

For office use only	Team Control Number	For office use only
T1 _____	8874	T1 _____
T2 _____		T2 _____
T3 _____	Problem Chosen	T3 _____
T4 _____	A	T4 _____

---

## A New Era for Ranking

(Shanghai, China - November 18, 2018) After days of development, FUNROLLER CO. Ltd., a startup composed of 4 ordinary Shanghai high school students, introduced a revolutionary roller coaster ranking app, FUN ROLLER, today. Not only is it an extraordinary app that provides users with detailed, personalized ratings and rankings, but also users will behold themselves to a variety of functional tools— this single app blends together multiple services such as navigation, ticketing, and most importantly, the ranking of roller coasters.

“FUN ROLLER is an innovation that will transform the concept of ranking and rating.” As R&D executive said, “FUN ROLLER CO. Ltd. continuously provides our users with new, optimized solutions in rankings and ratings.”

One function that sets this app apart from other ranking apps is its unique algorithms. The most common type of ranking system that is available on market is almost fully or partially subjective (Examples include user rating). In rare cases which the ranking is objective, the scope of ranking may be too parochial, as these ranks only focus on one detail of the roller coasters (examples include the ranking of speed and height of roller coasters).

Yet the algorithm behind the FUN ROLLER app is much more— it is an objective algorithm that achieves a state of balance by utilizing two concepts, entropy weight and a logarithmic fuzzy preference programming methodology. Overall, the algorithm is built on objective statistics of each rollercoaster — we considered multiple factors that influence the experience of rollercoaster rides and weighed them according to its importance in determining the experience. Yet the weighing of each factor is built on results of the company’s preliminary survey, therefore genuinely reflecting upon potential users’ priorities and needs.

Most important and unique of all, this logarithm is able to achieve a considerable level of customization to meet each individual user’s needs. As users create their accounts, they are required to fill out a survey regarding their preference for rollercoasters. They will be asked which factors they value the most in determining the rating of a roller coaster ride. As they filled out the survey, the results will be automatically analyzed and stored into the database. The weighing of factors in our algorithm will change in accord with the preferences they chose in the survey section, and the App will automatically generate a ranking of worldwide rollercoasters that suits the users’ preferences.

Besides the ranking, there is more to discover. The powerful yet easy-to-use Artificial Intelligence bilateral feedback system will aid anyone in finding the right rollercoaster for them. By simply talking to the app, users can get a list of roller coasters that they may be interested in. Edubank — the “shopping cart” and collection folder — will allow users to buy tickets within few clicks on the screen. The map section allows users to navigate a route to the desired roller coaster site or to find rollercoasters around them. These brief but practical tools will

#8874

guide users from searching for a rollercoaster to ticketing to transportation, and to finally rate their experiences in our app. These tools will make anyone's trip to their dream rollercoasters easier and more accessible.

This revolutionary logarithm and app will be live on the App Store in the following weeks. It would start a new era for ranking as the innovative approaches including personalization, AI feedback, and accurate delivery is first to be used in rankings. FUNROLLER CO. Ltd. will continue to contribute more tools of ranking a variety of products in the near future.

# How to evaluate various kinds of roller coaster for customers?

## Abstract

In this report, a mathematical model of roller coaster ranking system has been built. The ranking system is considered by three general factors that influences the overall roller coaster experience into three categories — excitement, safety, and innovativeness. Each category has a percentage weighing into the final rating. For each category, the score is a result of weighing of sub-categories factors (including height, speed, length, vertical angle, construction, type, open year, duration, inversions number, G-force etc.).

The weighing of each sub factor was calculated by incorporating both elements of objectivity and subjectivity. In order to balance these two elements, two concepts, entropy weight and fuzzy logarithm, were utilized for mathematical modelling. By conducting a survey, a general idea about how the potential users value each factor in determining the experience of a roller coaster ride was acquired, and one set of weighing values were generated for the double-layer fuzzy analysis. The other set of weighing values were calculated through entropy weight analysis. After that, all these weighing values were combined together to form the final weighing value of each factor. As a result, this amalgamation allowed the ranking system to achieve a balance between subjectivity and objectivity.

The ranking system was compared with two popular online rankings websites.

Initially the comparison results show much difference. The commonly recognized top rollercoasters in other websites were found to be out of the top five of the ranking. In an effort to discover the reason for this deviation, the algorithm was re-examined and found that when normalizing the data of roller coasters, the upper and lower limits were set floor for all kinds of values, in order to filter anomalies. However, some aspects of that specific roller coaster broke through the top limit of values. This caused the rating of that roller coaster to drop considerably. The ranking system was modified by cancelling the upper and lower limits, and then it shows much better consistency with online rankings.

The process of creating the UI is where originality and creativity sparkled. In the app's UI, elements of AI were incorporated for accurate delivery and customization. The algorithms behind the app can change accordingly to user preference, which was shown in a mandatory survey during registration and user viewing history within the app. so that the ranking would change accurately, Therefore, the ranking could change accurately to satisfy user's specific needs. Furthermore, an Artificial Intelligence bilateral feedback system, a "shopping cart" named Edubank section, and a navigation function were envisioned and demonstrated in the app.

The new ranking system combining with the customized UI could be a user-friendly app to give fans of roller coaster a new tool to find a roller coaster and improve the experience.

Keywords: Entropy weight / Logarithmic Fuzzy Preference Programming (LFPP) / Kendall tau distance / Roller coaster

# Contents

1. Introduction-----	5
2. Restatement-----	5
3. Assumptions-----	6
4. Symbol and justifications-----	6
5. Data preprocessing-----	6
5.1 filling of v-----	7
5.2 filling of G-force-----	8
5.3 filling of other data-----	9
5.4 error correcting-----	10
6. the new reference system-----	10
6.1 establishment of the model-----	10
6.1.1 Logarithmic Fuzzy Preference Programming (LFPP) -----	13
6.1.2 entropy weight model-----	14
6.1.3 the final comprehensive evaluation system -----	15
6.1.4 Kendall's tau distance model -----	15
6.2 solving the model-----	16
6.2.1 Data processing in each factor-----	19
6.3 simulation and conclusion-----	19
6.4 sensitive analysis-----	21
7. compare and improve our ranking list-----	22
7.1 The exist authoritative roller coaster ranking list-----	22
7.2 The geographic reason to cause the differentiation over the ranking list-----	22
7.3 Consistency between head and tail-----	23
7.4 Optimization about the limitations of minimum membership function-----	23
8. run roller (application) -----	24
9. Conclusion-----	33
10. Reference-----	33
11. Appendix -----	34

## 1. Introduction

A roller coaster is a type of amusement ride that employs a form of elevated railroad track designed with tight turns, steep slopes, and sometimes inversions. People ride along the track in open cars, and the rides are often found in amusement parks and theme parks around the world. The track in a coaster design does not necessarily have to be a complete circuit, as shuttle roller coaster demonstrate. Most roller coasters have multiple cars in which passengers sit and are restrained.

In the amusement park, the roller coaster will usually be the most popular item for the travelers to try. Roller coaster is so exciting that can attract a lot of people. Some amusement parks even become very famous because of the roller coaster inside, like the Cedar Point amusement park in Ohio state, America. Because of that, there are a lot of website and travel blogger spend time to create some different evaluation systems for the different roller coasters. There are usually more than 10 indexes for a professional evaluation system to evaluate a roller coaster. However, because the traveler and even the local people was unfamiliar to the various kinds of evaluation indexes of a single roller coaster and most of them only care about the safety, excitement and innovation, most website will only give a ranking list on safety, excitement and innovation, and the three main indexes also include some small indexes like the material to make the roller roaster and age of it. Some algorithm will be used to give a final ranking list of the roller coasters all over the world.

Also, the evaluation system of the roller coaster should be global and fit to all people around the world. That is to say, a complete and professional roller coaster should include the information of the location and name of the roller coasters. And the user of the evaluation system will depend on the ranking list of the safety, excitement, innovation and the location of the roller coaster to choose their favorite roller coaster and try it.

## 2. Restatement

There are already some websites make the evaluation systems for the roller coaster, however, they are not so professional as we expect. The ranking list of that system depends on the subjective inputs like the “excitement” or “experience” score of an “expert” rider to measure “thrill”. The result of it is the ranking list will misadvise the users.

To solve this problem, we will make a new evaluation system and give a new ranking list depends on the indexes we already get. The indexes include name, park, specific region, construction, type, status, date opened, height, speed, length, number of inversions, drop, duration, G force and vertical angle. Some kinds of roller coaster are not included in the evaluation system.

We are asked to make a new ranking list depends on the indexes we mentioned before, after that, we will get a make “Top 10 Roller Coasters in the world” list depend on our own algorithm. After that, we need to make a user-friendly application to help users to find a roller coaster

more precisely. In the end, we need to write a one-page non-technical news release describing your new algorithm, results and application.

### 3. Assumptions

Assumption (1): All the data in the chart is correct and didn't change for couple years.

Explanation (1): The fact is the technical parameter of a roller coaster will change each year, and that will bring the huge impact to the ranking list, we need to avoid that.

Assumption (2): There is no children / family roller coaster in the data we already have.

Explanation (2): The problem asks us to finish the problem without the children / family roller coasters, it's hard for us to determine whether the roller coaster belongs to the children / family coaster or not depend on its name or other indexes that we already have.

Assumption (3): The evaluation system of the online ranking list will not change in two years

Explanation (3): We need to compare the ranking list for the same website of two years, but the fact is that some website will change their evaluation system every year, we need to avoid this problem.

### 4. Symbols and justifications

Symbols	Justifications	units
$g$	acceleration of gravity	$m/s^2$
$v$	velocity	$m/s$
$r$	radius	$m$
$A$	Symbol of the matrix	/
$\tilde{A}$	Fuzzy matrix	/
$a_{ij}$	element	/
$\tilde{a}_{ij}$	element in fuzzy matrix	/
$\lambda$	minimum degree of membership	/
$\delta_{ij}$	non-error parameter	/
$\eta_{ij}$	non-error parameter	/
$\beta$	weight	/

### 5. Data preprocessing

We already get some data from the committee. However, the data are not complete at all. The loss of data happened in the indexes of Droop (feet), Duration (min: sec), G Force and vertical angle (also happened in length, speed and height, but they are not very common). Because of this, we need to do three main steps in the data preprocessing. The first step is to fill the blanks

and complete the data, the second step is to handle the wrong data and the final step is to define number for some indexes that used in the data.

45.0	2788.8	YES	10		01:32		
65.2	1476.4	YES	2		01:00	4	
45.0	1050.0	YES	3				97
62.0	2956.0	YES	6		02:47		
67.0	3828.0	YES	6	170.0	03:10	3.7	
51.6	2358.9	NO	0			4.3	
52.8	2879.8	YES	10				
66.0	4650.0	NO	0	147.0	02:23		55
55.9	3937.0	NO	0	40.0	02:10		
55.0	2900.0	YES	2	90.0	02:00		
50.1	2877.0	NO	0	87.3	03:00		

(Table.1 Examples of the loss of data)

Item	Height	Speed	Length	Drop	Duration	G-force	Verticle-angle
Number of loss	1 (line 124)	4	5	158	77	218	209

(Table.2 Kind of data lost and number of it)

For the first step, we need to fill the blanks and complete the data.

### Filling of $v$

After doing some calculations between the speed, length and duration, we find that the speed in the data is not the average speed, instead, it is the top speed of the roller coaster. That is to say, we cannot use two of these three values (speed, length and duration) to get the third one.

Also, we get that the height have some relationship with the speed. However, because the speed we have is the top speed, we cannot get a specific function to convey that, so we still cannot use the height and speed to fill the blanks and complete the data.

Now, we will use several steps to fill the data that disappear in the chart. The first step is to complete the value of drop  $h$ , which had been lost for 158 times. Depend on the principle of conservation of energy, we can get that:

$$\frac{1}{2}mv^2 = (mg \sin \alpha - \mu mg \cos \alpha - F) \times \frac{h}{\sin \alpha}$$

Because the friction and windage are very tiny in the roller coaster (roller coaster uses the rolling resistance and the speed of the roller coasters are quite low), and most of the roller coasters are motor less, they go up and down depend on the gravitation. For that, we can regard the speed at the bottom point as the top speed of the roller coaster and ignore the friction and windage. After that, we can get that:

$$v = \sqrt{2gh}$$

Now, if we don't have the value of  $v$  but we have the value of  $h$ , we can get the value of  $v$  from that equation.

### Filling of *G-force*

The next step to complete the value is to complete the value of G-force. The gravitation in the earth is very similar. However, if we do some acceleration or move around the circle, the gravitation will change a lot. The change of the gravitation has two main kinds, one is weightlessness and the other is overload. For the roller coaster that move around the circle, the overload is quite common.

Because different drop will make different top speed for the roller coaster and the roller coaster go up and down. As a result, the top speed point will generate the centripetal force when the roller coaster moves around the curve. And the centripetal, force is can be represented by:

$$F = m \frac{v^2}{r}$$

The value of F is the G-force in the chart. Now, if we assume that M is the G-force in the data, then we can get that:

$$m \frac{v^2}{r} + mg = Mmg$$

$$\frac{v^2}{r} + g = Mg$$

$$r = \frac{v^2}{(M-1)g}$$

Because we still have some data about the G-force and top speed, so, we can calculate the value of radius. After calculating the value of radius, we can combine radius and height together and do the linear regression by the existing algorithms. From the data we already have and do the linear regression, we can get that:  $r = 2.4652h - 189.01$ , it's a linear function. Then, we can change the r to h, we can get that:

$$m \frac{v^2}{2.4652h - 189.01} + mg = Mmg$$

$$\frac{v^2}{2.4652h - 189.01} + g = Mg$$

$$M = \frac{\frac{v^2}{2.4652h - 189.01} + g}{g}$$

For that, the G-force (M) can be solved easily.

### Filling of the other data

We already fill the two kinds of values. As we mentioned before, we have 7 kinds of data that have been lost. They are height, speed, length, drop, duration, G-force, angle. In these 7 kinds



of data, drop, duration, G-force and angle all lost more than 75 pieces of data, and height, speed and length lost 1, 4 and 5 pieces of data. For the last three, we only need to do some fitting, but for the data that had lost a lot, we need have two methods. The first method is to find the universal rule between different indexes and find the result, which had been already mentioned.

The rest method is to do the mechanism filling, and for the data of angle and time duration, we need to use this method. The reason for us to do that is because we cannot get a specific equation to get the value of angle and time duration. Just make the time duration as an example, we only know the length and top speed of the roller coaster. If we try to use the gravitation acceleration to calculate the average speed, it will be very complex and will cause a lot of errors.

The mechanism filling depends on the statistic application software.

The application software we use is the IBM SPSS statistic 25. We need to first input the data into the application. Then, we will use the left button to choose the useless data and use the right button to delete it. After that, choose the deletion analysis and use the EM (Expectation Maximization) algorithm to fill the blanks (complete the data). After that we can get the filled data like that:

Year/Date Opened	Height (feet)	Speed (mph)	Length (feet)	Number of Inversions	Drop (feet)	Duration (Second)	G Force	Vertical Angle (degrees)
2006	98.4	45.0	2788.8	10	65.3	92.0	4.618667	75.60981
2006	151.6	65.2	1476.4	2	148.0	60.0	4	78.55643
2018	72.0	45.0	1050.0	3	60.8	85.6	4.879061	97
1999	113.0	62.0	2956.0	6	128.4	167.0	4.451962	81.93878
1997	195.0	67.0	3828.0	6	170.0	190.0	3.7	71.42074
2014	108.3	51.6	2358.9	0	97.8	111.5	4.3	73.35087

(Table.3 Complete data after using SPSS)

SPSS has its own algorithm, actually the principle of SPSS depends on its own algorithm. When we input various of data, it will compare in each group of data and find the nearest one. After that, the computer will pair the lost data with exist one in specifically ratio and finally get the result (complete data).

### Error correcting

The next main step for us to do is to deal with the errors that exist in the data we already have.

Cedar Creek Mine Ride	Cedar Point	Sandusky	Ohio	United States	North America	Steel	Steel	Operating
Coaster Thrill Ride	Puyallup Fair	Puyallup	Washington	United States	North America	Wood	Sit Down	Operating
Corkscrew	Cedar Point	Sandusky	Ohio	United States	North America	Steel	Steel	Operating
Corkscrew	Valleyfair!	Shakopee	Minnesota	United States	North America	Steel	Sit Down	Operating
Cyclone	Lakeside Amusement Park	Denver	Colorado	United States	North America	Wood	Sit Down	Operating
Demon	California's Great America	Santa Clara	California	United States	North America	Steel	Sit Down	Operating
Demon	Six Flags Great America	Gurnee	Illinois	United States	North America	Steel	Sit Down	Operating

(Table.4 Errors in the chart 1)

Talon	Dorney Park & Wildwater Kingdom	Allentown	Pennsylvania	United States	North America	Steel	Inverted	Operating
Tatsu	Six Flags Magic Mountain	Valencia	California	United States	North America	Steel	Flying	Operating
Terminator Salvation: The Coaster	Six Flags Magic Mountain	Valencia	California	United States	North America	Steel	Wood	Operating
Texas Tornado	Wonderland Amusement Park	Amarillo	Texas	United States	North America	Steel	Sit Down	Operating
Timber Wolf	Worlds of Fun	Kansas City	Missouri	United States	North America	Wood	Sit Down	Operating
Timberhawk: Ride of Prey	Wild Waves Theme Park	Federal Way	Washington	United States	North America	Wood	Sit Down	Operating

(Table.5 Errors in the chart 2)

From picture 4 and picture 5, we can know that there are some mistakes in the chart, the type of the roller coaster should be one of the five following: sit down, stand up, inverted, suspended and wing. It should not be steel or wood, that’s an obvious mistake. After searching on the internet, we find that the actual type of these roller coasters should all be sit down.

Also, for the line 217, there is also an error exists. The actual kind of that roller coaster should be stand up instead of sit-down that showed on the chart. for line 35 and line 113, they have some huge difference with the actual data, which should be edited.

Now it’s the final step, to give a number of some indexes. There are two indexes very important but are not showed in the format of number. They are the construction and type. The number of them should be used later.

Now, we set that in construction, the wood is 1 and the steel is 2. Also, in the types, the sit-down will be 1, the inverted will be 2 and the rest kinds will all be 3.

Name	Construction	Type	Year/Date Opened	Height (feet)	Speed (mph)	Length (feet)	Number of Inversions	Drop (feet)	Duration (Second)	G Force	Vertical Angle (degrees)
10 Inversio	1	1	2006	98.4	45.0	2788.8	10	65.3	92.0	4.618667	75.60981293
Abismo	1	1	2006	151.6	65.2	1476.4	2	148.0	60.0	4	78.55642707
Adrenaline	1	1	2018	72.0	45.0	1050.0	3	60.8	85.6	4.879061	97
Afterburn	1	2	1999	113.0	62.0	2956.0	6	128.4	167.0	4.451962	81.9387849
Alpengeist	1	2	1997	195.0	67.0	3828.0	6	170.0	190.0	3.7	71.42073925
Alpina Blitz	1	1	2014	108.3	51.6	2358.9	0	97.8	111.5	4.3	73.35087248
Altair	1	1	2014	108.3	52.8	2879.8	10	93.6	121.7	4.635034	80.10702704
American E	2	1	1981	127.0	66.0	4650.0	0	147.0	143.0	3.606008	55
Anaconda	2	1	1989	118.1	55.9	3937.0	0	40.0	130.0	3.744433	74.48700271
Apocalyps	1	3	2012	100.0	55.0	2900.0	2	90.0	120.0	4.199191	75.06392315
Apocalyps	2	1	2009	95.0	50.1	2877.0	0	87.3	180.0	4.092888	70.67912825
Apollo's CH	1	1	1999	170.0	73.0	4882.0	0	210.0	135.0	4.1	65
Atlantica S	1	1	2005	98.4	49.7	1279.5	0	95.3	200.0	4.504907	80.63228168
Backlot Stu	1	1	2005	45.2	40.0	1960.0	0	31.2	64.0	4.236394	68.95606868

(Table.6 The data after being preprocessed.)

## 6. The new evaluation system

### 6.1 Establishment of the model

#### 6.1.1 Logarithmic Fuzzy Preference Programming (LFPP)

Assume that  $X = \{X_1, X_2 \dots X_n\}$  is a factor set, and  $X_i (i = 1, 2, \dots, n)$  is the important evaluate indexes in the evaluation system. The fuzzy judgement matrix A was showed below. In that matrix,  $a_{ij}$  represent the relative importance of index  $X_i$  over index  $X_j$ . It’s hard for the decision maker to have an accurate judge during the decision-making process. So, we need to input the Triangular Fuzzy Number to make the decision become more rational compared with parametric analysis (AHP). Depend on the research that made by Kahraman, the relative importance of index  $X_i$  over index  $X_j$  can be the value from  $(l_{ij}, u_{ij})$ ,  $m_{ij}$  represent the most possible relative importance. Now, the new matrix and Triangular Fuzzy Number are listed below:

$$A = (a_{ij})_{n \times n} = \begin{bmatrix} 1 & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & 1 \end{bmatrix}$$

$$\tilde{A} = (\tilde{a}_{ij})_{n \times n} = \begin{bmatrix} (1,1,1) & \cdots & (l_{1n}, m_{1n}, u_{1n}) \\ \vdots & \ddots & \vdots \\ (l_{n1}, m_{n1}, u_{n1}) & \cdots & (1,1,1) \end{bmatrix}$$

Grade	Implication	Triangular-fuzzy number	Triangular-fuzzy number expression	Expression in reciprocal
1	Element $i$ is equal to element $j$	$\tilde{1}$	(1,1,1)	(1,1,1)
3	A little important	$\tilde{3}$	(2,3,4)	(1/4,1/3,1/2)
5	More important	$\tilde{5}$	(4,5,6)	(1/6,1/5,1/4)
7	Very important	$\tilde{7}$	(6,7,8)	(1/8,1/7,1/6)
9	Extremely important	$\tilde{9}$	(8,9,9)	(1/9,1/9,1/8)

(Table.7 The scale of importance of triangular fuzzy number)

From that, we can get that:

$$l_{ij} = \frac{1}{u_{ij}}, m_{ij} = \frac{1}{m_{ji}}, u_{ij} = \frac{1}{l_{ij}}, 0 < l_{ij} \leq m_{ij} \leq u_{ij}, l_{ij} = m_{ij} = u_{ij} = 1$$

Different from the method of using AHP to use the single value to represent the relative importance of indexes, we will use the triangular fuzzy number to represent the scale of importance between the different evaluate indexes.

For the new fuzzy matrix that made represented by the triangular fuzzy number, Mikhailov came up with a new method of using a non-linear optimization for solving weight set, in that situation, we use the  $w_i/w_j$  as the variable. We can get a new formula that:

$$\mu_{ij}(w_i/w_j) = \begin{cases} \frac{(w_i/w_j) - l_{ij}}{m_{ij} - l_{ij}}, & \frac{w_i}{w_j} \leq m_{ij} \\ \frac{u_{ij} - (w_i/w_j)}{u_{ij} - m_{ij}}, & \frac{w_i}{w_j} \geq m_{ij} \end{cases}$$

In that expression, we use  $\mu_{ij}(w_i/w_j)$  to represent the importance of  $w_i/w_j$  fuzzy matrix

$$\tilde{a}_{ij} = (l_{ij}, m_{ij}, u_{ij})$$

One important step for us to do is to logarithm the fuzzy matrix, and we can get a similar equation in the logarithm of the triangular fuzzy number. The equation is listed below.

$$\ln \tilde{a}_{ij} \approx (\ln l_{ij}, \ln m_{ij}, \ln u_{ij}), i, j = 1, \dots, n$$

So, the logarithm of the fuzzy judgement matrix is listed below,

$$\mu_{ij} \left( \ln \left( \frac{w_i}{w_j} \right) \right) = \begin{cases} \frac{\ln(w_i/w_j) - \ln l_{ij}}{\ln m_{ij} - \ln l_{ij}}, & \ln \frac{w_i}{w_j} \leq \ln m_{ij} \\ \frac{\ln u_{ij} - \ln(w_i/w_j)}{\ln u_{ij} - \ln m_{ij}}, & \ln \frac{w_i}{w_j} \geq \ln m_{ij} \end{cases}$$

In that expression, we use  $\mu_{ij} \left( \ln \left( \frac{w_i}{w_j} \right) \right)$  to represent the degree of membership of  $\ln \left( \frac{w_i}{w_j} \right)$  over,  $\tilde{a}_{ij} = (l_{ij}, m_{ij}, u_{ij})$ .

If we use  $\lambda$  to represent the least degree of membership, it will be:

$$\lambda = \min \{ \mu_{ij} \left( \ln \left( \frac{w_i}{w_j} \right) \right) | i = 1, \dots, n - 1; j = i + 1, \dots, n \}$$

The reason for us to use the logarithm is that logarithm can turn the division into subtraction, which will save a lot of time and make the equation become less complex. Now, we only need to calculate the maximum or minimum value and it can turned to the non-linear optimization problems, it will be finally equal to:

*min*  $1 - \lambda$

$$s. t. \begin{cases} \ln w_i - \ln w_j - \lambda \ln \left( \frac{m_{ij}}{l_{ij}} \right) \geq \ln l_{ij}, & i = 1, \dots, n - 1; j = i + 1 \dots n \\ -\ln w_i + \ln w_j - \lambda \ln \left( \frac{u_{ij}}{m_{ij}} \right) \geq -\ln u_{ij}, & i = 1, \dots, n - 1; j = i + 1 \dots n \\ w_i \geq 0, & i = 1, \dots, n \end{cases}$$

To avoid the situation of no solution, we will add the non-error parameter  $\delta_{ij}$  and  $\eta_{ij}$ , the inequality will become:

$$\begin{aligned} \ln w_i - \ln w_j - \lambda \ln \left( \frac{m_{ij}}{l_{ij}} \right) + \delta_{ij} &\geq \ln l_{ij}, \\ -\ln w_i + \ln w_j - \lambda \ln \left( \frac{u_{ij}}{m_{ij}} \right) + \eta_{ij} &\geq -\ln u_{ij}, \end{aligned}$$

Therefore, the logarithm fuzzy programming model will be:

$$\min J = (1 - \lambda)^2 + M \times \sum_{i=1}^{n-1} \times \sum_{j=i+1}^n (\delta_{ij}^2 + \eta_{ij}^2)$$

$$s. t. \begin{cases} x_i - x_j - \lambda l n \left( \frac{m_{ij}}{l_{ij}} \right) + \delta_{ij} \geq l n l_{ij}, & i = 1, \dots, n-1; j = i+1 \dots n \\ -x_i + x_j - \lambda l n \left( \frac{u_{ij}}{m_{ij}} \right) + \eta_{ij} \geq -l n u_{ij}, & i = 1, \dots, n-1; j = i+1 \dots n \\ \lambda, x_i \geq 0, & i = 1, \dots, n \\ \delta_{ij}, \eta_{ij} \geq 0, & i = 1, \dots, n-1; j = i+1 \dots n \end{cases}$$

In this expression,  $x_i = l n w_i$ , ( $i = 1, \dots, n$ ),  $M$  is a big number we already know to make sure the validity of weight.

From the equalities we already know, we can get the optimal solution  $x_i^*$  and optimal parameter  $\lambda^*$ . And the weight of that fuzzy matrix will be:

$$w_i^* = \frac{\exp(x_i^*)}{\sum_{j=1}^n \exp(x_j^*)}, \quad i = 1, \dots, n$$

$\lambda^*$  is the consistency index, when  $\lambda^*$  is a negative number, it means the two matrixes have big differentiation, when  $\lambda^*$  is a positive number, it means that the two matrixes have little differentiation, when  $\lambda^*$  is equal to 0, it means that the two matrixes are the same.

### 6.1.2 Entropy weight model

According to the Second Law of Thermodynamics, entropy is a material state parameter that reflects the irreversibility of spontaneous processes. At first, in thermodynamics, entropy was used to describe the process of heat motions. It demonstrated that the process of heat transformation is directional and irreversible. In normal conditions, as the value of entropy gets greater, the degree of disorder of heat motions of atoms gets greater too. Therefore, the value of entropy also shows the intensity degree of atoms' motions. Later, in information theory, Shannon recognized entropy as the disordered measurements inside the systems, and he used entropy to represent the uncertainty in the occurrence of things. This allowed the concept of entropy to break through the scope of thermodynamics and to be applied to other areas of sciences.

Information entropy is all being called as Shannon entropy. It is used to describe the degree of uncertainty measurements in random variables. In information theory, it is also used to represent the degree of uncertainty in signals dispatched by sources of information. It can be understood in a way that, if a system's degree of disorder is lower, then the value of entropy is lower, and the quantity of information that the corresponding source is higher, vice versa. It can be concluded that the value of entropy corresponds to the degree of disorder, and that entropy is negatively correlated with the quantity of information.

According to the basic principles of information entropy, during the weighing calculations of evaluating indicator, entropy can be recognized as an ideal scale. Suppose that for a decision question with  $m$  amount of evaluation targets and  $n$  amount of evaluating indicators, the raw materials of the corresponding indicators of the evaluation targets can be constructed to the following judgment matrix:

$$A = (a_{ij})_{n \times n} = \begin{bmatrix} 1 & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & 1 \end{bmatrix}$$

In that  $a_{ij}$  represents the relative importance of the  $i$ th evaluating indicators to the  $j$ th evaluation target, ( $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ ).

First, normalization of the relative importance of the  $n$ th evaluating indicator to  $m$  quantities of evaluation objects is carried out, that is:

$$p_{ij} = \frac{a_{ij}}{\sum_{j=1}^n a_{ij}}$$

$$H_i = -k \sum_{j=1}^n p_{ij} \times \ln(p_{ij}), (i = 1, 2, \dots, m, j = 1, 2, \dots, n)$$

In the equation,  $k=1/\ln n$ . After calculating entropy of the its indicator, the entropy weight of the its indicator can be expressed as:

$$f_i = \frac{1 - H_i}{m - \sum_{i=1}^m H_i}$$

In that equation,  $0 \leq f_i \leq 1, \sum_{i=1}^m f_i = 1$ .

We need to use the fuzzy interval operation to get the whole entropy weight and fuzzy matrix because we use the triangular fuzzy number to represent the consistency between two indexes.

We will use the data in fig.7 to get the fuzzy matrix  $\tilde{A}$

$$\tilde{A} = \begin{bmatrix} \tilde{a}_{11} & \cdots & \tilde{a}_{1n} \\ \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \cdots & \tilde{a}_{nn} \end{bmatrix}$$

Then, we will use the triangular fuzzy number and exist algorithm to get the entropy weight of each index. Finally use the algorithm of entropy to get the specific entropy weight  $f_i$  of each index.

### 6.1.3 The final comprehensive evaluation system

Now, we already use the entropy weight and Logarithmic Fuzzy Preference Programming (LFPP) to make the model, and we need to combine them together.

The reason for us to do that is Entropy weight model find the entropy behind the data of each factor, which describes the objective information inside the system. The logarithmic fuzzy programming model, on the contrary, calculates the optimal weight through the subjective preference of individuals for each factor. As a result, we combine the two models to evaluate the advantages, and get more scientific results.

We set the different weight for entropy weight and LFPP, and finally, our weight for the entropy weight is  $\beta$  and the weight for LFPP is  $1-\beta$ . We don't set the weight of LFPP and entropy weight in half, instead, the weight of LFPP is actually bigger than the weight of entropy weight.

That is because entropy weight has its own exist algorithm. And the algorithm cannot favor the data we already get. The result of it has a huge differentiation from our expectation. Also, two of the all 11 indexes, (constructions and types), will be very hard to calculate in the entropy weight. That is because we only set two values of construction (wood is 1 and steel is 2) and three values of types (sit-down is 1, inverted is 2 and the others is 3). Because 2 is twice as many as 1 and 3 is three times bigger than 1. But the real situation is not like that. If we use the entropy weight to calculate that, it will be very chaotic, so we just delete these two variables in the entropy weight. Finally, our weight of LFPP is  $1-\beta$  and weight of entropy weight is  $\beta$ .

Another important step is to make a new scoring mechanism. Because some variables used twice or even three times in our model, and more important is, they represent the different meaning. For example, the speed of the roller coaster. If the top speed of it is very high, it will favor the excitement of that roller coaster, however, it will decrease the degree of safety.

To solve that problem, we separate the repeat factor and give each factor a single chance of gaining the score. For example, if the speed of the roller coaster appears twice in the model, then there will be two factors of speed for scoring.

We have three parts to score, each include several factors (we will mention that later). And each part of these all three have different weight. The final score of the factors will be the variation of minimum membership function. (we set that the top is 1 and the bottom is -1). When we get the score of each factor, we will use them to multiple the weight that we already set, to get a new score.

Finally, depend on the  $\beta$  of entropy and  $1-\beta$  of LFPP. We can get the final score and ranking list of different roller coasters.

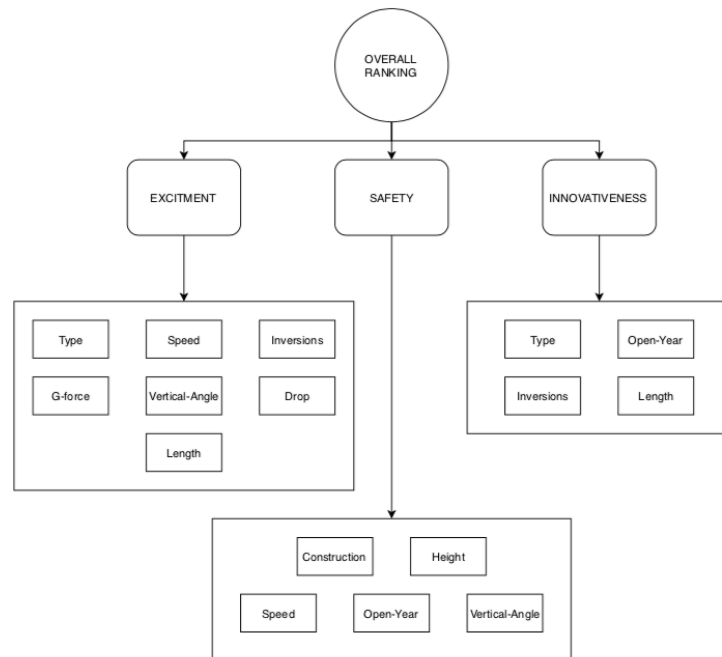
### 6.1.3 Kendall's tau distance model

Kendall's tau distance between permutations  $v$  and  $y$  is a right-invariant metric defined as the minimum number of pairwise adjacent transpositions needed to turn one permutation into the other. Assuming that one of the permutations, say  $y$ , is the identity permutation  $e$ , it can be written as:

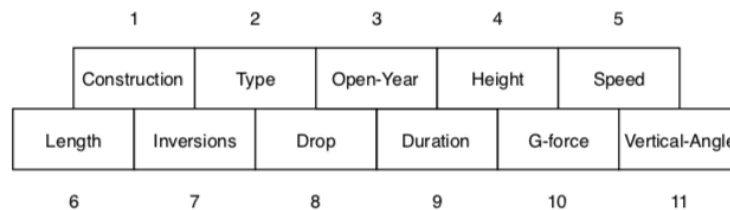
$$D_k(v) = V_i(v)$$

## 6.2 Solving the model

The fuzzy matrix and weight of each factor based on the questionnaire.



(fig.1 Three main parts to make the ranking list and different factors)



( fig.2 The number of each different factor )

The data we already get have 11 factors, and we made a questionnaire to help us to set the weight of each factor.

The result of the questionnaire tells us that there are three main parts that include a lot of different factors. the three parts are excitement, safety and innovation. Just like the picture showed below. Also, each factor will have its own number.

The questionnaire is generally about user’s interest on the type and construction of the roller coasters. Hundreds of people do the questionnaire we made, and they score the different types of roller coasters. Finally, we set the weight of each part based on the data of the questionnaire. Based on the questionnaire, we can get the fuzzy matrix showed below. (I represent innovation, E represent excitement and S represent safety).



Overall	Excitement	Safety	Innovation
Excitement	1	(2,3,4)	(4,5,6)
Safety	(1/4,1/3,1/2)	1	(1,2,3)
Innovation	(1/6,1/5,1/4)	(1/3,1/2,1)	1

(Table.8 Overall fuzzy matrix)

Excitement	Type	Speed	Length	Inversions	Drop	G-force	Verticle-angle
Type	1	(1/5,1/4,1/3)	(1/6,1/5,1/4)	(1/3,1/2,1)	(1/9,1/9,1/8)	(1/4,1/3,1/2)	(1/7,1/6,1/5)
Speed	(3,4,5)	1	(1/3,1/2,1)	(1,2,3)	(1/4,1/3,1/2)	(1,2,3)	(1/3,1/2,1)
Length	(4,5,6)	(1,2,3)	1	(2,3,4)	(1/3,1/2,1)	(1,2,3)	(1,1,1)
Inversions	(1,2,3)	(1/3,1/2,1)	(1/4,1/3,1/2)	1	(1/7,1/6,1/5)	(1,1,1)	(1/5,1/4,1/3)
Drop	(8,9,9)	(2,3,4)	(1,2,3)	(5,6,7)	1	(4,5,6)	(1,2,3)
G-force	(2,3,4)	(1/3,1/2,1)	(1/3,1/2,1)	(1,1,1)	(1/6,1/5,1/4)	1	(1/4,1/3,1/2)
Verticle-angle	(5,6,7)	(1,2,3)	(1,1,1)	(3,4,5)	(1/3,1/2,1)	(2,3,4)	1

(Table.9 The fuzzy matrix between the factors of “Excitement”)

safety	Construction	open-year	Height	Speed	Verticle Angle
Construction	1	(1/4,1/3,1/2)	(1/3,1/2,1)	(1,2,3)	(1/3,1/2,1)
Open-year	(2,3,4)	1	(2,3,4)	(4,5,6)	(1,1,1)
Height	(1,2,3)	(1/4,1/3,1/2)	1	(2,3,4)	(1/3,1/2,1)
Speed	(1/3,1/2,1)	(1/6,1/5,1/4)	(1/4,1/3,1/2)	1	(1/5,1/4,1/3)
verticle-angle	(1,2,3)	(1,1,1)	(1,2,3)	(3,4,5)	1

(Table.10 The fuzzy matrix between the factors of “Safety”)

Innovation	Construction	Open-year	Inversions	Duration
Construction	1	(1,1,1)	(6,7,8)	(1/3,1/2,1)
Open-year	(1,1,1)	1	(5,6,7)	(1/3,1/2,1)
Inversions	(1/8,1/7,1/6)	(1/7,1/6,1/5)	1	(1/9,1/8,1/7)
Duration	(1,2,3)	(1,2,3)	(7,8,9)	1

(Table.11 The fuzzy matrix between the factors of “innovation”)

Based on the fuzzy matrix and the result of the LFPP, we get the weight of the three parts (Excitement, Safety and Innovation).

Item	Excitement	Safety	Innovation
Weight	64.60%	22.80%	12.60%

(Table.12 Different parts and their weight)

Also, we get the different weights for all the 11 different factors. and they are showed below. The first chart is about the weight that we used in the LFPP model and the second chart is about the weight that we used in the entropy weight. As I mentioned before, there are two variables in the entropy weight be deleted, and there are no repeat values in entropy weight algorithm, so it only includes 9 variables while LFPP has 16 variables.

Also, the entropy weight can tell us the weight of 9factors in its model. (As we mentioned before, we delete the type and construction in the entropy weight because they will make the result very chaotic.)

Item	Type	Speed	Length	Inversion	Drop	G-force	Verticle angle	Construcion
weight	0.026	0.082	0.113	0.066	0.176	0.062	0.118	0.028
Item	Open-year	Height	Speed	Verticle angle	Type	Open-year	Inversion	Duration
weight	0.065	0.048	0.016	0.064	0.037	0.033	0.1	0.053

(Table.13 Weight for factors in LFPP model)

Item	Open-year	Height	Speed	Length	Inversion	Drop	Duration	G-force	Vertical Angle
weight	0.045	0.102	0.076	0.126	0.172	0.113	0.086	0.097	0.09

(Table.14 Weight for factors in entropy weight)

After we get the weight of each factor in two algorithms and three different parts, we can get the specific weight in all parts (excitement, safety and innovation).

Item	Type	Speed	Length	Inversions	drop	G-force	Verticle angle
weight	0.045	0.102	0.076	0.126	0.172	0.113	0.086

(Table.15 Weight of factors in *Excitement*)

Item	Type	Open-year	Inversion	Duration
weight	0.299	0.299	0.048	0.354

(Table.16 Weight of factors in *Innovativeness*)

item	Construction	Open-year	Height	Speed	Verticle-angle
weight	0.124	0.328	0.15	0.069	0.328

(Table.17 Weight of factors in *Safety*)

### 6.2.1 Data processing in each factor

The data we already get is different in each factor, they have different range and different unit, to make them easily be calculated in the matrix and entropy weight, we need to do another data process base on the data we already get.

We calculate the average number of each factor, then, use each data to subtract the average number, then, divide by the standard deviation of that factor, we can get the data like that:

construction	type	date	height	speed	length	number of inversions	drop	duration
-0.50	-0.50	0.41	-0.56	-0.89	-0.25	3.00	-0.92	-0.73
-0.50	-0.50	0.41	0.25	0.35	-1.15	-0.08	0.28	-1.46
-0.50	-0.50	1.32	-0.96	-0.89	-1.44	0.31	-0.98	-0.88
-0.50	0.00	-0.13	-0.34	0.15	-0.13	1.46	0.00	0.98
-0.50	0.00	-0.28	0.90	0.46	0.46	1.46	0.60	1.50
-0.50	-0.50	1.01	-0.41	-0.49	-0.54	-0.85	-0.45	-0.29
-0.50	-0.50	1.01	-0.41	-0.41	-0.19	3.00	-0.51	-0.05
0.50	-0.50	-1.50	-0.13	0.40	1.03	-0.85	0.27	0.43
0.50	-0.50	-0.89	-0.26	-0.22	0.54	-0.85	-1.28	0.13
-0.50	0.50	0.86	-0.54	-0.28	-0.17	-0.08	-0.56	-0.09

(Table.18 The data after processing in the model)

We want to make a minimum membership in the model, so we still need to continue the data processing, we set that the top value of each factor is 1 and least value is -1.

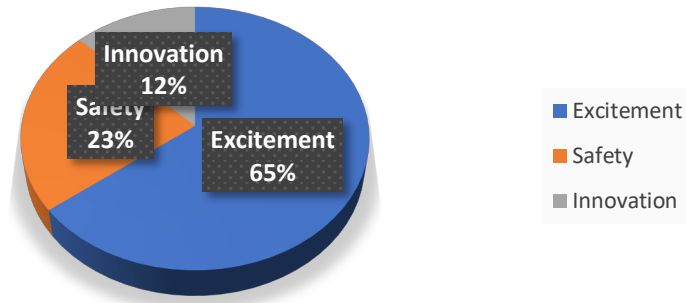
## 6.3 Simulation and conclusion

Finally, we get the ranking list of top-ten roller coaster based on our model, the result is showed below. The chart includes the name, park and the country.

Ranking	Name	Park	City/Region
1	GateKeeper	Cedar Point	Sandusky
2	Soaring Dragon & Dancing Phoenix	Nanchang Wanda Theme Park	Xinjian
3	Dinoconda	China Dinosaurs Park	Xinbei
4	Banshee	Kings Island	Mason
5	Eejanaika	Fuji-Q Highland	Fujiyoshida
6	Riddler's Revenge	Six Flags Magic Mountain	Valencia
7	Steel Vengeance	Cedar Point	Sandusky
8	Flying Aces	Ferrari World Abu Dhabi	Abu Dhabi
9	Hyperion	Energylandia	Zator
10	X2	Six Flags Magic Mountain	Valencia

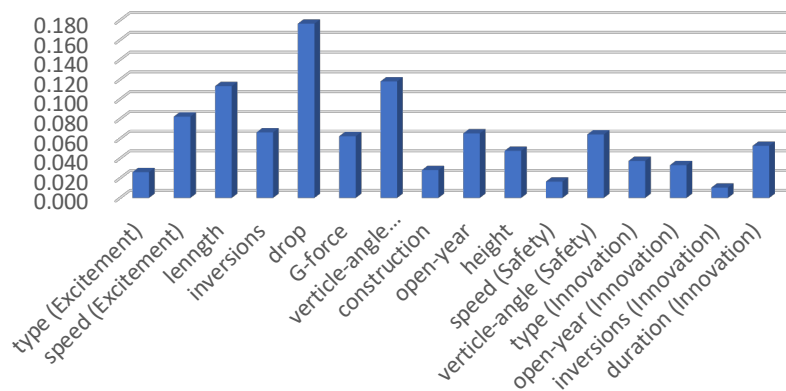
(Table.19 Top-ten roller coasters in our ranking list)

### weight of different part

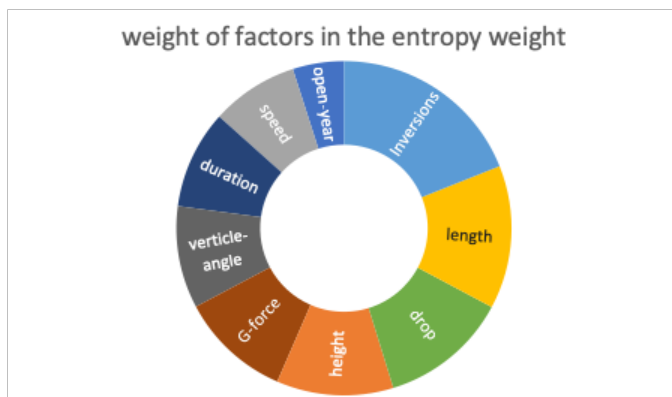


(fig.3 Weight of different part)

### weight of different factor



(fig.4 Weight of different factor in LFPP model)

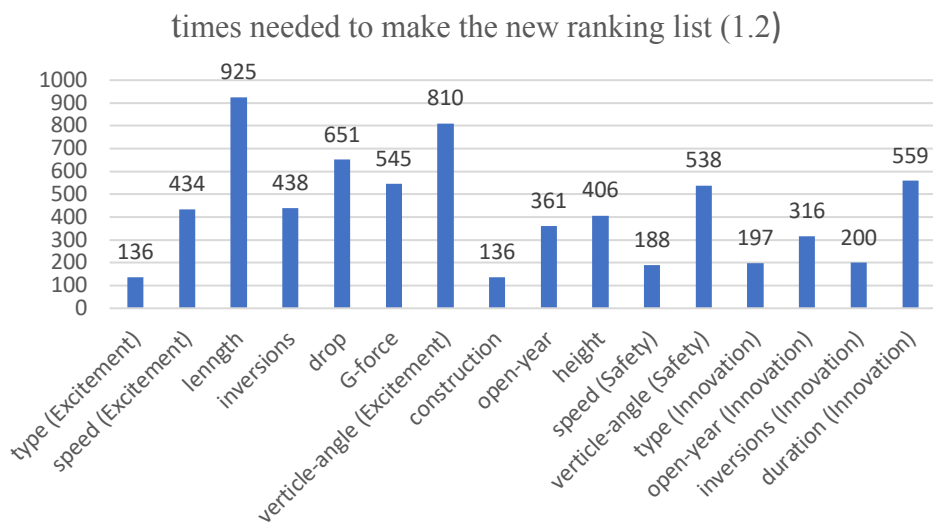


(fig.5 Weight of factors in the entropy weight)

### 6.4 Sensitive analysis

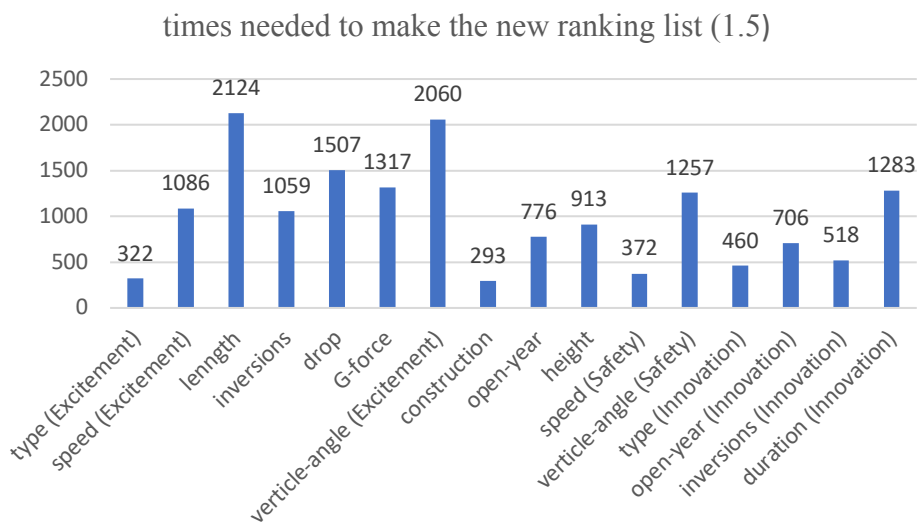
Our model includes more than 16 factors, they all have their different weight. If their weight changes, the ranking will be influenced, but the degree of impact will be different in different factors. So, we just change the weight of a single to find which weight will influence the most.

Firstly, we will make the factors  $q$  1.2 times larger than before. Because we have 16 factors and the sum of these 16 weights is 1, so the rest factors will decrease:  $\frac{1.2-1}{15} \times q = \frac{q}{75}$ . And we record the times needed to make the new ranking list after changing the weight based on the old ranking list that we already made before. That is to say, if the time needed to make the new ranking list is very large, the impact of that weight change is also very obvious. Now, the result of that are showed below.



(fig.6 Times needed to make the new ranking list (1.2))

Also, we do the same thing in 1.5 times, and the result are showed below,



(fig.7 Times needed to make the new ranking list (1.5))

## 7. Compare and improve our ranking list

### 7.1 The exist authoritative roller coaster ranking list

To make our ranking list more authoritative and authentic, we find some official ranking list online, such as the “wiki 2018”, “Coaster Buzz ranking” and “wiki 2017”. Also, we will compare our ranking list with the official one.

The ranking list in “wiki 2018” separate the steel roller coaster and the wooden roller coaster apart, so we need to combine them together. After we get the ranking list, we first compare the two online ranking list. And we find that 43 of them are repeated (both appear in the two-ranking list). We find these 43 roller coasters and ranking them based on our ranking list.

The chart below shows three different ranking lists. The left one is the ranking list from the “coaster buzz ranking”, the middle one is the ranking list from the “wiki 2018 ranking” and the right one is our own ranking list.

Coaster Buzz ranking		"wiki 2018"		our ranking list	
Ranking	Name	Ranking	Name	Ranking	Name
1	Steel Vengeance	1	Phoenix	1	Fury 325
2	Fury 325	2	Fury 325	2	Steel Vengeance
3	El Toro	3	El Toro	3	Banshee
4	Lightning Rod	4	Millennium Force	4	Leviathan
5	Millennium Force	5	Voyage	5	Intimidator 305
6	Twisted Colossus	6	Boulder Dash	6	X2
7	Boulder Dash	7	Steel Vengeance	7	Millennium Force
8	Wicked Cyclone	8	Lightning Rod	8	Twisted Colossus
9	Voyage	9	Outlaw Run	9	Intimidator
10	Maverick	10	Ravine Flyer II	10	Alpengeist

(Table.20 Three different ranking lists)

### 7.2 The geographic reason to cause the differentiation over the ranking list

After comparing the three different ranking lists, we find that they don’t have enough consistency, and there are few reasons for that phenomenon.

The first reason is the factor of the location. The factor of the location had been ignored before, it didn’t appear in the LFPP and entropy weight model. However, it will bring huge impact to the different ranking list. For example, in our ranking list, the first one is the roller coaster in Abu Dhabi, UAE. However, UAE is the country in middle east, and that is very far from both America and Europe. The exist authoritative online ranking list mostly include the ranking list in America, that’s a very important reason.

### 7.3 Consistency between head and tail

Because every ranking list has its own evaluation system, they will be different with each other. However, for some data that have a lot of difference, for example, the best one and the least one, they will usually stay the same. Even the evaluation system will be quite different, they will still keep the same for some data.

To use the “consistency between head and tail”, we need to first arrange the elements (data) in the descending order. For  $m$  elements in total, we need to find 20% elements at the top ( $x$  elements at all) and 20% elements at the bottom ( $y$  elements at all). And we can get the consistency like that:

$$S = \frac{x + y}{0.4m}$$

It’s pretty obvious that the range of  $S$  between 0 and 1. And the bigger the value of  $S$  is, the better the consistency between head and tail are. Because the evaluation system is different, so it will be very chaotic for us to calculate. To make the result clearer, we just calculate the top value and the bottom value.

By calculating “consistency between head and tail”, we find that the consistency between the two ranking lists on line is very high. However, the same between ours and the online authoritative ranking list is quite low. We find some reasons about that. The details will be mentioned later.

#### 7.4 Optimization about the limitations of minimum membership function

As we mentioned before, the consistency between our ranking list and the authoritative ranking list is not so high as them. The reason might be the limitations of minimum membership function. As we mentioned before, the highest score of each factor is 1 and the lowest score of each factor is -1. That is to say, some points that bigger than 1 will be regarded as 1. The only result is the total mark of the roller coasters will be influenced. If one of the score of the roller coaster is very high (much far from 1), it will be influenced even more seriously. We find two examples that have this problem.

number	name	park	city/region
105	Fury 325	Carowinds	Charlotte
233	Soaring Dragon & Dancing Phoenix	Nanchang Wanda Theme Park	Xinjian

(Table.21 Two examples that have problems)

/	/	type	speed	length	inversions	drop	G-force	Verticle-angle	construction
number	score	0.041	0.076	0.112	0.056	0.110	0.054	0.100	0.034
232	0.464	0.000	1.000	0.713	1.000	1.000	0.237	0.847	1.000
104	0.222	-1.000	1.000	1.000	-0.846	1.000	-1.000	0.511	1.000
number	score	open-year	height	speed	Verticle-angle	type	open-year	inversion	duration
232	0.464	0.066	0.057	0.027	0.067	0.059	0.051	0.026	0.064
104	0.222	1.000	-0.929	-1.000	-0.847	0.000	1.000	1.000	0.054
/	/	1.000	-1.000	-1.000	-0.511	-1.000	1.000	-0.846	1.000

(Table.22 Two examples and the data before optimization)

Ranking	Name	Ranking	Name	Ranking	Name
1	Steel Vengeance	1	Phoenix	1	Fury 325
2	Fury 325	2	Fury 325	2	Leviathan
3	El Toro	3	El Toro	3	Intimidator 305
4	Lightning Rod	4	Millennium Force	4	Steel Vengeance
5	Millennium Force	5	Voyage	5	Top Thrill Dragster
6	Twisted Colossus	6	Boulder Dash	6	Millennium Force
7	Boulder Dash	7	Steel Vengeance	7	Banshee
8	Wicked Cyclone	8	Lightning Rod	8	Intimidator
9	Voyage	9	Outlaw Run	9	Diamondback
10	Maverick	10	Ravine Flyer II	10	Behemoth

(Table.23 The ranking list after optimization)

## 8. FUN ROLLER (The Application)

Base on the mathematical model established above, an application Fun Roller was produced for fans of roller coaster worldwide. It is an app that gives users detailed information about

Base on the mathematical model established above, an application FUN ROLLER was produced for fans of roller coaster worldwide. It is an app that gives users detailed information about roller coasters around the world. Traditional online roller coaster websites give rankings subjectively from all users' feedback. Sometimes the ranking is objective but only focusing on one parameter e.g. speed, height, length, etc. However, FUN ROLLER can provide rankings based on the objective parameters of each roller coaster, and the weighing of each factor is calculated from users' personal preference. During registration, users are required to complete a survey about his/her basic information and his/her preference about roller coaster, as shown in Figure 8. Figure 9.1 and Figure 9.2. The survey includes gender, age, preferred type of roller coaster, which elements the user expect mostly, how he/she value excitement, safety, and innovativeness, etc. After that, the algorithm starts to work, and the weighing of each factor will be re-calculated according to the user preference. As a result, it provides a new customized roller coaster ranking to just for the user. If a user is new to roller coaster and he/she does not know his/her preference, the survey can be skipped, and the weighing is given from public feedback. Users can adjust the preference anytime and the ranking system will change simultaneously.

Besides the ranking, there is more to discover. The powerful yet easy-to-use Artificial Intelligence bilateral feedback system will aid anyone in finding the right rollercoaster for them. By simply talking to the app, users can get a list of roller coasters that they may be interested in. Edubank — the “shopping cart” and collection folder — will allow users to buy tickets within few clicks on the screen. The map section allows users to navigate a route to the desired roller coaster site or to find rollercoasters around them. These brief but practical tools will guide users from searching for a rollercoaster to ticketing to transportation, and to finally rate their experiences in our app. These tools will make anyone's trip to their dream rollercoasters easier and more accessible.

This revolutionary logarithm and app will be live on the App Store in the following weeks. It would start a new era for ranking as the innovative approaches including personalization, AI feedback, and accurate delivery is first to be used in rankings. FUNROLLER CO. Ltd. will



continue to contribute more tools of ranking a variety of products in the near future.

## 1 Introduction of "FUN ROLLER"

**FUN ROLLER** the world's most popular and authoritative source of rollercoaster. Check the list of roller coasters full of entertainment, maps, guide etc. Enjoy roller coasters, interesting video, comments, ticket price and your favorite food around the roller coaster. Yes, this is one stop destination for your family and friends. **FUN ROLLER** is an app that gives you knowledge about roller coasters around the world. The app gives you details about categories of roller coasters under it, if any. The app provides information about various adventurous, exciting roller coasters, both country wise and city wise. You can sort the roller coasters based on popularity and alphabetically.

Some of the famous roller coasters are Fury 325 (Charlotte), Millennium Force (Ohio), Steel Vengeance (Ohio), Expedition GeForce (Rhineland-Palatinate), Superman the Ride (Massachusetts), Apollo's Chariot (Virginia), Iron Rattler (Texas), Leviathan (Ontario), Maverick (Ohio). Travel to your favorite destinations with all the required information, pictures and videos in the palm of your hand.

Find out particulars roller coasters like: -

- \* Significance
- \* Best time to visit
- \* How to get the list of roller coaster while you are travelling
- \* Best roller coaster fit to you
- \* Fun video
- \* Evaluation of the roller coasters
- \* Feedback of the roller coasters
- \* Etc.

## 2 Features Of the App:-

- \* Description - Know the detail, country and city it belongs to, address and coordinates.
- \* Search - just type in whatever you are searching for in the search within a particular category.
- \* EduBank - Save your learning with EduBank.
- \* Contribute - If you think we missed something, please contribute and we'll undated it.
- \* Nearby or Map feature - It has 3 options: --
  - Draw Shape - We provide you the feature to look at nearby roller coasters of interest by drawing a shape around it.
  - Select Marker - Pinpoint the places through which you want to draw the route. Path will be drawn/marked. You can play the rout. Enjoy the virtual street view.
  - Reset - Refresh the map.
- \* Customize user's own rank
- \*Recommend the roller coasters to the user according to the data that we collect.
- \*Etc.

Know the detail about all roller coasters on finger trip. So what are you waiting for? It's a small world after all. Visit your favorite roller coasters with **FUN ROLLER** and learn more about your area of interest.

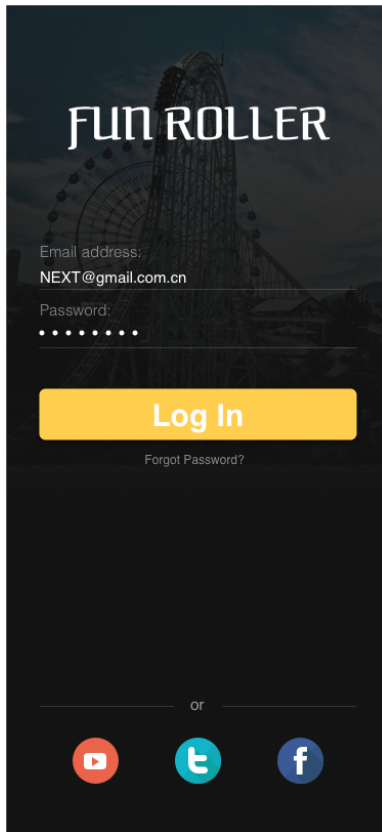


Fig. 8 Log in interface

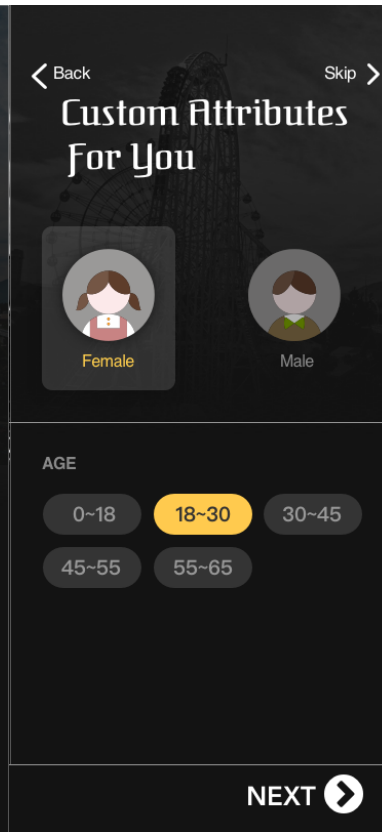


Fig. 9.1 Questionnaire

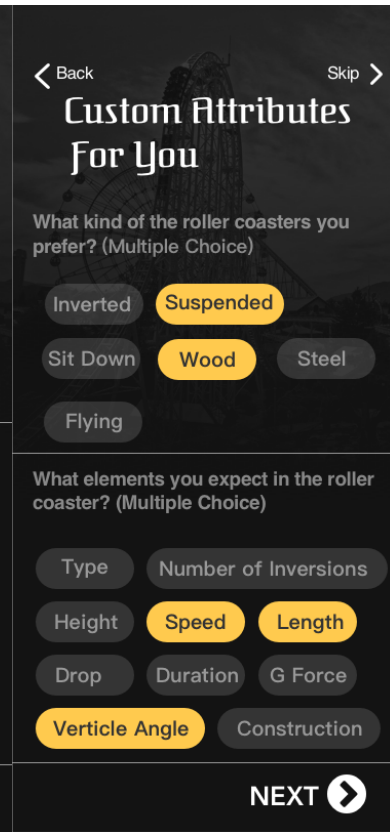


Fig. 9.2 Questionnaire

### 3 Log In Interface

Fig. 8 is the log in system of the APP. In the **FUN ROLLER**, there are four ways to log in. Log in by user’s email address or phone number is the first way. If the user forgets the password, he can click the button under the password, “Forgot Password?” to find user’s password again. Furthermore, users may use **YOUTUBE**, **TWITTER** or **FACEBOOK** to log in **FUN ROLLER**. **FUN ROLLER** will gather the user’s information from those platforms. For example, there is a user, who choose **YOUTUBE** to log in **FUN ROLLER**. Then we can browse his history watched in the **YOUTUBE** to find out his hobby. For instance, this user usually will search some fiction stuff on **YOUTUBE**, so **FUN ROLLER** will recommend him to go to the **TYCOON 3** at the **Universal Studio**.

### 4 Questionnaire

Fig. 9.1 and Fig. 9.2 is the questionnaire system after the user register account. Once the users registered an account at Fun Roller, they will be required to complete a questionnaire before entering the main page of the app. This questionnaire contains general questions like users’ gender, age, and family status, and also asks about specific questions related to roller coasters. Example questions can be objective like “What is your age?” or subjective like “From 1 t 10, please rate to what extent do you value height/G-force/vertical drop in elevation as an important determinant in determining the overall excitement of roller coasters?” By analyzing users’ completed questionnaire, we would get a broad sense of this users’ preferences. Once the questionnaire is completed, user’s result, keywords and general traits we captured from the questionnaire will be stored into our company’s database as well as the users’ personal section page. Our algorithm will then automatically change the weighing of each factors influencing

the roller coaster ranking according to this user's results from the questionnaire, and a more user-based, more accurate ranking then would be presented to the user.

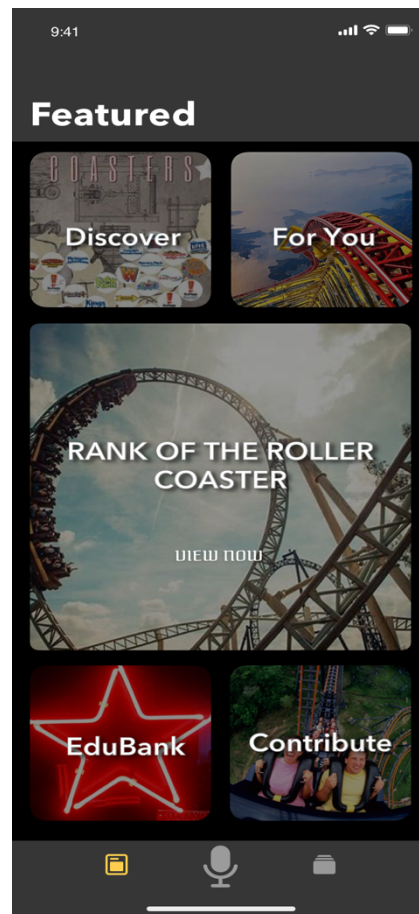


Fig. 10 Home page of FUNROLLER

### 5 Featured

**Feature** is the home page of the whole app. There are five functions on the home page. They respectively are Rank of the roller coaster, For You, Discover, Edubank, Contribute. Fig. 10 is showing those functions.

### 6 Rank Of the Roller Coaster

At the middle of the Fig. 10. In this section, there are different kind of the ranks for the different kind genre. Such as there are “TOP TEN HIGHEST ROLLER COASTERS OF THE WORLD”, “TOP TEN BEST ROLLER COASTERS OF THE WORLD”, “TOP TEN MOST EXCITING ROLLER COASTERS OF THE WORLD”, etc. Moreover, we will according to the user's browsing history to push the rank to the user. For instance, there is a user who always browses wooden roller coasters, then **FUN ROLLER** will push “TOP TEN WOODEN ROLLER COASTERS OF THE WORLD”. In addition, except the ranks of the whole world, also **FUN ROLLER** will according to the user's country to push the ranks of the whole country.

### 7 For You

At the top right corner of the Fig. 10. **FUN ROLLER** will guess what kind of the roller coaster that you like. In the beginning, when you register the account. **FUN ROLLER** will give you a

questionnaire to infer the roller coasters that you like. In the questionnaire there are include your city, do you fear of height, what kind of roller coaster that your interest in, etc.

Moreover, **FUN ROLLER** can log in by using TWITTER, FACEBOOK or YOUTUBE. We can collect data and cookies that from your TWITTER, FACEBOOK or YOUTUBE account to speculate the roller coasters genre that users like. In addition, TWITTER, YOUTUBE and FACEBOOK can share information with us. We can collect the forward statues of the users, and the watched history on the YOUTUBE. Furthermore, we gather user's following topic and what they like to calculate. Also, we can use cookies to judge the user's personality. A person's personality will affect the choice of the roller coaster. Then we use user's personality to determine what kind of the roller coasters suitable for users. All in all, we use those data to give a best choice to the user.

Furthermore, **FUN ROLLER** will record the user's history browsing and put the data in the background program to calculate the user's the most favorite flavor of the roller coaster. The data at the beginning (questionnaire, FACEBOOK.YOUTUBE and TWITTER) is not the fixed data. The data will change according to the user's history browsing. The users can buy or reserve the ticket in **FUN ROLLER**. We set a weight between the history browsing and reserve ticket. Because when users browse a roller coaster, which cannot express the user like this genre. But when customer reserve this roller coaster. We get 90% of the probability that the user like this kind of roller coaster. So that we set the weight of the history browsing is 20%, and the weight of the reserve ticket is 80%.

When we recommend the roller coasters to the users. Of course, we will consider the distance traveled between the roller coasters and the users. **FUN ROLLER** has an intelligential filter to.

## 8 EduBank

At the lower left corner of the Fig. 10. Edubank is similar to a collection folder or a shopping cart. Once discovered a rollercoaster that is of interest either in the description page, map section or on the ranking, you can add the rollercoaster to Edubank. If you click on the Edubank button, you can then review the rollercoaster that is of interest to you. You can make purchases of the admission tickets to these rollercoasters in Edubank and make an appointment for your trip by using our self-checkout system. As a result, we would get some commission from the roller coaster operators for each ticket sold. Once you made a purchase, you can then choose to plan the trip to the rollercoaster, in which you will be directed to the Map section.

We will also keep a record of the content that you've added to Edubank to optimize your search results and roller coaster recommendations for you.

## 9 Discover

At the top left corner of the Fig. 10. In the Map section, you will find a map powered by google maps. At first glimpse it is just like any other digital maps in the world — you can find a green dot in the center of the map, and you can see the streets around you. Yet as you zoom out, you will behold yourself to many little yellow dots — each dot represents a roller coaster site. By simply clicking on these dots you will see a brief description of the roller coaster which. If you are interested in that site you can either add the site to your Edubank List, see a more detailed description of the rollercoaster site by direct yourself to its main page in our App, or choose to navigate a route to the site. If you have already made a decision about where to go, you can also simply type in the name of the rollercoaster and get suggested routes to your destination. By choosing to plan a trip you will be provided with several options (automobile, walking, public transportation...) available so that you would approach your desired rollercoaster

destination easily.

There is one other function that the Map provides. Once you opened the map you can see the number one rollercoaster around you that is within the proximity of 300km (you can also adjust the scope to any distances you desire) at the bottom of your screen. You can scroll it leftward to see up to the tenth suggested rollercoaster around you. The ranking is automatically calculated by the logarithm provided by our App, and you can use it to determine where should you go that is not too far away but still highly rated. If you are planning to go to a trip and wants to ride a roller coaster during that trip, you can also type in the hotel/residence that you will be staying in during that trip, and the ranking of roller coasters around your trip destination will appear too.

## 10) Contribute

At the lower right corner of the Fig. 10. Our App is not perfect — it is impossible to acquire information about every single roller coaster in the world. To address that, we created a “contribute section”. In here, any person or entity (such as rollercoaster companies) can make an addition to our database of roller coasters by simply filling out relevant information, such as location, speed, height, of a roller coaster that isn’t in our database. Users can also report errors in our database or ranking. Then, our office would authenticate this information or, in cases which some information was wrong or is not filled out, we will conduct researches into that rollercoaster and filled incorrect information instead. Once the process is completed, the new roller coaster would be considered in our ranking and will appear in our map and database. The false information will also be updated to provide precise, authentic information to our users.

If a user isn’t satisfied with ranking and held strong disagreement against it, he/she may file a complaint or suggest something. The suggestion panel and form can be found on this page. Also, if someone wishes to get a more personalized result, he or she may take a survey regarding factors influencing the ranking of roller coasters. Once they’ve filled out their survey, our algorithm would more accurately capture his/her personal preference, therefore providing a more personalized ranking for each user. The survey result would also be transferred and stored in our database. Once more surveys are filled out, we will use these data to reexamine our algorithm. The algorithm will be adjusted accordingly as the users’ response and preference changes. For example, if our survey results show that there is an increasing amount of users who believe that the age of the roller coaster is the greatest determinant of the ride’s safety margin, then we would change our algorithm and increase the weighing of “age” of the safety margin factor.

## 11) Browse

Fig. 11 is showing the browse. There are three parts in the Browse. They are respectively Search, Library and Explore Genres. For search, just type in whatever you are searching for in the search within a particular category. You can tap the Amusement Park’s name, roller coaster’s name, the genre of the roller coasters, the material of the roller coasters, the location of the roller coasters, etc. The users can use search to research the roller coasters, which will make users more convenient.

For Library, there are Watch List and Recently Watched. In **FUN ROLLER** there are lots of fun videos about each roller coasters. Also, the users can upload the fun video on the **FUN ROLLER**. So after users watch the video. **FUN ROLLER** will save the history in the Recently Watched. Moreover, if the user, who is really interested in the video. The user can download it and put it in the Watch List. So next time, when the user wants to watch this video again, he

can find the video at the Watch List. And for Explore Genres, the users can find different kinds of the roller coaster in this section. For instance, inverted, suspended, sit down, steel, wood, flying. The users can find what types of the roller coaster is their favorite one. Then the users can click it and see more details.

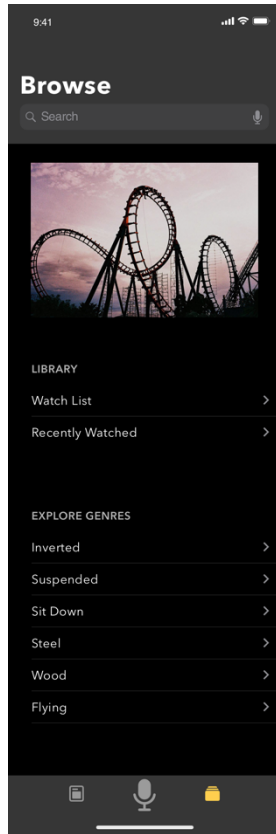


Fig. 11 Browse

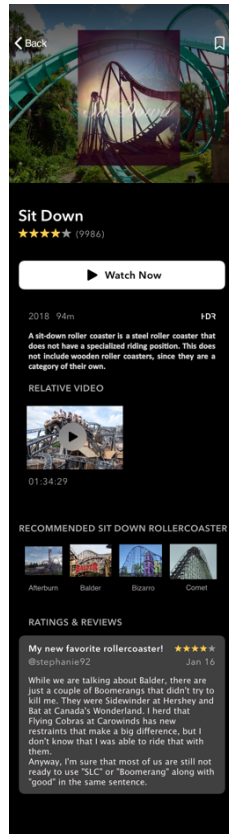


Fig. 12 Detail of the genres

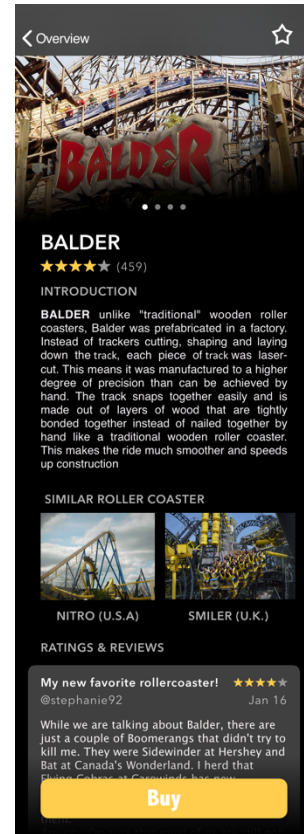


Fig. 13 Recommended system

## 12) Detail of the Explore Genres

Detail of the explore genres is at the Fig. 12. This slide is the detail of the Explore Genres. For example, there is a user, who are really interested with the Sit Down roller coaster. So he clicks SIT DOWN in the EXPLORE GENRE, then he can see that there are lots of the information about sit down roller coaster. Furthermore, there is the relative video of the sit-down roller. Of course, if the users have relative video, the users also can upload the video onto the **FUNROLLER**. There is the funny video, documentary, introduction video, live, etc. In addition, under the relative video, **FUN ROLLER** will have a recommended sit down rollercoasters to the users. We can according to the data and cookies that from your **INSTAGRAM** or **FACEBOOK** account to speculate the roller coasters genre that user's like. In addition, **INSTAGRAM** and **FACEBOOK** can share information with us. We can collect the forward statues of the users. Furthermore, we gather the user's following topic and what they like to calculate. Also, we can use cookies to judge the user's personality. Use user's personality to determine what kind of roller coasters suitable for users.

## 13) Fun Roller Coaster Video

After you click the video from the detail of the Explore genre. It will show a short video, documentary, experiment, etc. to the users. There is a control field at the bottom of the video.

The users may slide the progress bar to control the length of the video. In addition, the user can press the fast-forward button to move the video 15s. When the user wants to quit the video. The user may click the button locate at the top left corner. Moreover, the button that at top right corner can control the sound of the video.

#### 14) Comment System

At the bottom of the Fig. 12 there is a comment system for users. Users can add comments to the different kind of genres and roller coaster in the comment area. Users can express their own feeling and own experience of the roller coasts and genres. Moreover, there are interactions between the users and the users. Users can reply to other people's comments, to express their own opinions and altitude to the roller coasters. And each of the users can give a mark to the other people's comment. This comment system can improve the interactivity of the **FUN ROLLER**.

Furthermore, users can give the score to the roller coasters that recommended by **FUN ROLLER**. Users can give the stars out of five to evaluate the intelligent recommendation. If the recommendation is very accurate, which is very close to the roller coaster that the user like. The users may give the high evaluation to the **FUN ROLLER**. In contrast, if the intelligent recommendation makes some mistake, which the result make the user unpleasant or it is not the roller coasters that the users expected. The user may give a low evaluation to **FUN ROLLER**. According to the evaluation, users can have a better choice to choose roller coasters for themselves.

#### 15) FUN ROLLER Recommended the Roller Coaster

This part is showing at the Fig. 13. On this page, you can find a detailed description of the roller coaster. You will be introduced to a series of photos of the roller coasters that are uploaded by users by which you could scroll leftward to see more, and a detailed, objective (not user edited) description of the roller coaster, with its contents including but not limited to the height, type, speed, of the roller coaster so that you can have a holistic view about the roller coaster. In addition, the users can have a marking for this roller coaster. The users can choose the stars out of five stars. This will be a great suggestion for the next users.

After the users read the information of the roller coaster that **FUN ROLLER** recommended. If the users are interest with this roller coaster, the user can put this roller coaster in to the EduBank or the user may click the button under the information, to buy the ticket of the amusement park immediately. The users may get some discount if they buy the ticket on the **FUN ROLLER**.

Furthermore, at the bottom of the screen, we will recommend similar rollercoasters to users. This recommendation is built on user's preferences, view history, as well as rides similar to the one that you are viewing. Moreover, **FUNROLLER** will compare the data of the roller coaster to others. **FUNROLLER** will according to the HEIGHT, SPEED, TYPE, YEAR, LENGTH, CONSTRUCTION, INVERSIONS, DROP, DURATION, G-FORCE and VERTICLE ANGLE to determine the similar roller coasters to the users.

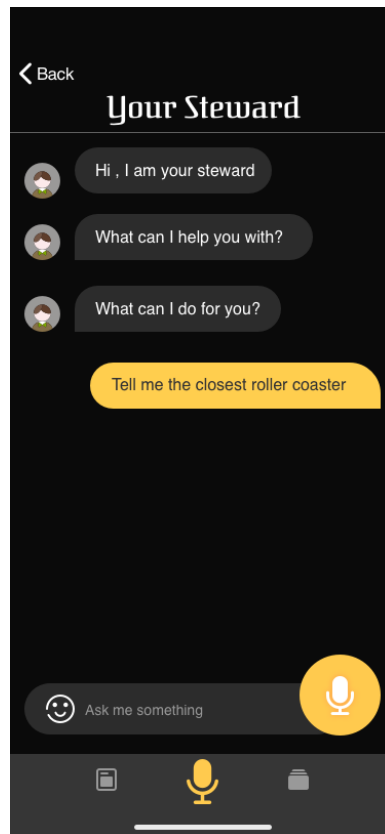


Fig. 14 Artificial intelligence bilateral feedback system

### 16) Artificial Intelligence Bilateral Feedback System

At Fig.14 is the artificial intelligence bilateral feedback system. On the bottom of the page there will be an icon that stands for this system. It is like a Siri that would help you to determine one's preferences for roller coasters. Users can talk to the system and tell them what kind of roller coaster do they want this time, and then a response will be given to the user without him/her clicking buttons or filling out surveys. It is called a bilateral feedback system because it not only gives the user the ranking they needed, the system will also capture keywords and determine the user's preference and characteristics, store them in our database, so that we would create a more accurate results to users. An example conversation would be:

“Hey, I want to go to a roller coaster that is thrilling, height and speed is no problem, but i just want the travel time to be within 2 hours and the ride to be more innovative. I want a roller coaster ride that surprises me.”

The system would then capture key words like thrilling, no problem with high and speed, innovative, and then provide a ranking of roller coasters within 2 hours ride that is calculated according to his personalized requests and preferences. These keywords will also be stored in our database for further reference.



## 9. Conclusion

In conclusion, our model is a smart, accurate model that achieves a state of balance and provide an impartial ranking for any potential users. The App fully uncovers the commercial value of this algorithm and makes it more accessible and precise for any users.

The most outstanding thing about our ranking is its accuracy. Even though the algorithm is different and fully based on objective data, our model yields similar results (rankings) compared to popular ranking sites. This reinforces the accuracy of our ranking. The algorithm incorporates the scores from three facets that would influence the overall experience of a roller coaster ride — safety, excitement and innovativeness, the score is each is acquired through sub-scores of factors under these facets. The percentage weighing of these scores into our final rating is hugely determined by a survey that we sent out — therefore truly represents the preferences of the actual potential users.

What makes out algorithm Unique are the methodologies that we used to derive our final weighing of each factors. The introduction and integration of double layered fuzzy algorithm ensures that the data we acquired from the survey not only represents potential users' priorities but is also accurate and reliable. In order to balance this subjective weighing, entropy weighting is used, therefore increasing the objectivity of the weighing of our logarithm. The most significant highlight of our project is this delicate balance between objective and subjective elements that we acquired through these two methods.

The UI that we designed contained a variety of creative elements — the use of customization and AI allows users to be sent the rankings and information they desired. The algorithm that we created would be adjusted automatically according to the traits that the user has shown. Furthermore, we envisioned a smart App that incorporates a variety of functions, that will guide the users easily and smoothing through every aspect of their roller coaster journey — through ticketing, searching, traveling, and rating. These useful, creative traits of the App would make our algorithm even better and more approachable to potential users.

## 10. Reference

- [1] [Ando and Zhang, 2005] Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.
- [2] [Berk, 2006] Berk, R. A. (2006). An introduction to ensemble methods for data analysis. *Socio- logical Methods and Research*, 34(3):263–295.
- [3] [Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag.
- [4] [Klementiev et al., 2008b] Klementiev, A., Roth, D., and Small, K. (2008b). Unsupervised rank aggregation with distance-based models. In *Proc. of the International Conference on Machine Learning (ICML)*.
- [5] [Liu et al., 2007] Liu, Y.-T., Liu, T.-Y., Qin, T., Ma, Z.-M., and Li, H. (2007). Supervised rank aggregation. In *Proc. of the International Conference on World Wide Web (WWW)*.

- [6] [Fligner and Verducci, 1986] Fligner, M. A. and Verducci, J. S. (1986). Distance based ranking models. *Journal of the Royal Statistical Society*, 48:359–369.
- [7] [Clemen and Winkler, 1999] Clemen, R. T. and Winkler, R. L. (1999). Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19(2):187 – 203.
- [8] [Cooke, 1991] Cooke, R. M. (1991). *Experts in Uncertainty: Opinion and Subjective Probability in Science*. Oxford University Press.
- [9] En.wikipedia.org. (2018). Amusement Today. [online] Available at: [https://en.wikipedia.org/wiki/Amusement\\_Today](https://en.wikipedia.org/wiki/Amusement_Today) [Accessed 19 Nov. 2018].
- [10] “Roller Coaster.” Wikipedia, Wikimedia Foundation, 18 Nov. 2018, [en.wikipedia.org/wiki/Roller\\_coaster](https://en.wikipedia.org/wiki/Roller_coaster).
- [11] Wenku.baidu.com. (2018).[online] Available at: <https://wenku.baidu.com/view/ea19c116ec630b1c59eef8c75fbfc77da369977f.html> [Accessed 19 Nov. 2018].

## 11. Appendix

Code (1):

```
clear
clc

%%=====OVERALL DATA READING=====
[data1,data2]=xlsread('weight.xlsx','overall');
[sx,sy]=size(data2);
AL=zeros(sx,sy);
AM=zeros(sx,sy);
AU=zeros(sx,sy);
[AL,AM,AU]=data_trans(data2)
%=====
m=3;
n=m*(m-1);
X=zeros(n,m);
Y=zeros(n,1);
B=[];
N=size(AM,1);
i=1;
for a=1:m-1
    for b=a+1:m
        X(i,a)=-1;
        X(i,b)=1;
        Y(i,1)=log(AM(a,b)/AL(a,b));
    end
end
```

#8874

```
B(i)=log(AL(a,b)^(-1));
i=i+1;
X(i,a)=1;
X(i,b)=-1;
Y(i,1)=log(AU(a,b)/AM(a,b));
B(i)=log(AU(a,b));
i=i+1;
end
end
Z=-eye(n);
A=[X,Y,Z];
[Am,An]=size(A);
M=100000;
fun=@(x)(1-x(N+1))^2+M*x(N+2:end)*x(N+2:end)';
x=fmincon(fun,rand(N*N+1,1)',A,B,[],[],zeros(N*N+1,1),2*ones(N*N+1,1));

W_1_1=[];
x_total=sum(exp(x(1:m)));
W_1_1=exp(x(1:m))/x_total;

%Excitement
%%=====OVERALL DATA READING=====
[data1,data2]=xlsread('weight.xlsx','excitement');
[sx,sy]=size(data2);
AL=zeros(sx,sy);
AM=zeros(sx,sy);
AU=zeros(sx,sy);

[AL,AM,AU]=data_trans(data2)
%=====

m=7;
n=m*(m-1);
X=zeros(n,m);
Y=zeros(n,1);
B=[];
N=size(AM,1);
i=1;
for a=1:m-1
    for b=a+1:m
        X(i,a)=-1;
        X(i,b)=1;
        Y(i,1)=log(AM(a,b)/AL(a,b));
        B(i)=log(AL(a,b)^(-1));
        i=i+1;
        X(i,a)=1;
        X(i,b)=-1;
        Y(i,1)=log(AU(a,b)/AM(a,b));
        B(i)=log(AU(a,b));
    end
end
```

#8874

```
        i=i+1;
    end
end
Z=-eye(n);
A=[X,Y,Z];
[Am,An]=size(A);
M=100000;
fun=@(x)(1-x(N+1))^2+M*x(N+2:end)*x(N+2:end)';
x=fmincon(fun,rand(N*N+1,1)',A,B,[],[],zeros(N*N+1,1),2*ones(N*N+1,1));

W_2_1=[];
x_total=sum(exp(x(1:m)));
W_2_1=exp(x(1:m))/x_total;

%Safety

AL=[1,1/4,1/3,1,1/3;
    2,1,2,4,1;
    1,1/4,1,2,1/3;
    1/3,1/6,1/4,1,1/5;
    1,1,1,3,1];

AM=[1,1/3,1/2,2,1/2;
    3,1,3,5,1;
    2,1/3,1,3,1/2;
    1/2,1/5,1/3,1,1/4;
    2,1,2,4,1];

AU=[1,1/2,1,3,1;
    4,1,4,6,1;
    3,1/2,1,4,1;
    1,1/4,1/2,1,1/3;
    3,1,3,5,1];

m=5;
n=m*(m-1);
X=zeros(n,m);
Y=zeros(n,1);
B=[];
N=size(AM,1);
i=1;
for a=1:m-1
    for b=a+1:m
        X(i,a)=-1;
        X(i,b)=1;
        Y(i,1)=log(AM(a,b)/AL(a,b));
        B(i)=log(AL(a,b)^(-1));
        i=i+1;
        X(i,a)=1;
        X(i,b)=-1;
    end
end
```

#8874

```
        Y(i,1)=log(AU(a,b)/AM(a,b));
        B(i)=log(AU(a,b));
        i=i+1;
    end
end
Z=-eye(n);
A=[X,Y,Z];
[Am,An]=size(A);
M=100000;
fun=@(x)(1-x(N+1))^2+M*x(N+2:end)*x(N+2:end)';
x=fmincon(fun,rand(N*N+1,1)',A,B,[],[],zeros(N*N+1,1),2*ones(N*N+1,1));
```

```
W_2_2=[];
x_total=sum(exp(x(1:m)));
W_2_2=exp(x(1:m))/x_total;
```

%Innovativeness

%%=====Innovativeness

DATA

READING=====

```
[data1,data2]=xlsread('weight.xlsx','Innov');
```

```
[sx,sy]=size(data2);
```

```
AL=zeros(sx,sy);
```

```
AM=zeros(sx,sy);
```

```
AU=zeros(sx,sy);
```

```
[AL,AM,AU]=data_trans(data2)
```

%%=====

```
m=4;
```

```
n=m*(m-1);
```

```
X=zeros(n,m);
```

```
Y=zeros(n,1);
```

```
B=[];
```

```
N=size(AM,1);
```

```
i=1;
```

```
for a=1:m-1
```

```
    for b=a+1:m
```

```
        X(i,a)=-1;
```

```
        X(i,b)=1;
```

```
        Y(i,1)=log(AM(a,b)/AL(a,b));
```

```
        B(i)=log(AL(a,b)^(-1));
```

```
        i=i+1;
```

```
        X(i,a)=1;
```

```
        X(i,b)=-1;
```

```
        Y(i,1)=log(AU(a,b)/AM(a,b));
```

```
        B(i)=log(AU(a,b));
```

```
        i=i+1;
```

```
    end
```

```
end
```

```
Z=-eye(n);
```

```
A=[X,Y,Z];
```

```
[Am,An]=size(A);
```

#8874

```
M=100000;
fun=@(x)(1-x(N+1))^2+M*x(N+2:end)*x(N+2:end)';
x=fmincon(fun,rand(N*N+1,1)',A,B,[],[],zeros(N*N+1,1),2*ones(N*N+1,1));

W_2_3=[];
x_total=sum(exp(x(1:m)));
W_2_3=exp(x(1:m))/x_total;

%=====
%Overall Weight

Data=xlsread('last_data.xlsx','final_data','A1:P300');
Alldata=[];
for i=1:300
    Alldata(i,1)=i;
    for j=1:16
        if Data(i,j)>1
            Data(i,j)=1;
        elseif Data(i,j)<-1
            Data(i,j)=-1;
        end
        Alldata(i,j+1)=Data(i,j);
    end
end
end

Score=[];

W_shangquan=[0.0493872941721330,0.112778409598995,0.0842470978328826,0.1387544
47173779,0.189411442897627,0.124165708896747,0.0951614034327795,0.1071315060190
72,0.0989626899759948];

W_2=[W_2_1*W_1_1(1),W_2_2*W_1_1(2),W_2_3*W_1_1(3)];

W_final([1,8,13])=W_2([1,8,13]);

W_shengyu=1-sum(W_final([1,8,13]));

W_shangquan=W_shangquan*W_shengyu;

W_final(2)=W_shangquan(3)*W_1_1(1)/(W_1_1(1)+W_1_1(2))*0.2+W_2(2)*0.8;

W_final(3)=W_shangquan(4)*0.2+W_2(3)*0.8;

W_final(4)=W_shangquan(5)*W_1_1(1)/(W_1_1(1)+W_1_1(3))*0.2+W_2(4)*0.8;

W_final(5)=W_shangquan(6)*0.2+W_2(5)*0.8;

W_final(6)=W_shangquan(8)*0.2+W_2(6)*0.8;

W_final(7)=W_shangquan(9)*W_1_1(1)/(W_1_1(1)+W_1_1(2))*0.2+W_2(7)*0.8;
```

#8874

```
W_final(9)=W_shangquan(1)*W_1_1(2)/(W_1_1(2)+W_1_1(3))*0.2+W_2(9)*0.8;
W_final(10)=W_shangquan(2)*0.2+W_2(10)*0.8;
W_final(11)=W_shangquan(3)*W_1_1(2)/(W_1_1(1)+W_1_1(2))*0.2+W_2(11)*0.8;
W_final(12)=W_shangquan(9)*W_1_1(2)/(W_1_1(1)+W_1_1(2))*0.2+W_2(12)*0.8;
W_final(14)=W_shangquan(1)*W_1_1(3)/(W_1_1(2)+W_1_1(3))*0.2+W_2(14)*0.8;
W_final(15)=W_shangquan(5)*W_1_1(3)/(W_1_1(1)+W_1_1(3))*0.2+W_2(15)*0.8;
W_final(16)=W_shangquan(7)*0.2+W_2(16)*0.8;

Score=Alldata(:,2:17)*W_final';

[~,Index]=sort(-Score(:,1));
Index_top50=Index(1:50);

%Sensitive analysis
ALLRANK=[];
for i=1:16
    W_final(i)=W_final(i)*1.2;
    Sum=sum(W_final);
    W_final=W_final/Sum;
    Score=Alldata(:,2:17)*W_final';
    [~,index]=sort(-Score(:,1));
    ALLRANK(1:300,i)=index;
end
```

Code (2):

```
function [al,am,au]=data_trans(data2)

[sx,sy]=size(data2);

al=zeros(sx,sy);
am=zeros(sx,sy);
au=zeros(sx,sy);

for i=1:sx
    for j=1:sy
        if (j ~= i)
            t = data2(i,j);
            tt=t{1};
            pos=strfind(tt,',');
            len=length(tt);
            al(i,j)=str2num(tt(1:pos(1)));
```

#8874

```
    am(i,j)=str2num(tt(pos(1)+1:pos(2)-1));  
    au(i,j)=str2num(tt(pos(2)+1:len));  
  
    else  
        al(i,j)=1;  
        am(i,j)=1;  
        au(i,j)=1;  
    end  
end  
end  
end
```