65;

! A && ! C && D && ! E && ! F; => Output: 65;

! A && ! C && !D && E && ! F; => Output: 7;

! A && ! C && ! D && ! E && F; => Output: 5;

Else; ; => Output: 60;

Final Overdue score:

The above is the rule for a single card scoring. The final score for multiple card scoring is as follows.

1. If the scores of multiple cards >= 60, the final score is the lowest one among the ones which higher than 60

For example: {65, 60, 5, 7, 10}; => Output: 60

2. If 2 or more cards meet condition A; => Output: 65;
3. If 2 or 3 cars meet condition E and don't meet any other conditions; => Output: 3;
4. If 3 or more cards meet condition E and don't meet any other conditions; => Output: 0;
5. If 2 cards meet condition F and don't meet any other conditions; => Output 0;

6. If 1 card meets condition E and don’t meet any other conditions && another 1 card meets condition F; => Output 65;

4.2.6 Other factors

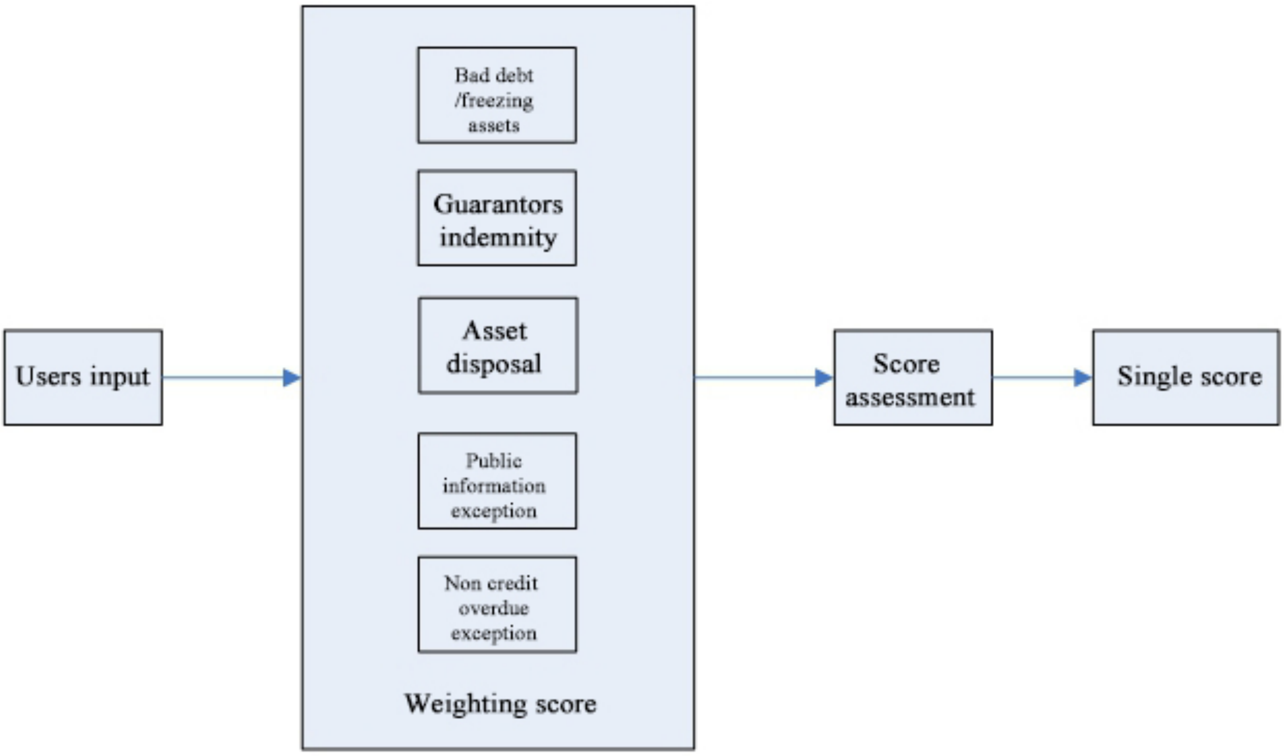

Figure 6. Other factors model

Score of Other factors is either ’50’ or ’60’ if any of the following rule is met.

Rules:

There is bad debt or frozen; => Output: 50;

Guarantor compensation; =>Output: 50;

Asset disposal; => Output: 50;

Public Info exception; => 50;

Non-overdue exceptional records => 60;

4.3 Overall Rating

Overall rating model calculates a synthetic score on the individual's credit according to the above six models. It’s based on the result of big data and statistics.

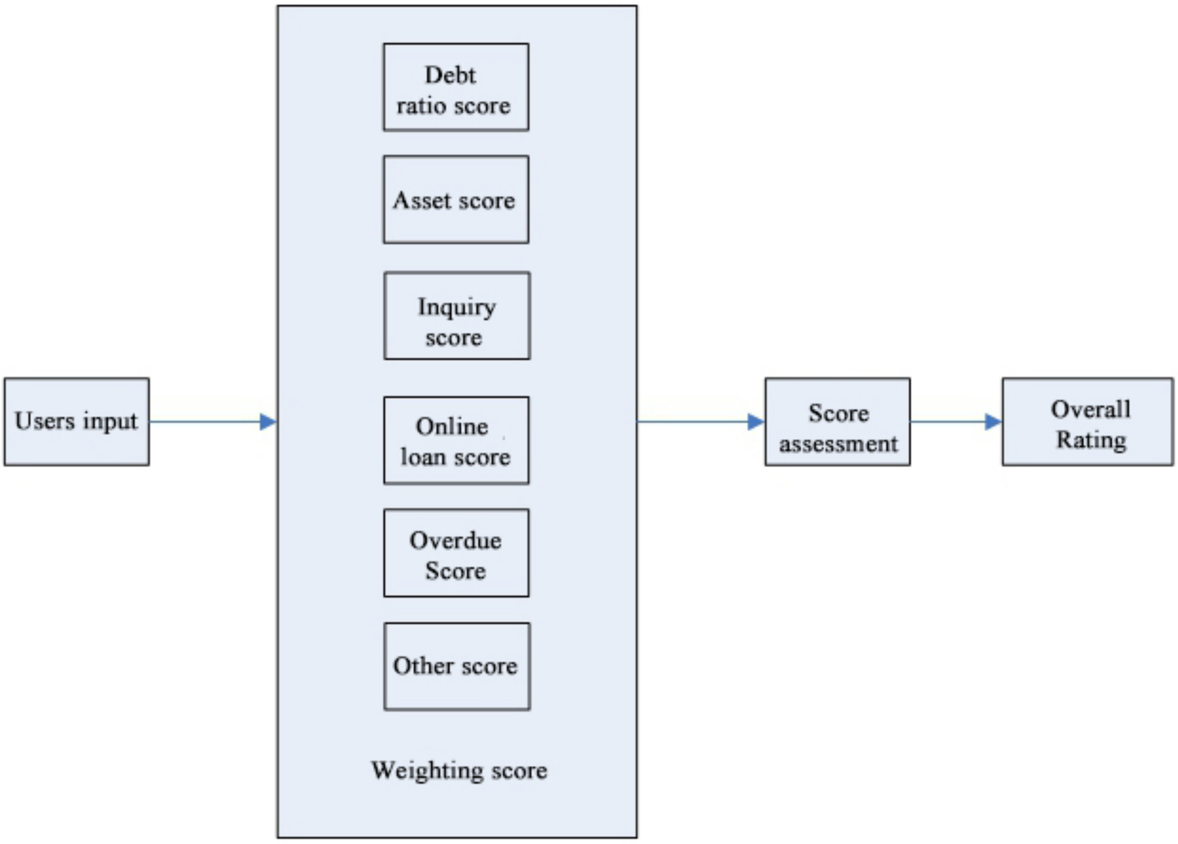

Figure 7. Overall rating model

Overall rating rules:

As long as Overdue score or Other factor score >= 50; then Overall rating = Overdue score || Other factor score;

Else, Overall rating = 50 + Debt ration score + Asset score + Credit inquiry score + Online loan score + Overdue score;

Then Credit Scoring (CS) will label user credit qualification as:

Excellent (90-100);

Good (75-89);

Average (60-74);

Poor (50-59);

e.g. Return of a good qualification:

```
{
    "TotalScore": 80,        /* Overall rating */
    "UnitSocre":             /* Single item score */
    {
      "LoadFactor": 8,       /* Debt ratio score */
      " Asset ":6,           /* Asset Score */
      " Query ":5,            /* Inquiry score */
      "Net L oan": 3,        /* net loan score */
      " Overdue ": 6,         /* Overdue score */
      "Other":0              /* Other factors score */
    }
}
```