|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | For office use only | | | T1 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | T2 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | T3 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | T4 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |  | | --- | | Team Control Number **8784** | |  | | Problem Chosen **A** | | |  |  | | --- | --- | | For office use only | | | F1 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | F2 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | F3 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | F4 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |   **2018 HiMCM Summary Sheet** |

Blablablablabla

**Key Words:** Roller Coasters, Principle Component Regression, Bayes Distinction, BP Neural Network Fitting, XG Boosting algorithm

**News Release: Hop till you Drop**

**----Team 8784’s New Techniques Shed Light on**

**Unique Roller Coaster Experience**

Have you ever been in want of going to an amusement park and ride the roller coasters? Have you ever be troubled by the issue that you do not know which roller coaster is the most suitable for you? When you are planning to ride the roller coasters, have you once wondered to have a scoring system that can have an objective rating system towards all the roller coasters around the globe which is not affected by personal opinions or perceptions? If this is the case, you do not need to despair anymore. Fortunately for you, team 8784 has skillfully addressed the concern, who invented a set of algorithms which specially deal with the rating of roller coasters. Don’t forget that taking the roller coasters is the obsolescence for almost everyone. With the rating of roller coasters, you are able to reap the most overwhelming sense when riding them!

Based on 300 roller coasters all over the world, team 8784 utilized several techniques to evaluate each one of them eloquently. They take diverse accounts into consideration, not only the standards that can flash into your mind, such as the speed, height, or the number of inversions, but also some factors that are less concerned but play a significant role in the consideration of people, including the type of coasters, even the material that the coasters are made from. Their algorithms are also well-considered, having a notable performance in different kinds of variables. They employed the cutting-edge BP Neural Network, simulating the principle of human brains and achieve an accuracy of over 99%. They applied XG Boosting algorithm to synthesize the various methods to the problem, of which the fundamental theory is that a complicated issue can be better estimated when synthesizing the judgment of each expert than that of a sole expert. Hence, their solution is credible and reliable.

We believe that you are eager to know which roller coaster leads the rank, and here it comes. The one which is on the top of the list names T Express locates in Everland Park, Yongin-si, Gyeonggi-do, South Korea, followed by roller coasters in France, the US, China, and Japan. The best roller coaster in the US is Apocalypse Six Flags America opened in 2012, lying in Upper Marlboro Maryland with a 90-feet drop, 100.0 feet high, 55.0 mph speed, 2900.0 feet long. A single loop costs approximately 2 minutes. No matter you want to run after the best roller coasters or have a suburban trip, the new ranking will never let you drop.

Want to personalize your recommendation? Don’t hesitate to download the newly-designed app given by team 8784. It features several functions. From the best roller coasters in the world to the best coasters only for your interest, from the enumeration of the entire roller coasters to the search engine of your roller coasters, you can find whatever you want in the application. Based on big data of all the users and your personal information, it can personalize your preference and dynamically adjust the recommendation just for you. You can also search the parameters of the roller coasters if you want.

No matter whether you are a Spartan or a spare time traveler, no matter whether you are a crazy fancier or a casual visitor, as long as you come and visit, a sea of roller coasters will be unveiled in front you. With the guide from team 8784, you will definitely be overwhelmed with the bliss on the top, abuzz with the loop, and hop till you drop.

**Contents**

1. **Background**
   1. **Research Background**
   2. **Restatement of the Problem**
   3. **Research Method and Train of Thinking**
2. **Assumptions**
   1. **Assumptions**
   2. **Definitions**
3. **Data Procurement and Process**
   1. **Data Cleaning and Interpolation**
   2. **Cluster**
   3. **Ideal Solution**
   4. **Principal Component Analysis**
4. **Modeling**
   1. **Basic Statistics**
   2. **AHP Method**
   3. **Linear Regression**
   4. **KNN Algorithm**
5. **Optimization**
   1. **Principal Component Regression**
   2. **Bayes Distinction**
   3. **BP Neural Network Fitting**
   4. **XG Boosting Algorithm**
6. **Comparison of the top 10 Roller Coasters**
7. **Concept and design for a user-friendly app**
   1. **Initial Recommendation**
   2. **Recommendation Base on Preference**
   3. **Search Engine for Roller Coasters**
   4. **Auxiliary Functions**
8. **Analysis**
   1. **Sensitivity Analysis**
   2. **Strength and Weakness**
9. **Reference**
10. **Appendix**
    1. **MATLAB Code**
    2. **PYTHON Code**
    3. **Final Result**
11. **Background**
    1. **Research Background**

Roller coaster is an exciting entertainment fascinated by many of today's youngsters because of its stimulation and pleasure. However, the rating system of roller coasters is relatively scarce and mainly based on people's own experience, lacking quantitative analysis of roller coasters' different traits. For instance, Coaster Buzz posted the current top 100 roller coasters on its website, but the rating process is largely based on its members' track records and subjective experience inputs. Admittedly, one of the method's main advantages proves to be the vast sample size and the ratings' rigorous selection in order to exclude the anomalies. But even with the rating results refreshed weekly, the poll on the internet only reflects the opinion of any one person and the riders who provide their experience and scores may mostly come from the same region, so the ranking is not based on the world's scale and will certainly still lose some great rides. Therefore, most rating methods are highly unstable and unconvincing, making the roller coaster-choosing process inaccurate and the riders being dissatisfied. According to the current needs and lack of quantitative methods of rating, a proper method for ranking the roller-coasters is in dire need.

* 1. **Restatement of the Problem**

The question is based on the fact that nowadays roller coaster ranking systems are largely dependent on riders' own subjective inputs, with few considering the roller coasters' own properties. Providing us with the basic information of 300 roller coasters around the world, the question asked us to decide the top ten roller coasters using quantitative assessing methods, compare them with other methods currently being used and analysis the strength and weakness. Besides, we are required to develop the concept of a user-friendly APP which aims to help the potential riders finding the proper roller coasters that will satisfy their needs. Finally, we write a News Release to publicize our quantitative methods, the result of top-10 roller coasters based on the data given and the concept of our newly-designed APP.

* 1. **Research Method and Train of Thinking**

We do the data cleaning first and interpolate the missing data, while extracting useful and relevant data and conducting basic statistics for further research. Next, we come to the data procurement part to examine whether the score of the roller coasters online can be a training set of our model with the help of Principal Component Analysis. Then we apply the results of data procurement for modeling. In the modeling process, we apply results from Principal Component Analysis to the Analytical Hierarchy Process, KNN, and Linear Regression. At this point, we have reached the conclusion of the rank of different independent factors. Furthermore, we conduct optimization for each model. We optimize KNN by Bayes Distinction, optimize Linear Regression by Principal Component Regression, and utilize BP Neural Network Fitting to achieve higher accuracy. Afterward, we employ the XG Boosting algorithm to synthesize the three methods and reach a conclusion over the ranking. Finally, we compare our rating and raking with those online results and design the notion of our desired application. Figure 1 below presents the whole modeling process, and if the method is marked red, it indicates the result of this analysis is not applied to further modeling and optimization.

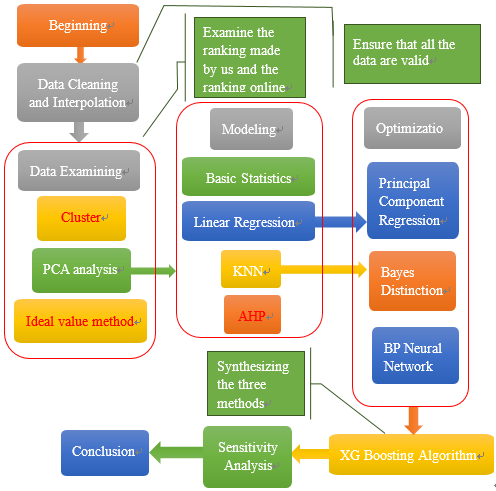
****

Figure 1: The flow chart of the whole modeling process

1. **Assumptions**
   1. **Assumptions**

* All the data are credible and reliable, which means they have no error.
* The parameters of the roller coasters are constant and do not change at a different time in a day. For instance, the speeds of the roller coasters are always the value given in the table. No matter it is in the morning or evening, or no matter how many people there are on the roller coasters, they will always travel at the speeds given.
* Except the parameters given in the data, all other properties of the roller coasters are exactly the same, which indicates it has no impact on the final score to the riders.
  1. **Definitions**

Table 1: the definition of notations

|  |  |
| --- | --- |
| Notation | Definition |
|  | The element in Row and Column in matrix |
|  | The independent variables matrix |
|  | Row vector of independent variables |
|  | Row vector of dependent variables |
|  | The algebra average of several data |
|  | The Mahalanobis distance of the data |
|  | The covariance matrix |
|  | The original variable |
|  | The New variable |
|  | The number of samples |
|  | The number of variables in each sample |
|  | The standardized data at row and column |
|  | The data at row and column before standardization |
|  | The correlation coefficient matrix in principal component analysis |
|  | The characteristic roots or eigenvalues in Weight determination Technique |
|  | The characteristic vectors |
|  | The th value of the characteristic vectors |
|  | Weight vector in AHP |
|  | The number of choices of target layer in AHP |
|  | The eigenvector in AHP |
|  | Coefficient matrixes of the original data |
|  | Coefficient matrixes of Principal Component Regression |
|  | The probability that satisfies condition |
|  | Reliability in Regression |
|  | Parameters to be estimated of the ensemble in Regression |
|  | The confidence upper limit in Regression |
|  | The confidence lower limit in Regression |
|  | Posteriori probability in Bayes Distinction |
|  | Priori probability in Bayes Distinction |
|  | The frequency at which the sample appears in Bayes Distinction |
|  | The ensemble in Bayes Distinction |
|  | Probability density function of in Bayes Distinction |
|  | The priori probability of In Bayes Distinction |
|  | The number of in Bayes Distinction |
|  | The conditional probability of wrongly categorizing the sample of to the ensemble |
|  | The loss caused by the wrong categorization |
|  | A division of a set of distinction samples |
|  | The average wrong distinction loss |
|  | The overall loss of each classifier |
|  | Classification function |
|  | function of each classifier to reduce the loss |
|  | The score of the data to show the accuracy of the prediction |

1. **Data Procurement and Process**
   1. **Data Cleaning and Interpolation**

Original Data

Speed (mph): 74.0

Duration (min:sec): 3:00

Name: Big One

Type: Sit Down

Drop (feet): 205.0

Construction: Steel

Park: Blackpool Pleasure Beach

Geographic Region: Europe

Status: Operating (discarded due to no variation)

Number of Inversions: 0

City/Region: Blackpool

Inversions (YES or NO): NO (discarded due to already contained)

Height (feet): 213.0

City/State/Region: Lancashire, England

Year/Date Opened: 1994

G Force: 3.5 (discarded due to too many missing data)

Country/Region: United Kingdom

Length (feet): 5497.0

Length (feet): 5497.0

Length (feet): 5497.0

Vertical Angle (degrees): 65 (discarded due to too many missing data)

Duration (sec): 180 (Interpolation if needed)

Number: 32 (added by given order)

Construction: 1

Geographic Region: 2

Height (feet): 213.0 (Interpolation if needed)

Speed (mph): 74.0 (Interpolation if needed)

Length (feet): 5497.0 (Interpolation if needed)

Drop (feet): 205.0 (Interpolation if needed)

Type: 1

Figure 2: Data Processing diagram

Finally, we obtain 9 variables that we mainly use, which are Geographic Region, Construction, Type, Year Opened, Height, Speed, Length, Duration, and Number of Inversions. The columns that are not mentioned above are regarded as the identification of each roller coaster, which will not be used for modeling. The 293 data after cleaning can be seen in the appendix. The following figure 2 illustrates the process mentioned above.

As we downloaded the data, we first number the 300 roller coasters from 1 to 300 and do the data cleaning as the foundation of the entire model. We remove the drop column, the G Force column and the Vertical Angle column since there are more than a half of the data missing, which renders it void for us to interpolate the missing value. We also remove the status column, since all the roller coasters are operating. Then with the help of XLRD and XLWR module in PYTHON, we convert the expression of the duration cells from both the minutes and seconds into seconds only. We also enumerate the Geographic Region column, the Construction column, and the Type column. For the Geographic Region column, we employ 1 to 8 represent Asia, Europe, North America, Central America, South America, Middle East, Oceana, and Russia respectively. For the Construction column, we use 1 to 2 represent steel and wood respectively. For the Type column, we use 1 to 6 represent sit down, inverted, stand up, suspended, flying, and wing respectively. We also notice that some of the Type cells are filled in steel or wood, which is not a possible choice of Type, which we use 0 to represent the two choices. We removed the unit in the cells of Height in order that it is able to be dealt with further.

We discover that some of the data in the Height, Speed, Length, and Duration column are missing; thus we consider that we use the interpolation method to fill in the missing number. We examine the correlation coefficients between the columns, and find that the correlation coefficient between Height and Speed is 0.836280084187907, and the one between Length and Duration column is 0.619704366781674, indicating that the two groups of column reveal a strong tendency of correlating, which means we can use the two columns in each group to interpolating the missing data of each other. We sort the interpolating variable and calculate the arithmetic means of the interpolated variable if an interpolating variable refers to more than one interpolated variable in the given data set before we utilize Piecewise Cubic Hermite Interpolation to interpolate our variable. We do the same process for the rest three columns and fill in all the data.

The reason why Piecewise Cubic Hermite Interpolation is suitable for our problem is that it avoids the oscillation between the point series, while we do not pay much attention to the smoothness of the interpolation function. We eliminated some data that miss both interpolating variables and interpolated variable.

Finally, we obtain nine variables that we mainly use, which are Geographic Region, Construction, Type, Year Opened, Height, Speed, Length, Duration, and Number of Inversions. The columns that are not mentioned above are regarded as the identification of each roller coaster, which will not be used for modeling. The 293 data after cleaning can be seen in the appendix. The following figure 2 illustrates the process mentioned above.

* 1. **Cluster**

We want to rank the roller coasters at the beginning and compare the ranking produced by our method with the ranking of the scoring system online. If these two are similar, we can regard the online scoring system as a learning set and establish a model to rate all the roller coasters. [1]

We utilize cluster to decide which of the roller coasters are similar. It can be predicted that similar roller coasters are more likely to have a similar rating and ranking; therefore we can divide all the roller coasters into several groups. If the roller coasters that are in the same group are more likely to lie in the same online score interval, such as the high score or the low score, we can determine that our ranking system is consistent with the online scoring system, which makes it viable for us to establish a model with the online system.

We use Mahalanobis distance for clustering and draw the dendrogram. The formula is as the following formula 1.



(1)

Among the formula, and denote two row vectors; denotes the covariance matrix; denotes the obtained Mahalanobis distance of the data. The result is shown in figure.

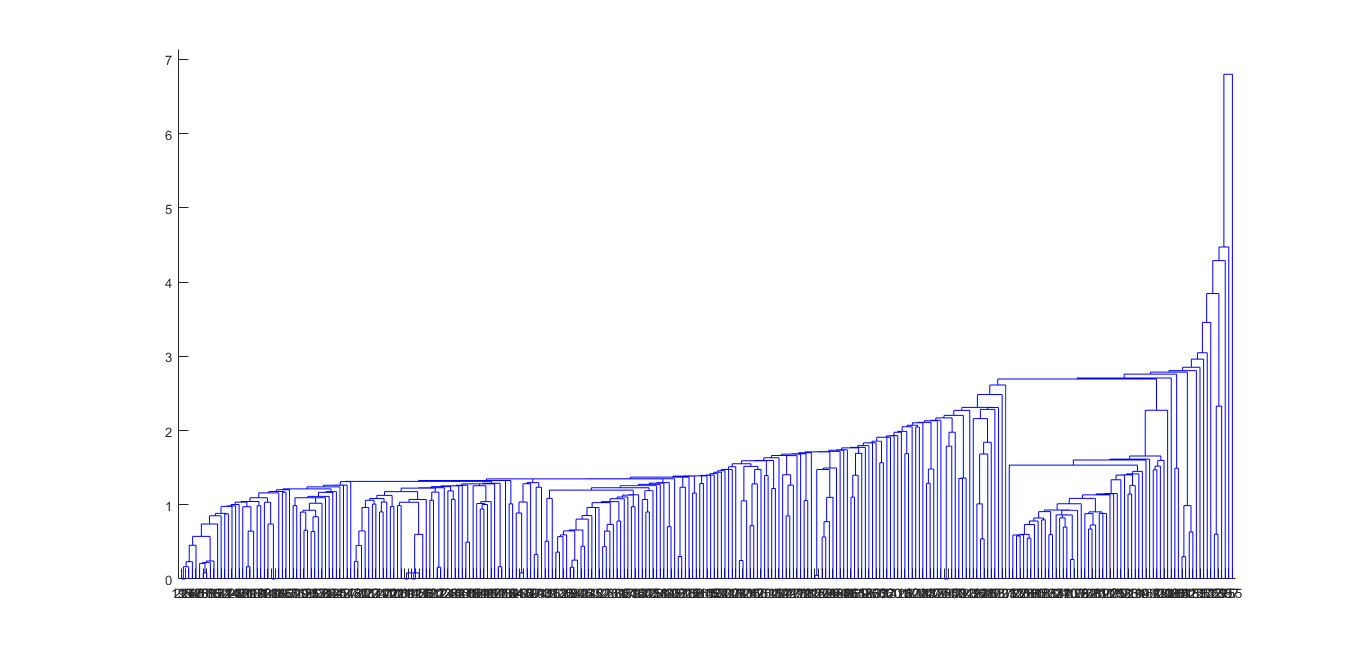


Figure 3: Mahalanobis Clustering Dendrogram, some of the categories contain too less roller coasters

From figure 3 above, we can see that some of the categories contain too fewer roller coasters, which shows that this method is complicated to set the roller coasters apart. Hence, we consider using other methods.

* 1. **Ideal Solution**

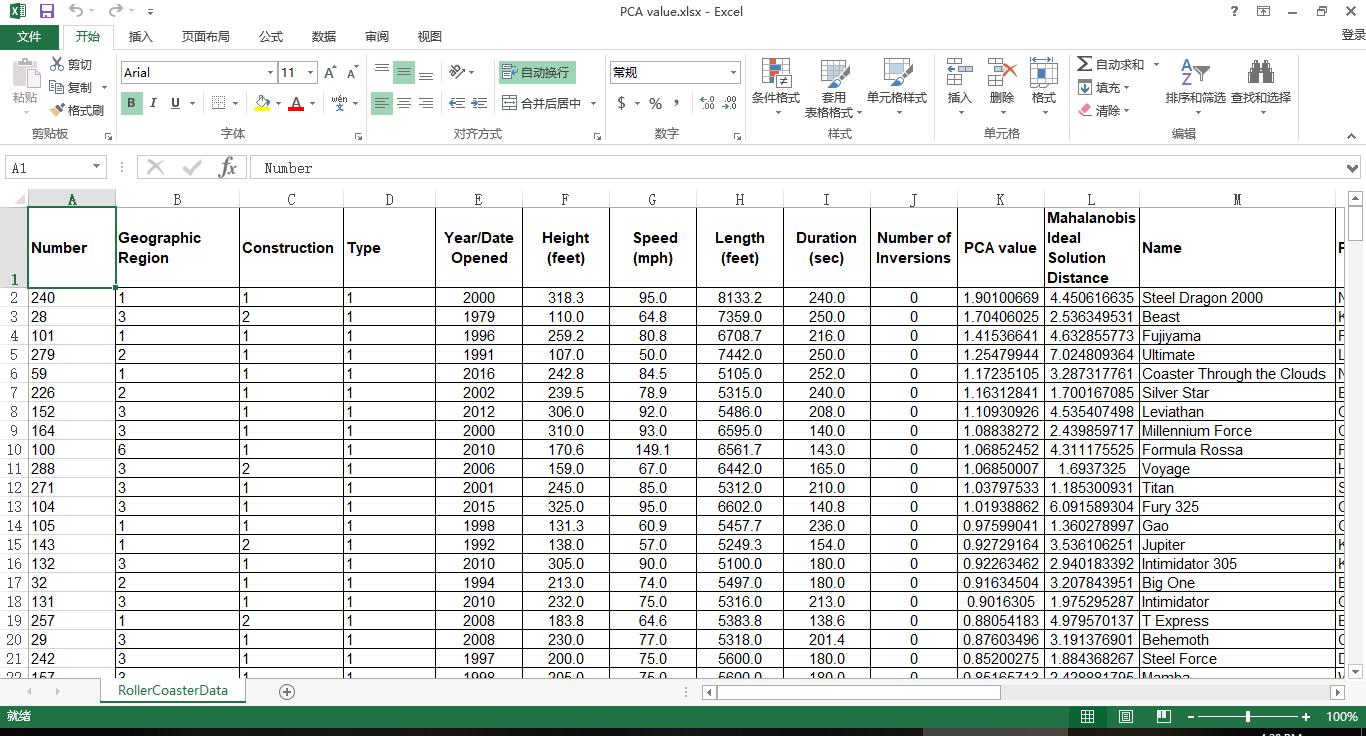
We also come up with a way that enables us to get the maximum value of year opened, height, speed, duration, length, and number of inversions in the data, setting them as an ideal solution. We then calculate the Mahalanobis distance between each data and the ideal solution, taking advantage of the avoidance of the effect of the dimension. The formula 2 is shown following:



(2)

Among the formula, and denote two row vectors; denotes the covariance matrix; denotes the obtained Mahalanobis distance of the data. Part of the results is shown in table 2, the rest of which are in the appendix.

Table 2: Ideal Solution result



However, we find that it is flawed for us to set the highest score as the ideal one with no direct evidence supporting. Therefore, we still need to consider other methods.

* 1. **Principal Component Analysis**

With the help of PCA, we are able to rank the roller coasters.

We utilize the 9 original variables mentioned in 3.3 as the original data. We still use to denote independent variables matrixes and the dependent variables. The original variables are ; the new variables are . We use to denote the number of samples and use to denote the number of variables in each sample. Thus, the data matrix is as matrix 3[3]

(3)

Since the data vary in dimensions and ranges, we need to standardize the data. We adopt the variance standardization technique to operate the data so that the variance of the standardized data is 1, while we conduct the central translation so that the mean of the data is 0. The formula is shown as formula 4-5

denotes the standardized data at row and column ; denotes the data at row and column before standardization. denotes total column number and denotes total row number.

Then we establish the correlation coefficient matrix . The formulas are shown in formula 6-7.

(4-5)



(7)

(6)

Then we obtain the characteristic vectors which satisfy for and characteristic vectors to determine the load on each new principal component variables of the original variables , which are equal to the larger characteristic values of the correlation matrix corresponding to the eigenvectors. is the value of the characteristic vectors. The formula is as formula 8:

(8)

In the formula, denotes each characteristic vector, denotes each characteristic value. The characteristic roots are shown in table 3. Characteristic vector matrix is in the appendix.

Table 3: Principal Component Analysis Characteristic Value

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.045393 | 0.232633 | 0.438638 | 0.607466 | 0.853558 |
| 1.011198 | 1.468301 | 1.707678 | 2.635135 |  |

The contribution rate formula and the total contribution rate formula is as formula 9-10.

and

(9-10)

We obtain the total contribution rate until the fifth principal component is 85.29%, which is larger than 85%. Therefore, we take the first fifth eigenvalue as the principal component. Suppose the principal component is formula set 11

(11)

In accordance with the first 5 scores of the principal component, we use the contribution rate as the weight and obtained the total score of each of the 293 roller coasters. Ranking the roller coasters, we put the top 5 in table 4, and the rest of roller coasters can be seen in the appendix.

Table 4: First 5 roller coasters of PCA analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number** | **Geographic Region** | **Construction** | **Type** | **Year/Date Opened** | **Height (feet)** |
| 240 | 1 | 1 | 1 | 2000 | 318.3 |
| 28 | 3 | 2 | 1 | 1979 | 110.0 |
| 101 | 1 | 1 | 1 | 1996 | 259.2 |
| 279 | 2 | 1 | 1 | 1991 | 107.0 |
| 59 | 1 | 1 | 1 | 2016 | 242.8 |
| **Number** | **Speed (mph)** | **Length (feet)** | **Duration (sec)** | **Number of Inversions** | **PCA value** |
| 240 | 95.0 | 8133.2 | 240.0 | 0 | 1.901007 |
| 28 | 64.8 | 7359.0 | 250.0 | 0 | 1.70406 |
| 101 | 80.8 | 6708.7 | 216.0 | 0 | 1.415366 |
| 279 | 50.0 | 7442.0 | 250.0 | 0 | 1.254799 |
| 59 | 84.5 | 5105.0 | 252.0 | 0 | 1.172351 |

Searching the top roller coasters online in our ranking [2], we find that all of the top 10 roller coasters online ranked the top one-third of our ranking. Several top 10 coasters online are in the top 20 coasters provided by us. Thus it shows that the result online can be used as the training set. We download the scores from Costerbuzz [2] and use them for further modeling.

1. **Modeling**
   1. **Basic Statistics**

After obtaining the original data, we do the basic statistics process. We download the score from the website, Coaster buzz, and set it as the dependent variables, while the variables given in the chart as independent variables. On the one hand, we make pie charts, as well as line charts, reveal the proportions of the roller coasters with each characteristic over the ensemble, as shown in figure 4-5.

Figure 4: Line Chart of Duration. The Duration focus on 100-150 seconds interval.

Figure 5: Pie Chart of Geographic Region. The roller coaster from North America takes a major proportion.

The previous charts demonstrate, for instance, that most of the given roller coasters locate in North America. The duration concentrates in 100-200 seconds interval.

* 1. **Analytical Hierarchy Process**

In order to determine the weight among diverse factors and judge the condition of roller coasters, we utilize the Analytic Hierarchy Process (AHP) to achieve the goal and determine the weight of each option in complicated and uncertain problems. We define each roller coasters as scheme layer, the 9 properties of the roller coasters as the standard layer, and the scores as the target layer to build up the 3-layer AHP model. The structure diagram is shown in the following figure 6. [4]

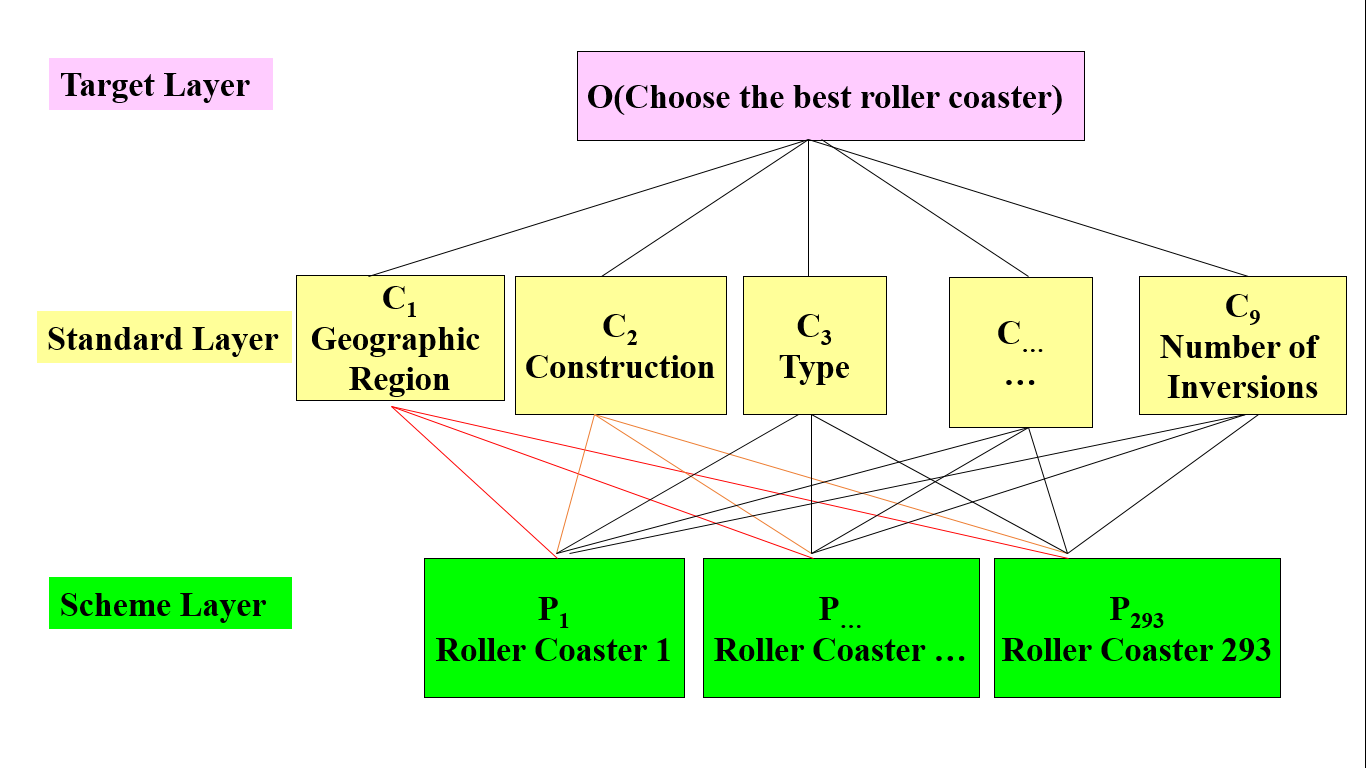


Figure 6: Structure diagram.

First, we define the number of roller coasters that possess certain properties under certain types of properties as , which refers to the amount of a certain target choice under a certain scheme layer condition. In accordance with the target choice, we obtain a weight vector ( stands for the number of choices of target layer). We compute the ratio between the number, , of each scheme layer choice under a common target layer choice and regard it as the weight of paired comparison matrix. As they are consistent matrixes, we do not need to apply consistency tests to the matrixes, for they are automatically consistent, which means that the eigenvalues are all identical. With the help of the formula of the eigenvalue and eigenvectors shown in formula 12,

(12)

we can obtain the eigenvectors, . The following tables 5-6 respectively shows the paired comparing matrix and eigenvector.

Table 5: Paired Comparing matrix from standard layer to target layer

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2.866024 | 2.306462 | 1.349377 |
| 0.348915 | 1 | 0.80476 | 0.470819 |
| 0.433565 | 1.242606 | 1 | 0.585042 |
| 0.741082 | 2.12396 | 1.709278 | 1 |

Table 6: Paired Comparing matrix from standard layer to object layer

|  |
| --- |
| 0.396265 |
| 0.138263 |
| 0.171807 |
| 0.293665 |

Then we repeat the process from standard layer to scheme layer, compose the eigenvalues of each scheme, and obtain a matrix of weight vector from scheme layer to standard layer. Multiplying the two weight matrixes, we obtain the final weight matrix, which is the weight vector from scheme layer to target layer.

To define the paired comparison matrix from the standard layer to the object layer, we calculate the correlation coefficients between the online score and each given standard of data and the cross-ratio between the correlation coefficients. We discover that the possible value of Geographic Region varies too less, which means there are only two values that are different from the rest in the data with online score. The numbers of inversions exist too much zeros. The correlation coefficients of construction, type, and duration are too low for further analysis. Thus we merely take four standards to do further analysis, which are Year, Height, Speed, and Length, discarding the rest variables.

Finally, we draw the statistical chart with each weight vector, such as scatterplot, to clearly express the weight of the result of the roller coasters. The charts are shown in figure 7.

Figure 7: Result analysis. The weight of the result is irregular.

We can clearly see that the weight of each roller coaster is irregular and since the irregularity may result from the low correlation coefficients, we need to consider a better method to solve the problem.

* 1. **Linear Regression**

The third modeling method we use is Linear Regression. We can regard the properties of roller coasters as independent variables, and the online scores as dependent variables. Based on the samples, each data can be viewed as a mapping from the independent variables, which are the properties, to the dependent variables, which are scores online. As each information is expressed numerical, we can find the function from the independent variables to the dependent variables through linear regression from the data. [5]

Let to respectively denote the nine properties respectively. Let denotes online scores. The value of the independent variables and dependent variables is the numbers of each option. We utilize regression formula 13.

(13)

Let denotes the independent variables matrix; denote dependent variables matrix; denotes coefficient matrixes. We apply Least Square Regression Method to the issue, of which the formula is shown in formula 14:

The formula is set to solve out the value of the coefficient matrixes of point estimation. With MATLAB giving solution, we obtain the coefficient matrixes which are presented in table 7:

(14)

Table 7: Linear Regression Coefficient

|  |  |
| --- | --- |
|  | -18.2625 |
|  | 0.089799 |
|  | 0.08964 |
|  | 0.017563 |
|  | 0.010858 |
|  | -0.00166 |
|  | 0.006959 |
|  | 8.52E-05 |
|  | -0.00053 |
|  | -0.01581 |

Point estimation possesses a drawback that it cannot express the accuracy of the data obtained. Thus we utilize interval estimation to reuse the Least Square Regression Method, the formula as in formula 15:

(15)

(16)

denotes the parameters to be estimated of the ensemble; denotes probability; denotes Confidence upper limit; denotes Confidence lower limit; denotes reliability which satisfies . In this way, we obtain formula 16

With the MATLAB program, we set as 0.95, under which the regression coefficient bound is shown in table 8.

The residual graph is shown in figure 8. When examining correlation coefficients, we find the correlation coefficients are 0.336705.

Table 8: Linear Regression Coefficient Bound

|  |  |  |
| --- | --- | --- |
|  | Lower Bound | Lower Bound |
|  | -28.9973 | -7.52761 |
|  | -0.35907 | 0.538668 |
|  | -0.07901 | 0.258286 |
|  | -0.04829 | 0.083415 |
|  | 0.005448 | 0.016268 |
|  | -0.00453 | 0.001203 |
|  | -0.00717 | 0.021093 |
|  | -6.05E-07 | 0.000171 |
|  | -0.00273 | 0.00167 |
|  | -0.04853 | 0.01691 |

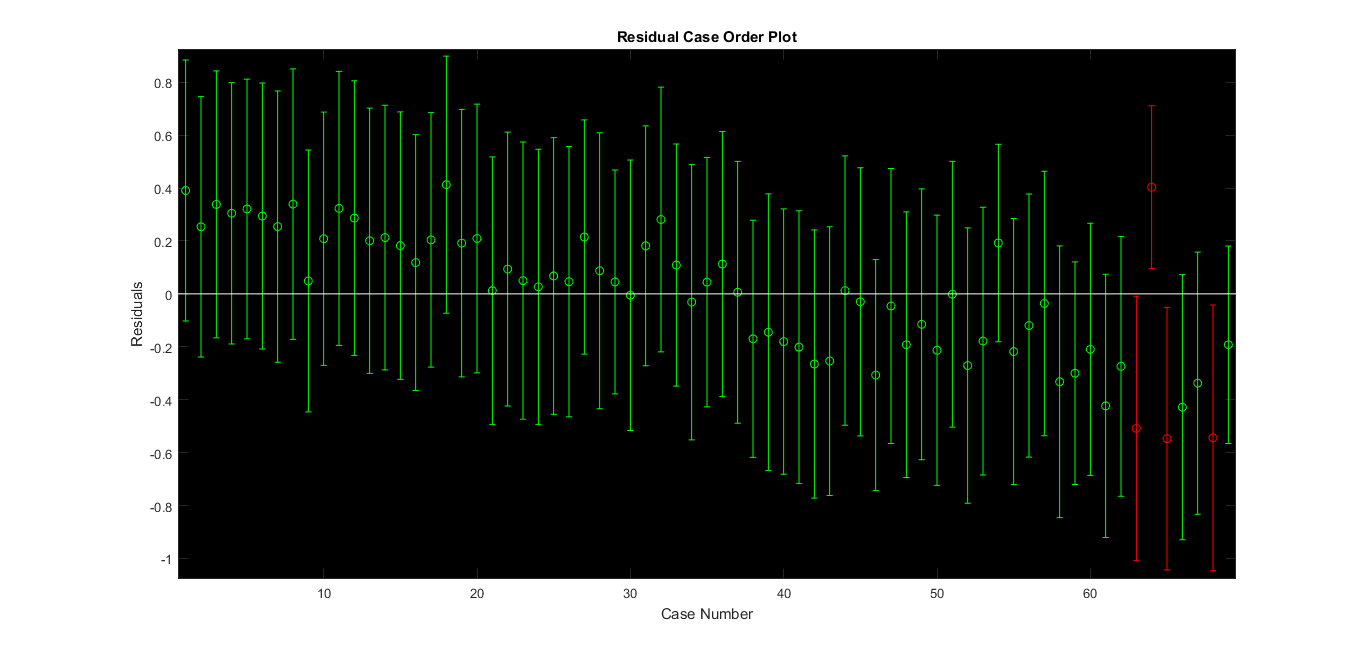


Figure 8: Residual Case Order Plot of Linear Regression

In light of the fact that the accuracy is relatively low, which is insufficient to reveal the features of each variable precisely, we consider taking the advantage of other methods. Principal Component Regression is applied later as an optimized method in part 5.1.

* 1. **KNN Algorithm**

In accordance with the given data, we try to use the data of which the online scores are matched to conduct the KNN algorithm to highly merge the vast amount of the data and find the shared features and characteristics of each sample to obtain the common properties of the roller coasters under similar condition to determine the relationship. [6]

We utilize Mahalanobis distance distinction to operate these data, which is processed after principal component analysis and features eradicating the dimension of each independent variables. The formula is as the following formula 17.



(17)

Among the formula, and denote two row vectors; denotes the covariance matrix; denotes the obtained Mahalanobis distance of the data.

For the accuracy, we correctly categorized 57 samples out of 69, achieving an accuracy of 83%. We made an optimization of this method in 5.2.

1. **Optimization**
   1. **Principal Component Regression**

Principal Component Regression suits explicitly for the problems that have a vast amount of independent data types, not all of which are tightly connected to the dependent data, which means some of the data are loosely related to the data. In view of considering that our problem has 9 independent variables, the method is highly compatible with our research.

We can still do as part 4.3, regarding the properties of roller coasters as dependent variables and the online scores as independent variables. We try to reduce the dimensionality, diminishing the vast amount of the original data and variables into fewer data and variables, while the new variables can retain the information in the original data by and large. [7]

We utilize the 9 original variables mentioned in 3.4 as the original data. We still use to denote independent variables matrixes and the dependent variables. The original variables are ; the new variables are . We use to denote the number of samples and use to denote the number of variables in each sample.

Applying Least squares regression, point estimation and interval estimation method which has previously been mentioned, we obtain the principal coefficient matrix as shown in table 9 with formula 18.

(18)

Table 9: Coefficient Matrix of principal component

|  |  |  |
| --- | --- | --- |
| Point Estimation | Interval Estimation | |
| -17.9038 | -28.228 | -7.5796 |
| 0.017885 | 0.002086 | 0.033684 |
| -0.01057 | -0.01941 | -0.00174 |
| 0.013596 | -0.00226 | 0.029449 |
| 0.078525 | 0.004955 | 0.152094 |
| -0.02736 | -0.06604 | 0.01132 |

The correlation coefficient of this method is 0.322210105118927. Although there is no discernable elevation in the coefficient, the method focuses more on the principal variables.

Ultimately, we conduct the inverse standardization process and obtain the equation interpreted in the original data, which is formula 19, and the final coefficient matrix, as shown in table 10.

(19)

Table 10: Final Coefficient Matrix of original variables

|  |  |  |
| --- | --- | --- |
| Point Estimation | Interval Estimation | |
| -17.9038 | -28.228 | -7.5796 |
| 0.075562 | 0.024277 | 0.126847 |
| 0.033707 | 0.016531 | 0.050884 |
| 0.020766 | 0.019247 | 0.022286 |
| 0.010701 | -0.05472 | 0.076122 |
| -0.00221 | 0.007277 | -0.0117 |
| 0.008843 | 0.01558 | 0.002106 |
| 0.000105 | -0.00306 | 0.003271 |
| -0.00103 | -0.01252 | 0.010466 |
| -0.00745 | -0.00913 | -0.00578 |

* 1. **Bayes Distinction**

Bayes Distinction ideally satisfies the requirements of such issue that each individual of the ensemble exists at different frequencies, which indicates that we need to take into consideration that the different possibilities that each individual exists. As for our research, each roller coaster is obviously impossible to appear at identical frequencies, so we apply Bayes Distinction to our study.

In the distance distinction method above, it does not take into account the frequency of each sample as a whole and does not take into account the loss caused by the wrong distinction. The Bayes distinction method modifies on the basis of distance distinction, and the formula is defined as in formula 20: [8]



(20)

Among which represents a posteriori probability; represents a prior probability; represents the frequency at which the sample appears; represents the total covariance matrixes. The distinction rule is that the posterior probability is the highest and the average wrong distinction loss is the lowest, which brings out the rule is as follows: If the condition meets the following formula 21:



(21)

Then we categorize into , among which is the ensemble, is the probability density function of , is prior probability of , which is the probability that it belongs a certain category when sample occurs, and is the number of . The solution formula for distinction analysis is as the following formulas 22-23:



(22)

(23)



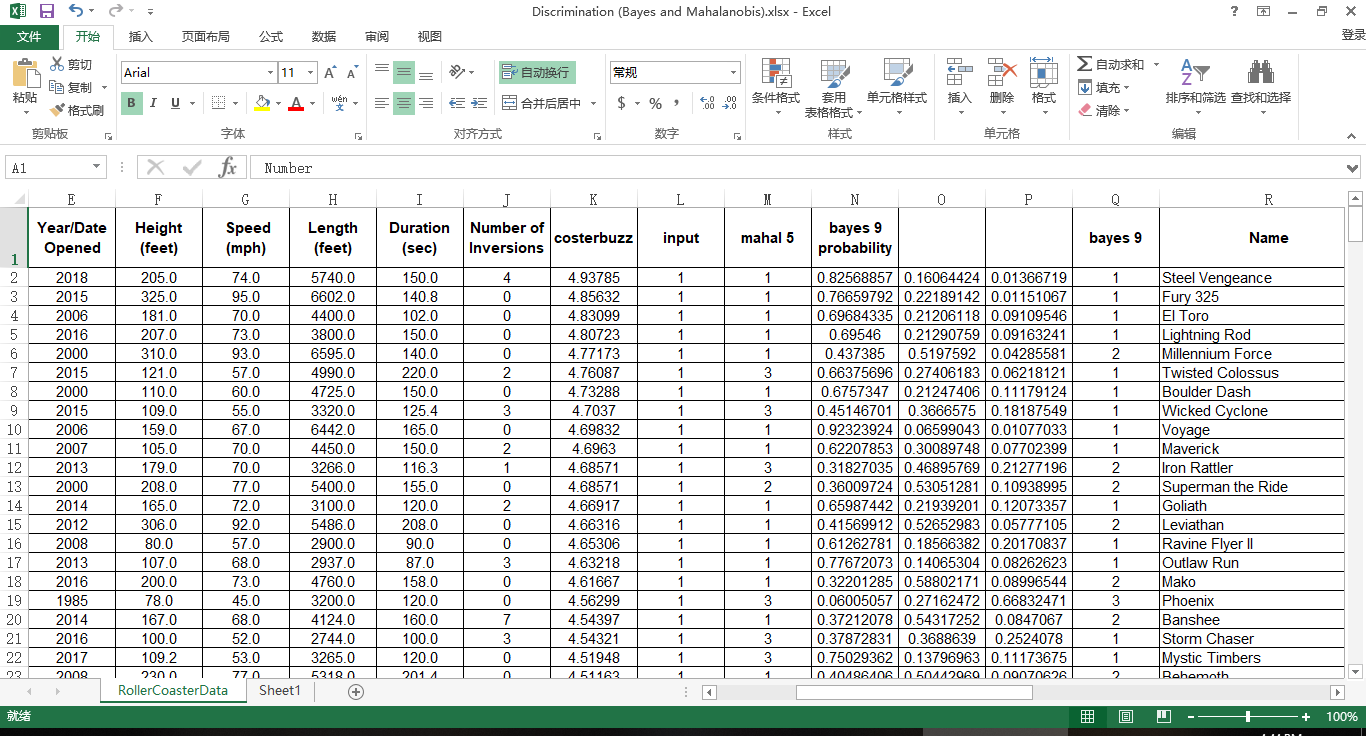
In this case, represents the conditional probability of wrongly categorizing the sample of to the ensemble . is the loss caused by this categorization. is a division of a set of distinction samples. is the average wrong distinction loss. The solution to a Bayes distinction analysis is to make the smallest set of solutions.

We divide the result of Bayes distinction into 5 categories, which are less than 4, 4 to 4.5, and 4.5 to 5. For the training set, if the online score lies in 4.5 to 5, we define the roller coaster as category 1. Likewise, we define the roller coaster of which the score is from 4 to 4.5 as category 2. We randomly pick out a certain amount of data from ALL the data which has no score online or the score is lower than 4 and define them as category 3. Using the MATLAB program, we still use all the data with the online score to carry out Bayes distinction solution.

The result is shown in the appendix, part of which is as following figure 8-9 and table 20. For instance, the number “36” shows that there are 36 samples with sit down type are judged as Category 1, which is the high score category.

For the accuracy, we correctly categorized 59 samples out of 69, achieving an accuracy of 85%, which is relatively higher than the accuracy obtained from KNN algorithm. The following table 11 is a part of the result.

Table 11: Bayes Distinction Result



We also make various charts and tables to exhibit our results, part of which are as the following figures 9-10 and table 12:

Figure 9: Bayes Result of Construction in Low Score

Figure 10: Bivariate plot from Height to Bayes Category

Table 12: Bayes Category to Type

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Sit down | Inverted | Stand up | Suspended | Flying | Wing |
| Category1 | 36 | 0 | 0 | 0 | 0 | 0 |
| Category2 | 59 | 17 | 2 | 2 | 5 | 3 |
| Category3 | 132 | 21 | 0 | 4 | 0 | 1 |

From the results given, we can clearly figure out the trend that the roller coasters which are in the place far away from North America tend to have a high score, especially the ones locate in Middle East, Oceana, and Russia. The roller coasters that are made from wood are more likely to have a higher score. A newly opened roller coasters are more welcomed. If the roller coaster is relatively higher, it is more possible to achieve a better score. 2 and a half minutes and 60 mph are a proper time for a loop and a satisfactory speed respectively. If the number of inversions is too high, it may conversely do harm to the passion of tourists to ride.

* 1. **BP Neural Network Fitting**

BP Neural Network is a kind of multilayer feed-forward network, which highly fits for the problem that there are data with a certain scale, the relationship between which is not too complicated to identify. When it comes to our target, we have a middle-sized database, and since the fitting process is not too intricate, the model can be applied to our goal.

We utilize BP neural network fitting as another method to promote the accuracy of the regression. BP neural network aims to encode itself with its high-dimensional features and to carry out dimension reduction processing towards high-dimensional data. It is marked by a feature extraction model with unsupervised learning, which can also combine a few basic features to obtain higher-layer abstract features. [9]

We utilize Tangent Sigmoid function as the transfer function; we use Levenberg Marquardt algorithm (trainlm) as the training algorithm; we use the Gradient descent with momentum weight and bias learning function (learngdm) as the learning algorithm; we use the mean square error (MSE) method as the learning function. The structure of the network and the performance plot are shown in figure 11 and 12.



Figure 11: BP Neural Network Structure. The layer number, which is 20, does not consumes too much time while the result is satisfactory.

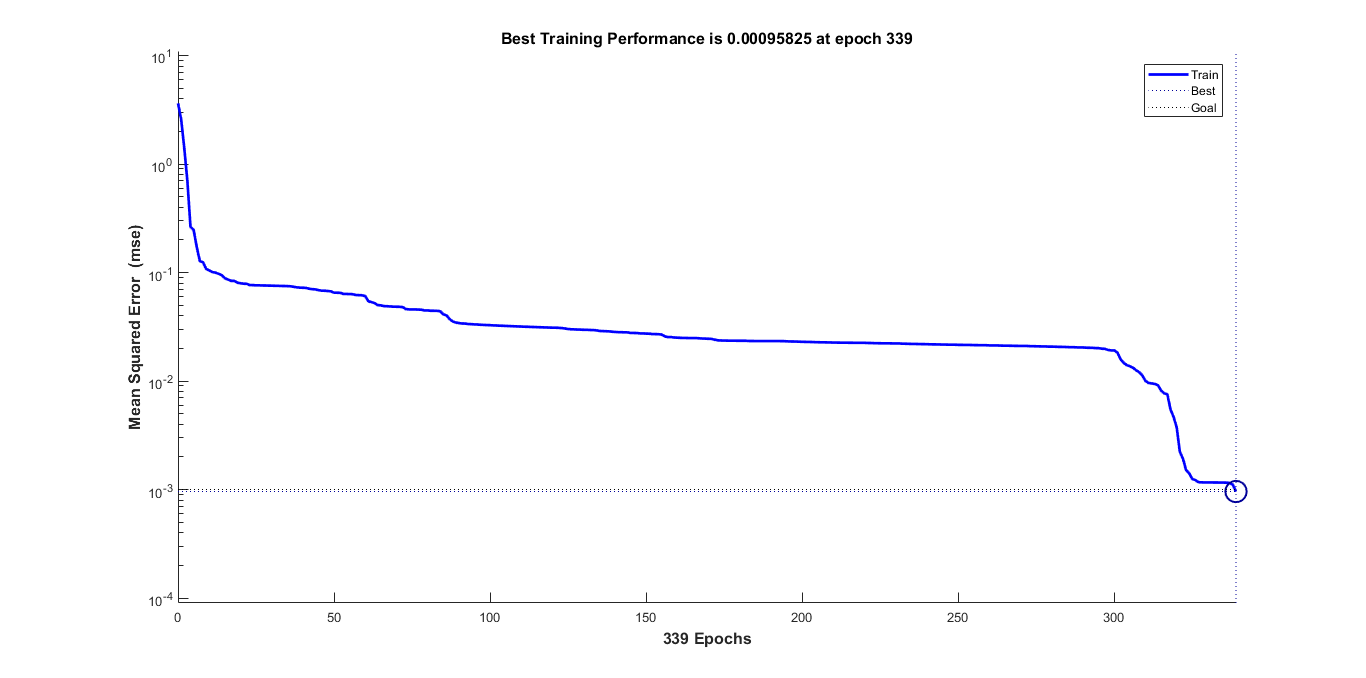
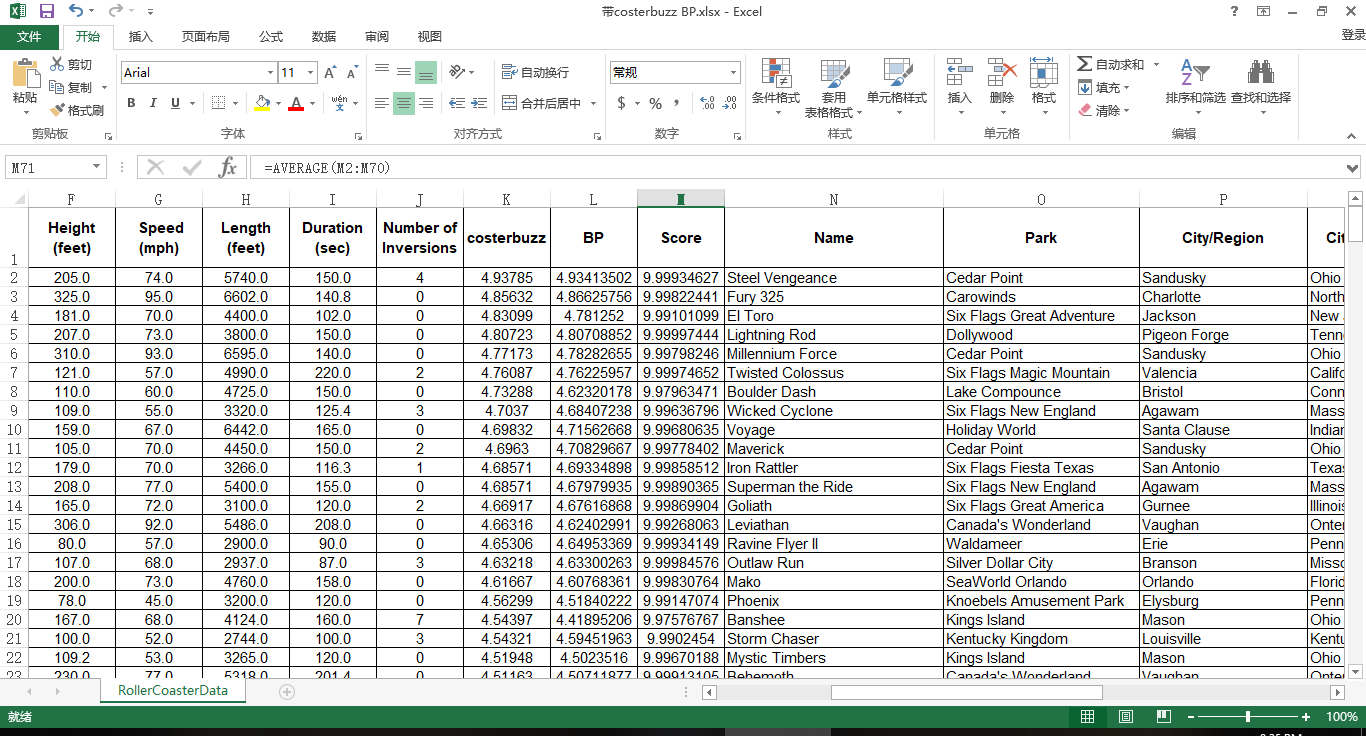


Figure 12: the performance plot of BP Neural Network. The training performance is improving rapidly.

Applying the MATLAB program, we use all the data with online score to carry out the BP neural network fitting.

We consider dividing the learning samples into three groups, each time using two of the groups to carry out a model and then test it with the test set. In light of the fact that there are mere 69 training data, it is not sufficient enough for us to conduct in this way. Hence, we use all the training data to training the BP Neural Network Algorithm. The result is in the appendix, part of which is as the following table 13.

Table 13: BP Neural Network Result. The error of some numbers is lower than 1%.



It can be seen that some of the predicted data run an accuracy that is higher than 99%.

* 1. **XG Boosting Algorithm**

We utilize XG Boosting algorithm to obtain the average value of each method of the samples. The basic formula is as the following formula 24

(24)

In the formula, denotes the overall loss of each classifier, denotes each classification function, and is a function of each classifier to reduce the loss. denotes the original result of Principal Component Analysis. denotes the result of Bayes distinction. denotes the original result of BP neural network fitting. For each category in Bayes distinction, we utilize the mid-value of each interval to numerate each category. We divide the result of Bayes distinction into 5 categories, which are less than 4, 4 to 4.5, and 4.5 to 5. Therefore, we use 3.75, 4.25, and 4.75 to denote the 3 result of the categories.

The main theory of BOOST algorithm is as follows. A complicated issue can be better estimated when synthesizing the judgment of each expert than that of a sole expert. For each step, we generate a model to accumulate each model to a whole model, which enables us to analyze the problems. Hence, we need to assemble several weak learners into a strong learner by determining the loss functions, (y\_i ) ̂, to minimalize the error and loss of misjudgment.

We input the predicted result of the three learners into the algorithm as the learning set and the real result as the target goal. We regard test set in the Bayes distinction and BP Neural Network as the testing set. With the help of XG Boosting module in PYTHON, we are able to determine the weight of the three learners to generate the final result. [10]

We utilize a formula to measure the error of our estimation, reaping an average score same as the original result and receiving almost a full score of 10, which shows that this model can successfully reflect the trend. The formula is as the following formula 25.

(25)

In the formula, denotes the score of the data, while and respectively denote the predicted value and the real value of the data.

We discover that many roller coasters have the same value of XG Boosting, which may due to the reason that there is too less training set while too much testing set. Since the BP Neural Network reaps a relatively accurate outcome among the 3 optimized models, we decide to use the result of the BP Neural Network Fitting as the final score and ranking if the outputs of XG Boosting are identical.

1. **Comparison of the top 10 Roller Coasters**

To give out the final ranking of our model, we decide to use the result of XG Boosting as the ranking criterion. As mentioned above, we find a flaw that some scores of roller coasters of XG Boosting are identical, so it is difficult for us to distinguish which roller coasters should be the top ones. In light of the high accuracy of the BP Neural Network Fitting given in 5.3, we decide to use the result of the BP Neural Network to rank the roller coasters if the results of XG Boosting are exactly the same. In other words, the first ranking criterion is the result of XG Boosting, and the second one is the result of BP Neural Network. The top 10 roller coasters and their scores are shown as following table 14.

Table 14: Final Ranking

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number** | **Name** | **Park** | **City/Region** | **City/State/Region** | **Country/**  **Region** | **Geographic Region** |
| 257 | T Express | Everland | Yongin-si | Gyeonggi-do | South Korea | Asia |
| 9 | Anaconda | Walygator Parc | Maizieres-les-Metz | Lorraine | France | Europe |
| 66 | Crazy Coaster | Loca Joy Holiday Theme Park | Yongchuan | Chongqing | China | Asia |
| 10 | Apocalypse | Six Flags America | Upper Marlboro | Maryland | United States | North America |
| 33 | Big Thunder Mountain | Disneyland Resort Paris | Marne la Vallee | Ile-de-France | France | Europe |
| 273 | Tonnerre de Zeus | Parc Asterix | Plailly | Picardie | France | Europe |
| 143 | Jupiter | Kijima Kogen | Beppu | Oita | Japan | Asia |
| 59 | Coaster Through the Clouds | Nanchang Wanda Theme Park | Xinjian | Nanchang, Jiangxi | China | Asia |
| 87 | Firehawk | Kings Island | Kings Mills | Ohio | United States | North America |
| **Number** | **Inversions (YES or NO)** | **Status** | **Construction** | **Type** | **Drop (feet)** | **Year/Date Opened** |
| 257 | NO | Operating | Wood | Sit Down | 150.9 | 2008 |
| 9 | NO | Operating | Wood | Sit Down | 40.0 | 1989 |
| 66 | YES | Operating | Steel | Sit Down |  | 2013 |
| 10 | YES | Operating | Steel | Stand Up | 90.0 | 2012 |
| 33 | NO | Operating | Steel | Sit Down | 39.3 | 1992 |
| 273 | NO | Operating | Wood | Sit Down |  | 1997 |
| 143 | NO | Operating | Wood | Sit Down |  | 1992 |
| 59 | NO | Operating | Steel | Sit Down | 255.9 | 2016 |
| 87 | YES | Operating | Steel | Flying |  | 2007 |
| **Number** | **Height (feet)** | **Speed (mph)** | **Length (feet)** | **Duration (min:sec)** | **Duration (sec)** | **Number of Inversions** |
| 257 | 183.8 | 64.6 | 5383.8 |  | 138.6 | 0 |
| 9 | 118.1 | 55.9 | 3937.0 | 2:10 | 130.0 | 0 |
| 66 | 108.3 | 52.8 | 2870.8 |  | 178.4 | 10 |
| 10 | 100.0 | 55.0 | 2900.0 | 2:00 | 120.0 | 2 |
| 33 | 72.2 | 40.4 | 4921.3 | 3:56 | 236.0 | 0 |
| 273 | 98.0 | 52.0 | 4044.0 | 2:05 | 125.0 | 0 |
| 143 | 138.0 | 57.0 | 5249.3 | 2:34 | 154.0 | 0 |
| 59 | 242.8 | 84.5 | 5105.0 | 4:12 | 252.0 | 0 |
| 87 | 115.0 | 50.0 | 3340.0 | 2:10 | 130.0 | 5 |
| **Number** | **BP** | **XGBoosting** | **G Force** | **Vertical Angle (degrees)** |  |  |
| 257 | 6.706793 | 4.68571 |  | 77 |  |  |
| 9 | 6.004326 | 4.68571 |  |  |  |  |
| 66 | 5.979574 | 4.68571 |  |  |  |  |
| 10 | 5.926418 | 4.68571 |  |  |  |  |
| 33 | 5.87154 | 4.68571 |  |  |  |  |
| 273 | 5.777834 | 4.68571 |  |  |  |  |
| 143 | 5.754164 | 4.68571 |  | 45 |  |  |
| 59 | 5.726858 | 4.68571 |  |  |  |  |
| 87 | 5.54524 | 4.68571 | 4.3 |  |  |  |

Apart from the score online we use in the modeling part, we find a second scoring website from the website Coaster Critic [11], of which the score can be referred to the appendix. Comparing the two top 10 roller coasters, the most significant difference we find is that our scores focus on not only the roller coasters in the US but also the roller coasters on a world scale. The location of our top 10 roller coasters includes the US, France, Korea, China, and Japan, which demonstrates that we truly achieve the goal that selecting the roller coasters based on objective and quantitative data rather than personal, subjective opinion. We are able to recommend the roller coasters only from the properties of themselves rather than personal opinions.

There is also some consistency between the two scores. For instance, wooden roller coasters are both highly rated, of which the reason may lie at people prefer the obsolescence of conventional wood roller coasters. The top 10 roller coasters of both are less likely to have inversions comparing with the roller coasters ranked after 10. The opening years both cover a wide range, from the 1980s to two years ago. The roller coasters with average speed, length, duration, or height are both leading the top of the rank.

We also compare our result with other websites, such as the result from MostLuxuriousList [12] or TheTopTens® [13], the result of which can be seen in the appendix (some roller coasters in the two websites are missing in the given database of roller coasters). The results of the comparison are also similar to those of the previous website. The top roller coasters online concentrate in the US, while the top coasters our model gives out involves a broader range. The comparison with the ranking from MostLuxuriousList and TheTopTens® also reveal that the parameters of the top coasters lie in an average interval, which manifests that middle-interval coasters are more warmly welcomed.

1. **Concept and design for a user-friendly app**

The user-friendly app we construct mainly aims to satisfy the riders' needs on roller coaster riding selection and meet individual demands. The app mainly contains 3 aspects--- recommendation of roller coasters based on all the applied riders' experience on a global scale, the specific recommendation of roller coasters to individuals after the data processing and the analysis on the individual's past preference, and the selection of roller coasters by the filter to meet the users' needs. The roller coasters' own prosperities form the primary database for the selection and recommendation, and the algorithm will help with the analyzing process.

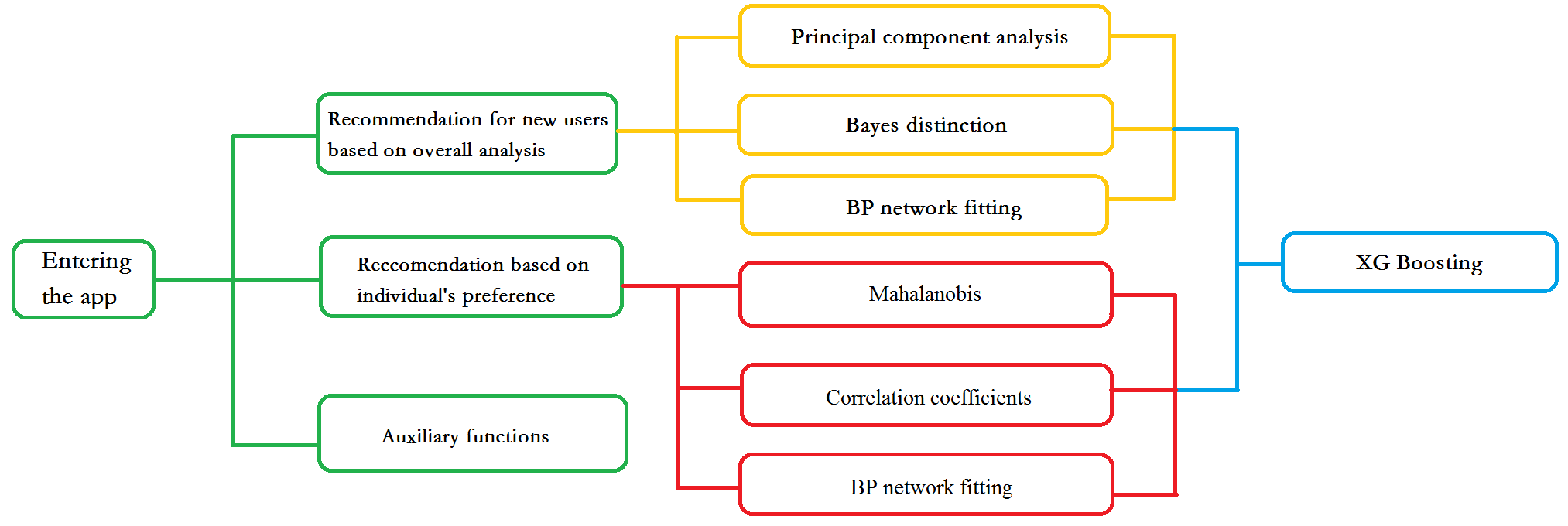
Figure 13 shows a flowchart of our desired application, and figure 14 shows the effect of the app. 

Figure 13: Flow chart of our desired application. We set various functions and models to realize the functions.



Figure 13: Desired Panel of our application. Our panel is attracting!

* 1. **Initial Recommendation**

First, the app will ask for the individuals' personal information including the region they live in. Then it can first select the roller coasters from that region and make corresponding recommends. The registered riders will be required to rate the roller coasters they have rid after the thrilling experience, and each piece of information they record will be put in the database for the analysis. In order to encourage the registered riders to make a contribution to the database, some rewards may be provided. The question may involve the following aspects: the feeling after the ride, the degree of excitement and stimulation based on the individuals' experience, the rating of roller coasters as a whole, so and so forth. All of these questions are the users' subjective inputs, based on the rating they have given, and we can use them to refresh the ranking of the roller coasters at every moment. In this way, the personal information can be turned as the input of quantitative analysis, improving the original model's accuracy and stability. At the same time, this ranking would be used to provide the new users with the top roller coasters and encourage them to experience the best ride.

In order to achieve the goal, we can do as what we have done in the previous parts, using the Principal Component Analysis, Bayes Distinction, and BP Neural Network to obtain the score of each roller coaster and use XG Boosting algorithm to synthesize the result of the three methods to achieve the best accuracy of the prediction. Since the results are based on common preference, it will never prove to be fallible by a typical user and tend to recommend the roller coasters that most users want to ride. The method suit well for the problem because when the scored data increases as it continues to be collected from the users, the model will be progressively accurate and achieve a more precise recommendation.

* 1. **Recommendation Base on Preference**

The information continuously provided by one individual---the track record---can also provide useful information on the individual's own preference. We have two functions, behind which the basic algorithm tries to determine the correlation of the riders' record and the roller coasters in the database, and the correlation of the riders’ identity and the ones in the database. We can recommend the roller coasters that manifest a more substantial correlation with the roller coasters that have already been rid, or recommend the roller coasters that the users which have a more significant correlation with the users have rid, eliminating the roller coasters that have been rid by the users.

To define the similarity between the historical data of the users and the data in the database, we can set each data of the user or the data in the database as a row vector and calculate the correlation coefficients between the two. Then we can rank the roller coasters by the correlation coefficients from the largest one to the smallest one, recommending the ones with several largest correlation coefficients; additionally, we can rank the users who show high correlation and recommend the coasters which the similar users have rid. We can also calculate the Mahalanobis distance between the row vectors previously mentioned and rank the roller coasters as above, taking the advantage that the method does not take the dimension of the data into account. Based on the algorithm, the app can thus successfully achieve its second crucial function, and make the recommendation based on the quantitative analysis, spotting the users' need and saving the users' time for searching.

Besides, we can use Neural Network to achieve a personal and private recommendation. We can gather the information of the users, such as gender, region, so and so forth, treating them as independent variables as well as the properties of the roller coasters. In this way, the recommendation of the program can not only take the information of the roller coasters into consideration but also take the properties of the users into account, which gives rise to the exact match between the users and the roller coasters.

* 1. **Search Engine for Roller Coasters**

The app could also set up a selecting system to meet the riders' special needs. The system will be much like a search engine, but it will be entirely based on the property of the roller coasters. To make the sifting process more user-friendly, the options for the potential riders to choose will not include specific numbers. For instance, if the potential riders want to select a roller coaster with longer duration time, the search engine will not require them to put in specific numbers, but only choose from different levels such as short(30-60sec), medium(60-120sec), and long(>120sec). Different selecting options will thus minimize the number of roller coasters based on the rider's demand and correspondingly make the proper recommendation.

To algorithmically achieve this goal, we can treat the word typed by the users as a string and find the strings in the database of which the substrings include the string which the users type and print the name of the corresponding roller coasters.

* 1. **Auxiliary functions**

Besides from the main purposes, auxiliary functions may also be included. First, a community will be set up to let the riders share their own riding experience, which may boost their sense of belonging with others who also like roller-coaster riding. They may even find the app useful as it can allow them to make friends with those who share the same interest with them. Besides, basic information of the roller coaster sites around the globe will be provided, in the form of both pictures and videos to give the potential riders a real sense of spectacularity, and every rider is welcomed to write their own experience and comments. For those especially love the thrilling feeling, they can also keep a journal in this app, and write down anything they want to recall about every one of their stimulating experience. Furthermore, up-to-date news about the roller coasters around the world will be timely reported by converging the information online, capturing the riders interest and promote them to have a try. Some related commercial products like keychains and postcards could also be provided after the cooperation with certain entertainment companies.

In brief, the app we construct will use the algorithm and quantitative analysis to meet the potential riders' needs and help them decide the best option, guaranteeing them a satisfying and enjoyable experience.

1. **Analysis**
   1. **Sensitivity Analysis**

Sensitivity analysis is a method of studying and analyzing the sensitivity of the model to changes in system parameters or surrounding conditions. In the optimization methods of our team, it can detect the stability of our model, especially when the given data is not accurate.

In this part, we will mainly discuss the sensitivity of the application part. If we give the test set of the data an increase or a decrease of 1%, by changing the value of the original data matrix on the program, we discover that the output data of the principal component regression changes precisely 1%; almost all the results in the Bayes Distinction part have no difference in categories; the majority of the output of BP neural network model fluctuates 1% approximately. The output after the change is small enough for us to make a further adjustment. Therefore, it is acceptable in the modeling. This sensitivity analysis also indicates that our model has universality and can be applied to more situations. For instance, if there is some error in the data, out final result does not vary rapidly correspondingly. Therefore, our model is relatively stable. The data of Sensitivity Analysis can be referred to the appendix.

* 1. **Strength and Weakness**

The strength of the model for the rating of roller coasters and the algorithm being used mainly include the following aspects:

The methods applied in this model includes both qualitative and quantitative analysis, and different conclusions from various methods can be obtained through our modeling process. As for the qualitative analysis, for instance, from the results of Bayes Distinction, we can figure out the trend that the roller coasters which are in the place far away from North America tend to have a high score and that the roller coasters made from wood are more likely to have a higher score. A newly opened roller coasters are more welcomed. If the roller coaster is relatively higher, it is more feasible to achieve a better score. 2 and a half minutes and 60 mph are a proper time for a loop and a satisfactory speed respectively. All of these are obtained through the qualitative analysis which provides us with valuable information. As for the quantitative analysis, it is evident that nearly all the algorithms being applied need the input of the database and the whole rating process our model depends on needs the analysis and testing of quantitative properties of every roller coasters. Far from the common evaluating methods based on the riders' subjective inputs currently, our model is based on facts, analysis, and training, which will undoubtedly provide more accurate and scientific results on the ranking.

Our methods also take into consideration both continuous and discrete variables, the fact of which makes our model especially suitable for the question's requirement. For instance, in our optimized model, the data of discrete independent variables, like the material used for the construction of the roller coasters, are mainly analyzed by Principal Component Analysis, which is suitable for the processing of discrete data. On the other hand, Bayes Distinction mainly served to analyze the continuous variables including average speed, maximum height, so and so forth. For the XG Boosting algorithm, it is suitable for all independent variables and can successfully synthesize the result of the optimized model. Thus, our model has sufficiently taken into account the property of different types of data, and the result yielded will, in turn, be highly persuasive.

Another outstanding point of the whole modeling process is the variety of methods being used---from the basic statistics and linear regression to the optimized model of BP Network Fitting and XG Boosting algorithm. The final ranking is produced through exploration on various methods, and we have been continuously evaluating the viability of different models and thinking about how to improve our results through more advanced methods further. It is our endeavor for excel that guarantees a more accurate and suitable method as a whole for the quantitative analysis of the rating. The multiplicity of methods not only shows our clear logic chain from the perspective of pragmatic problem solving but also ensures a more stable and precise result.

The model is also propagable as a standard method for the rating of any new roller coaster given. In other words, besides the 300 roller coasters provided in the table, if any new roller coaster with basic properties given are added to the database of our whole modeling process, a rating can also be produced and be added to the original ranking.

Furthermore, our model is highly flexible and can be applied to various situations. For instance, if another new property, no matter continuous or discrete, is added to all the roller coasters, the method and logic behind of our modeling can also be applied, since the ranking is based on a whole series of analysis instead of randomly assigning weight to each property. Thus, a different ranking is likely to be produced. Besides, it is also worth noticing that if more roller coasters with ratings are added to the training set in the XG Boosting algorithm, a more accurate result will be produced. This enhancement in accuracy is because specific optimized methods, like XG Boosting algorithm, yield results based on the self-learning of data input, which indicates more data analysis and more source for learning will boost the accuracy and stability of the method. Therefore, the model we proposed can constantly learn from the additional data input, so it is highly flexible and makes possible the dynamic adjustments, which makes the model suitable for being applied under various circumstances.

Finally, our model yields the precise rating results instead of just the ranking of the top 10. Actually, it is also possible to produce the quantitative result of any roller coaster given. This result will definitely be better than the vague ranking result which is a much weaker conclusion compared. The accurate rating can reflect the difference in a quantitative way between each roller coasters and give the potential riders more compatible information.

The weakness of our model mainly includes the aspects following:

The data of the roller coasters given is sometimes not ample and sufficient enough for the evaluation of one particular independent variable. This sufficiency will, in turn, cause the data input being inaccurate since some information is missing and cannot be applied to the analysis. Though interpolation is done during the data cleaning process, there are still some data left vacancy because of the insufficiency of existed data, for the loss of a large quantity of data will make the interpolation process meaningless. So the loss of data will affect our final rating, even though the methods we applied have considered maintaining the original information as much as possible.

Besides, there are only 300 pieces of information given for the learning and testing of data, the fact of which will unavoidably make the method like XG Boosting not accurate enough. For methods like this, more pieces of data being learned will further boost the accuracy. However, despite some deficiency, it is still a suitable method for the synthesis of the optimized result and provide us with the relatively more stable and precise rating result.

1. **Reference**

[1] https://wenku.baidu.com/view/c6e8f183b9d528ea81c779af.html, Clustering Analysis with Matlab.

[2] https://coasterbuzz.com/RollerCoasters/Top100, Top 100 Roller Coasters: The CoasterBuzz 100.

[3] <http://blog.csdn.net/MATLAB_matlab/article/details/59483185?locationNum=10&fps=1>, MATLAB principal component analysis.

[4] Mingbei C., Gang H., Guoufu Z. Comprehensive evaluation of takeaway website based on AHP method——Eleme website as an example [J]. Modern Business, 2015, 12: 57-58.

[5] <https://wenku.baidu.com/view/99c8408e6529647d272852cd.html>, Interval estimation and linear regression analysis with MATLAB.

[6] Jiang W. Comparative Study of Fisher Discriminant and Mahalanobis Distance Discriminant [J]. Journal of Ningbo Polytechnic, 2017, 21(5); 91-94.

[7] Yimeng F. Analysis of influencing factors of customer purchase behavior in E-commerce [J], Industrial & Science Tribune, 2014, 13(8): 138-139.

[8] Haiwei W. Yu X., Yalin W.A bivariate hierarchical Bayesian approach to predicting customer purchase behavior [J], Journal of Harbin Engineering University, 2007, 28(8): 949-954.

[9] Wu P. Application of Cigarette Sales Forecasting Based on Neural Network [J], Computer Simulation, 2012, 29(3): 227-230.

[10] https://blog.csdn.net/weixin\_42029738/article/details/81675234, Hand-in writing XG Boost programs.

[11] http://www.coastercritic.com/roller-coaster-reviews/, Coaster Reviews List – CoasterCritic.

[12] https://www.mostluxuriouslist.com/top-10-best-roller-coasters-in-the-world/, Best Roller Coasters in the World - List of Top Ten Ranking.

[13] https://www.thetoptens.com/best-roller-coasters/, Top Ten Best Roller Coasters - TheTopTens®.

1. **Appendix**
   1. **MATLAB Code**

[m,n]=size(X);

x=X(:,1);

y=X(:,2);

temp=x(1,1);

count=0;

sum=0;

row=0;

for i=1:m

if x(i)==temp;

sum=sum+y(i);

count=count+1;

else

row=row+1;

Y(row,1)=temp;

Y(row,2)=sum/count;

count=0;

sum=y(i);

count=1;

temp=x(i);

end

end

Y(row+1,1)=temp;

Y(row+1,2)=sum/count;

x=Y(:,1);

y=Y(:,2);

y1 = interp1(x,y,a,'pchip') ;

d=pdist(A,'Mahal');

z= linkage(d);

H=dendrogram(z,293)

T=cluster(z,30);

stdr=std(x);

[n,m]=size(x);

sddata=x./stdr(ones(n,1),:);

[p,princ,egenvalue]=princomp(sddata);

per=100\*egenvalue/sum(egenvalue);

[m,n]=size(a);

for i=1:m

B{i}=a(i,:);

end

for i=1:m

C{i}=zeros(n,n);

for j=1:n

for k=1:n

C{i}(j,k)=B{i}(j)/B{i}(k);

end

end

end

egenvector=[];

for i=1:m

t=C{i};

[x,lumda]=eig(t);

r=abs(sum(lumda));

n=find(r==max(r));

max\_lumda\_A(1,i)=lumda(n,n);

max\_x\_A{i}=x(:,n); %ÌØÕ÷Öµ

max\_x\_A{i}=max\_x\_A{i}./sum(max\_x\_A{i});

egenvector=[egenvector max\_x\_A{i}];

end

egenvector=egenvector';

%yangbenµÚÒ»ÁÐÊÇ·ÖÀàÇÃ½øÈ¥

%bÊÇ´ýÅÐµÄÇÃ½øÈ¥£¬gÇÃ½øÈ¥

%iiiÊÇ¸ÅÂÊ£¬½á¹û

%HÊÇºóÑé¸ÅÂÊ£¬½á¹û

%g-group·ÖÀàÊý£¬ºóÀ´Ð´ÁË¸ö×Ô¶¯¼ì²â·ÖÀàÊýµÄ£¬²»¹ýÃ»ÔÚmatlabÏÂÐ©£¬ºÇºÇ

[m,n]=size(yangben);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:g

groupNum(i)=0;

group(i)=0;

for j=1:m

if yangben(j,1)==i

group(i)=group(i)+1;

end

end

if i==1

groupNum(i)=group(i);

else

groupNum(i)=groupNum(i-1)+group(i);

end

end

group;

groupNum; %¼ÆËã·ÖÀà¸öÊýÊý×é

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%¼ÆËã×ÜÆ½¾ùÖµ

% for j=1:n-1

% TotalMean(j)=0;

% for i=1:m

% TotalMean(j)=TotalMean(j)+yangben(i,j+1);

% end

% TotalMean(j)=TotalMean(j)/m;

% end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GroupMean=[];

for i=1:g

if i==1

low=1;

up=groupNum(i);

else

low=groupNum(i-1)+1;

up=groupNum(i);

end

matrix=yangben(low:up,:);

MatrixMean=mean(matrix); %¸÷·ÖÀà×éÆ½¾ùÖµ

GroupMean=[GroupMean;MatrixMean];

for u=low:up

for v=2:n

C(u,v-1)=yangben(u,v)-MatrixMean(v);

end

end

end

C;

GroupMean;

V=C'\*C/(m-g);

V\_inv=inv(V); %¶Ô¾ØÕóVÇóÄæ

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GroupMean=GroupMean(:,2:n);

Q1=GroupMean\*V\_inv;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:g

lnqi(i)=log(group(i)/m);

mat=GroupMean(i,:);

Q2(i)=lnqi(i)-0.5\*mat\*V\_inv\*mat';

end

lnqi;

Q2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[u,v]=size(b);

result=[];

for i=1:u

x=b(i,:);

yy=Q1\*x'+Q2';

result=[result yy];

end

res=result'; %¼ÆËãµÄ´ýÅÐÊý¾Ý¶Ô¸÷±ê×¼Êý¾ÝµÄÏßÐÔ¼ÆËãÖµ

[rows,cols]=size(result);

for i=1:cols

iljj=0;

mlljj=result(:,i);

for j=1:rows

iljj=iljj+exp(result(j,i)-max(mlljj));

end

for j=1:rows

houyangailv(j,i)=exp(result(j,i)-max(mlljj))/iljj;

end

end

H=houyangailv'; %ºóÑé¸ÅÂÊ

iii=[];

for a=1:u

k=max(H(a,:));

for ii=1:g

if k==H(a,ii)

iii=[iii;ii];

end

end

end

clear c catagory detection i j k m n

for i=1:7

c{i}=[];

end

for i=1:293

catagory=b(i,1);

for j =1:7

[m,n]=size(c{j});

if n~=0

detection=0;

for k =1:n

if c{j}(5,k)==a(i,j)

c{j}(catagory,k)=c{j}(catagory,k)+1;

detection=1;

end

end

if detection==0

c{j}(5,(n+1))=a(i,j);

c{j}(catagory,(n+1))=(c{j}(catagory,(n+1)))+1;

end

end

if n==0

c{j}(5,1)=a(i,j);

c{j}(catagory,1)=(c{j}(catagory,1))+1;

end

end

end

for i=1:7

c{i}=c{i}';

end

%??????

%p=[-1 -1 3 1;-1 1 5 -3];

%t=[-1 -1 1 1];

%??????BP??

net=newff(minmax(p),[20 1],{'tansig','purelin'},'trainlm');

%??????

net.trainParam.epochs=10000;

net.trainParam.goal=0.001;

net.trainParam.show=50;

net.trainParam.lr=0.05;

net.trainParam.mc=0.9;%????????0.9

net=train(net,p,t); % ????

A=sim(net,traini); %????

* 1. **PYTHON Code**

import xlrd

import xlwt

ExcelFile=xlrd.open\_workbook(r'C:\Users\tianzhy\Desktop\COMAP\_RollerCoasterData\_2018 - Copy.xlsx')

sheet=ExcelFile.sheet\_by\_name('RollerCoasterData')

workbook = xlwt.Workbook(encoding = 'ascii')

worksheet = workbook.add\_sheet('My Worksheet')

for i in range (1,219):

temp = sheet.cell(i,17).value

#temp = str.split(temp,":")

timee=round(float(temp)\*1440)

worksheet.write(i, 0, label = str(timee))

workbook.save('Excel\_Workbook.xls')

import pandas as pd

import xgboost as xgb

from sklearn import preprocessing

import numpy as np

train = pd.read\_csv(r'D:\XGBoost\_learn\click rate\train1.csv', header=0)

tests = pd.read\_csv(r'D:\XGBoost\_learn\click rate\test\_pre.csv', header=0)

# trains=train.iloc[:, 1:].values

# labels=train.iloc[:,:1].values

# test = tests.iloc[:, :].values

'''

train['time\_stamp'] = pd.to\_datetime(pd.Series(train['time\_stamp']))

tests['time\_stamp'] = pd.to\_datetime(pd.Series(tests['time\_stamp']))

train['Year'] = train['time\_stamp'].apply(lambda x: x.year)#Year

train['Month'] = train['time\_stamp'].apply(lambda x: x.month)#Month

train['weekday'] = train['time\_stamp'].dt.dayofweek#weekday

train['time'] = train['time\_stamp'].dt.time#time

tests['Year'] = tests['time\_stamp'].apply(lambda x: x.year)#Year

tests['Month'] = tests['time\_stamp'].apply(lambda x: x.month)#Month

tests['weekday'] = tests['time\_stamp'].dt.dayofweek#weekday

tests['time'] = tests['time\_stamp'].dt.time#time

train = train.drop('time\_stamp', axis=1)

train = train.dropna(axis=0)

tests = tests.drop('time\_stamp', axis=1)

tests = tests.fillna(method='pad')

'''

for f in train.columns:

if train[f].dtype=='object':

if f != 'shop\_id':

print(f)

lbl = preprocessing.LabelEncoder()

lbl.fit(list(train[f].values))

train[f] = lbl.transform(list(train[f].values))

for f in tests.columns:

if tests[f].dtype == 'object':

print(f)

lbl = preprocessing.LabelEncoder()

lbl.fit(list(tests[f].values))

tests[f] = lbl.transform(list(tests[f].values))

print("test")

print(tests.info())

# for f in train.columns:

# if f !='':

# train[f] = train[f].astype(float)

print(train.info())

# train = train.astype(float)

# tests = tests.astype(float)

trains = train.iloc[:, 1:].values

labels = train.iloc[:, :1].values

test = tests.iloc[:, 1:].values

feature\_columns\_to\_use = ['wifi\_strong1','wifi\_strong2','wifi\_strong3']

big\_X = train[feature\_columns\_to\_use].append(tests[feature\_columns\_to\_use])

train\_X = big\_X[0:train.shape[0]].as\_matrix()

test\_X = big\_X[train.shape[0]::].as\_matrix()

train\_y = train['shop\_id']

gbm = xgb.XGBClassifier(silent=1, max\_depth=10, n\_estimators=1000, learning\_rate=0.05)

gbm.fit(train\_X, train\_y)

predictions = gbm.predict(test\_X)

submission = pd.DataFrame({'row\_id': tests['row\_id'],

'shop\_id': predictions})

print(submission)

submission.to\_csv("submission.csv", index=False)

'''

print(trains)

parameters={

'silent':1,

'max\_depth': 3,

'n\_estimators':300,

'learning\_rate':0.005,

}

feature\_types={

'float','str','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float',

}

# feature\_types = {

# 'str', 'float',

# }

ft=list(feature\_types)

ParamLst = dict(parameters.items())

offset = 15

num\_rounds = 1

xgtest = xgb.DMatrix(tests)

print("//////////////////////////")

xgtrain = xgb.DMatrix(trains[:offset,:], label=labels[:offset])

print("//////////////////////////")

xgval = xgb.DMatrix(trains[offset:, :], label=labels[offset:])

watchlist = [(xgtrain, 'train'), (xgval, 'val')]

# training model

model = xgb.train(ParamLst, xgtrain, num\_rounds, watchlist, early\_stopping\_rounds=100)

# model.save\_model('./model/xgb.model') # 用于存储训练出的模型

preds = model.predict(xgtest, ntree\_limit=model.best\_iteration)

np.savetxt('submission\_xgb\_MultiSoftmax.csv', np.c\_[range(1, len(test)+1), preds],

delimiter=',', header='ImageId,Label', comments='', fmt='%d')

'''

* 1. **Final Result**

|  |  |  |  |
| --- | --- | --- | --- |
| **BP** | **XGBoosting** | **Name** | **Park** |
| 6.70679312 | 4.68571 | T Express | Everland |
| 6.00432594 | 4.68571 | Anaconda | Walygator Parc |
| 5.9795744 | 4.68571 | Crazy Coaster | Loca Joy Holiday Theme Park |
| 5.92641799 | 4.68571 | Apocalypse | Six Flags America |
| 5.87153982 | 4.68571 | Big Thunder Mountain | Disneyland Resort Paris |
| 5.77783393 | 4.68571 | Tonnerre de Zeus | Parc Asterix |
| 5.75416379 | 4.68571 | Jupiter | Kijima Kogen |
| 5.7268579 | 4.68571 | Coaster Through the Clouds | Nanchang Wanda Theme Park |
| 5.54524029 | 4.68571 | Firehawk | Kings Island |
| 5.48925231 | 4.68571 | Silver Star | Europa Park |
| 5.48420322 | 4.68571 | Road Runner Express | Six Flags Fiesta Texas |
| 5.4459491 | 4.68571 | Hyper Coaster | Land of Legends Theme Park |
| 5.43000649 | 4.68571 | Corkscrew | Valleyfair! |
| 5.39706717 | 4.68571 | Gao | Greenland |
| 5.38403092 | 4.68571 | Nessie Superrollercoaster | Hansa Park |
| 5.3701208 | 4.68571 | Do-Dodonpa | Fuji-Q Highland |
| 5.30667194 | 4.68571 | Shambhala | PortAventura Park |
| 5.29401189 | 4.68571 | Wildfire | Kolmarden |
| 5.27847101 | 4.68571 | Velikolukskiy Myasokombinat-2 | Wonder Island |
| 5.27753705 | 4.68571 | Altair | Cinecittà World |
| 5.26020897 | 4.68571 | Python in Bamboo Forest | Nanchang Wanda Theme Park |
| 5.25935882 | 4.68571 | Jungle Trailblazer | Fantawild Dreamland |
| 5.25552236 | 4.68571 | Bandit | Movie Park Germany |
| 5.22338425 | 4.68571 | Coaster Express | Parque Warner Madrid |
| 5.22169978 | 4.68571 | Formula Rossa | Ferrari World Abu Dhabi |
| 5.2118465 | 4.68571 | Bat | Kings Island |
| 5.20377624 | 4.68571 | Saw - The Ride | Thorpe Park |
| 5.10973073 | 4.68571 | Hyperion | Energylandia |
| 5.09837333 | 4.68571 | Superman el Último Escape | Six Flags Mexico |
| 5.09435065 | 4.68571 | Fujiyama | Fuji-Q Highland |
| 5.04751478 | 4.68571 | Batwing | Six Flags America |
| 5.0305753 | 4.68571 | Goliath | Six Flags Fiesta Texas |
| 5.00009406 | 4.68571 | Schwur des Kärnan | Hansa Park |
| 4.99280655 | 4.68571 | Black Mamba | Phantasialand |
| 4.98454445 | 4.68571 | Kong | Six Flags Discovery Kingdom |
| 4.98018076 | 4.68571 | Balder | Liseberg |
| 4.94734848 | 4.68571 | Batman The Ride | Six Flags Over Texas |
| 4.93413502 | 4.68571 | Steel Vengeance | Cedar Point |
| 4.92042702 | 4.68571 | Star Mountain | Beto Carrero World |
| 4.9100592 | 4.68571 | Mind Eraser | Elitch Gardens |
| 4.9100592 | 4.68571 | Riddler Revenge | Six Flags New England |
| 4.90436075 | 4.68571 | Batman The Ride | Six Flags Great America |
| 4.89876472 | 4.68571 | Batman The Ride | Six Flags St. Louis |
| 4.89428793 | 4.68571 | Batman The Ride | Six Flags Magic Mountain |
| 4.88834528 | 4.68571 | Flight of the Phoenix | Harborland |
| 4.8863265 | 4.68571 | Mind Eraser | Six Flags America |
| 4.87972149 | 4.68571 | Desert Race | Heide-Park Soltau |
| 4.86625756 | 4.68571 | Fury 325 | Carowinds |
| 4.86352935 | 4.68571 | Ultimate | Lightwater Valley |
| 4.82884134 | 4.68571 | Eurosat Can Can Coaster | Europa Park |
| 4.81417083 | 4.68571 | Incredible Hulk | Universal Studios Islands of Adventure |
| 4.80708852 | 4.68571 | Lightning Rod | Dollywood |
| 4.79831948 | 4.68571 | Flight Deck | California's Great America |
| 4.79566951 | 4.68571 | Viper | Six Flags Great America |
| 4.79116216 | 4.68571 | Flight of Fear | Kings Island |
| 4.78282655 | 4.68571 | Millennium Force | Cedar Point |
| 4.781252 | 4.68571 | El Toro | Six Flags Great Adventure |
| 4.77747615 | 4.68571 | Incredicoaster | Disney California Adventure Park |
| 4.76225957 | 4.68571 | Twisted Colossus | Six Flags Magic Mountain |
| 4.75340133 | 4.68571 | Big Thunder Mountain Railroad | Disneyland |
| 4.74740279 | 4.68571 | Desperado | Buffalo Bill's Resort & Casino |
| 4.74407976 | 4.68571 | Demon | California's Great America |
| 4.72200037 | 4.68571 | Dinoconda | China Dinosaurs Park |
| 4.71729409 | 4.68571 | Great White | SeaWorld San Antonio |
| 4.71562668 | 4.68571 | Voyage | Holiday World |
| 4.70829667 | 4.68571 | Maverick | Cedar Point |
| 4.70266547 | 4.68571 | Wodan Timbur Coaster | Europa Park |
| 4.69972678 | 4.68571 | El Toro | Freizeitpark Plohn |
| 4.69566562 | 4.68571 | Screamer | Scandia Amusement Park |
| 4.69334898 | 4.68571 | Iron Rattler | Six Flags Fiesta Texas |
| 4.68887365 | 4.68571 | Katun | Mirabilandia |
| 4.68407238 | 4.68571 | Wicked Cyclone | Six Flags New England |
| 4.68204327 | 4.68571 | Taron | Phantasialand |
| 4.67979935 | 4.68571 | Superman the Ride | Six Flags New England |
| 4.62402991 | 4.66316 | Leviathan | Canada's Wonderland |
| 4.60768361 | 4.61667 | Mako | SeaWorld Orlando |
| 4.67198947 | 4.56299 | Taunusblitz | Taunus Wunderland |
| 4.66709971 | 4.56299 | Demon | Six Flags Great America |
| 4.65862381 | 4.56299 | Spatiale Experience | Nigloland |
| 4.62829192 | 4.56299 | Shock Wave | Six Flags Over Texas |
| 4.51840222 | 4.56299 | Phoenix | Knoebels Amusement Park |
| 4.40930736 | 4.56299 | Vortex | Kings Island |
| 4.28909299 | 4.56299 | Flight Deck | Canada's Wonderland |
| 4.27182751 | 4.56299 | Thunder Dolphin | Tokyo Dome City |
| 4.26692164 | 4.56299 | Batman The Ride | Six Flags Great Adventure |
| 4.24676007 | 4.56299 | Cyclone | Lakeside Amusement Park |
| 4.20131376 | 4.56299 | Superman / la Atracción de Acero | Parque Warner Madrid |
| 4.41895206 | 4.54397 | Banshee | Kings Island |
| 4.59451963 | 4.54321 | Storm Chaser | Kentucky Kingdom |
| 4.50711877 | 4.51163 | Behemoth | Canada's Wonderland |
| 4.50503851 | 4.51163 | Joker | Six Flags Discovery Kingdom |
| 4.67616868 | 4.50813 | Goliath | Six Flags Great America |
| 4.65658529 | 4.50813 | Montezum | Hopi Hari |
| 4.64953369 | 4.50813 | Ravine Flyer II | Waldameer |
| 4.63300263 | 4.50813 | Outlaw Run | Silver Dollar City |
| 4.62320178 | 4.50813 | Boulder Dash | Lake Compounce |
| 4.54304209 | 4.50813 | New Texas Giant | Six Flags Over Texas |
| 4.50768788 | 4.50813 | Diamondback | Kings Island |
| 4.5023516 | 4.50813 | Mystic Timbers | Kings Island |
| 4.48568363 | 4.50813 | Medusa Steel Coaster | Six Flags Mexico |
| 4.46380478 | 4.50813 | Nitro | Six Flags Great Adventure |
| 4.45337255 | 4.46222 | Intimidator | Carowinds |
| 4.4307661 | 4.43777 | Top Thrill Dragster | Cedar Point |
| 4.58992314 | 4.42857 | Apocalypse the Ride | Six Flags Magic Mountain |
| 4.5237153 | 4.42857 | Steel Eel | SeaWorld San Antonio |
| 4.47451966 | 4.42857 | Goliath | Six Flags Over Georgia |
| 4.43569959 | 4.42231 | X2 | Six Flags Magic Mountain |
| 4.38948843 | 4.39308 | Firewhip | Beto Carrero World |
| 4.38948843 | 4.39308 | Raptor | Fantasilandia |
| 4.3819371 | 4.39308 | Intimidator 305 | Kings Dominion |
| 4.6724907 | 4.38889 | Time Traveler | Silver Dollar City |
| 4.59561861 | 4.38889 | American Eagle | Six Flags Great America |
| 4.53588273 | 4.38889 | RailBlazer | California's Great America |
| 4.39780638 | 4.38889 | Bizarro | Six Flags Great Adventure |
| 4.37058773 | 4.38889 | Manta | SeaWorld Orlando |
| 4.39043002 | 4.3881 | Montu | Busch Gardens Tampa |
| 4.34981481 | 4.3881 | Timber Drop | Fraispertuis City |
| 4.66160605 | 4.37838 | Mammut | Erlebnispark Tripsdrill |
| 4.64420473 | 4.37838 | Oblivion | Alton Towers |
| 4.64214495 | 4.37838 | Superman - Ultimate Flight | Six Flags Great America |
| 4.63806862 | 4.37838 | Alpina Blitz | Nigloland |
| 4.56457103 | 4.37838 | Flash | Lewa Adventure |
| 4.4709039 | 4.37838 | Big One | Blackpool Pleasure Beach |
| 4.46320546 | 4.37838 | Rock 'n' Roller Coaster | Disneyland Paris - Walt Disney Studios Park |
| 4.36906464 | 4.37838 | Phantom's Revenge | Kennywood |
| 4.3671142 | 4.37838 | Poltergeist | Six Flags Fiesta Texas |
| 4.34860096 | 4.3591 | Apollo's Chariot | Busch Gardens Williamsburg |
| 4.34932918 | 4.34434 | Tatsu | Six Flags Magic Mountain |
| 4.28819617 | 4.31104 | Griffon | Busch Gardens Williamsburg |
| 4.32757797 | 4.30056 | Storm Runner | Hersheypark |
| 4.36589051 | 4.28594 | Flying Aces | Ferrari World Abu Dhabi |
| 4.28375259 | 4.28594 | Beast | Kings Island |
| 4.27319958 | 4.24891 | Skyrush | Hersheypark |
| 4.19819596 | 4.2069 | Prowler | Worlds of Fun |
| 4.19192301 | 4.19718 | Superman - Ride Of Steel | Six Flags America |
| 4.19102035 | 4.19718 | Boss | Six Flags St. Louis |
| 4.15581122 | 4.12821 | Renegade | Valleyfair! |
| 4.12180975 | 4.12397 | Ride of Steel | Darien Lake |
| 4.12005868 | 4.12076 | Raptor | Cedar Point |
| 4.10768203 | 4.12016 | Wild One | Six Flags America |
| 4.10462369 | 4.12016 | Afterburn | Carowinds |
| 4.12770836 | 4.11187 | GateKeeper | Cedar Point |
| 4.34634359 | 4.10753 | Pyrenees | Parque Espana-Shima Spain Village |
| 4.32987577 | 4.10753 | Nemisis Inferno | Thorpe Park |
| 4.17347312 | 4.10753 | iSpeed | Mirabilandia |
| 4.14619499 | 4.10753 | Stampida | PortAventura Park |
| 4.12792951 | 4.10753 | Talon | Dorney Park & Wildwater Kingdom |
| 4.09411696 | 4.10526 | Xcelerator | Knott's Berry Farm |
| 4.31723254 | 4.08911 | Desafio | Parque de la Costa |
| 4.26963215 | 4.08911 | Batman the Ride | Six Flags Mexico |
| 4.13894062 | 4.08911 | Superman Krypton Coaster | Six Flags Fiesta Texas |
| 4.06907828 | 4.08824 | Full Throttle | Six Flags Magic Mountain |
| 4.25193541 | 4.0875 | Dragon Mountain | Marineland Theme Park |
| 4.08067718 | 4.0875 | Alpengeist | Busch Gardens Williamsburg |
| 4.04958275 | 4.07302 | Raging Bull | Six Flags Great America |
| 4.14629444 | 4.06499 | Riddler's Revenge | Six Flags Magic Mountain |
| 4.05910171 | 4.06499 | Magnum XL-200 | Cedar Point |
| 4.03555208 | 4.04211 | Comet | Walygator Parc |
| 4.34843187 | 4.04032 | Ranier Rush | Puyallup Fair |
| 4.29762004 | 4.04032 | Expedition GeForce | Holiday Park |
| 4.17210333 | 4.04032 | Soaring Dragon & Dancing Phoenix | Nanchang Wanda Theme Park |
| 4.14370694 | 4.04032 | Soaring with Dragon | Hefei Wanda Theme Park |
| 4.0781567 | 4.04032 | blue fire Megacoaster | Europa Park |
| 4.06196024 | 4.04032 | Velikolukskiy Myasokombinat | Wonder Island |
| 4.04526043 | 4.04032 | Timber Wolf | Worlds of Fun |
| 4.03930645 | 4.04032 | Raven | Holiday World |
| 3.7831679 | 4.04032 | Extreme Rusher | Happy Valley |
| 4.03730789 | 4.03614 | Medusa | Six Flags Discovery Kingdom |
| 4.03648309 | 4.03478 | Kumba | Busch Gardens Tampa |
| 4.00687324 | 4.00382 | Valravn | Cedar Point |
| 3.7069171 | 4.00382 | GhostRider | Knott's Berry Farm |
| 3.97159131 | 3.96875 | Kingda Ka | Six Flags Great Adventure |
| 3.90329942 | 3.96875 | Scream! | Six Flags Magic Mountain |
| 3.87043606 | 3.96875 | Big Apple Coaster | New York, New York Hotel & Casino |
| 1.88781432 | 3.96875 | Bocaraca | Parque de Diversiones |
| 4.01204397 | 3.96835 | Iron Dragon | Cedar Point |
| 4.00220784 | 3.96835 | Timberhawk: Ride of Prey | Wild Waves Theme Park |
| 3.99107834 | 3.96835 | Monster | Walygator Parc |
| 3.98441308 | 3.96835 | Revenge of the Mummy the Ride | Universal Studios Hollywood |
| 2.93061948 | 3.96835 | Kawazemi | Tobu Zoo Park |
| 4.00932638 | 3.94444 | Titan | Six Flags Over Texas |
| 3.96834865 | 3.94253 | Kraken | SeaWorld Orlando |
| 3.42695244 | 3.94253 | Montana Rusa | VulQano Park |
| 3.37684872 | 3.94253 | Quimera | La Feria Chapultpec |
| 2.95635599 | 3.94253 | Tower of Terror II | Dreamworld |
| 2.69454328 | 3.94253 | Titan Cascabel | Selva Magica |
| 2.59486873 | 3.94253 | Montana Rusa | Salitre Magico |
| 2.18047744 | 3.94253 | Katapul | Hopi Hari |
| 3.97533146 | 3.93923 | Legend | Holiday World |
| 3.59844299 | 3.93923 | Doble Loop | Salitre Magico |
| 2.43050132 | 3.93923 | Green Lantern Coaster | Warner Bros. Movie World |
| 1.89932077 | 3.93923 | Whirl Wind Looping Coaster | Wonder Island |
| 4.09617288 | 3.93056 | Giant Dipper | Belmont Park |
| 3.97549644 | 3.93056 | Batman - The Dark Knight | Six Flags New England |
| 3.9571465 | 3.93056 | Adrenaline Peak | Oaks Amusement Park |
| 3.95227031 | 3.93056 | Pandemonium | Six Flags Over Texas |
| 3.92738431 | 3.93056 | Giant Dipper | Santa Cruz Beach Boardwalk |
| 3.92421912 | 3.93056 | Pandemonium | Six Flags Fiesta Texas |
| 3.92004445 | 3.93056 | Smiler | Alton Towers |
| 4.00342987 | 3.92632 | Hades 360 | Mt. Olympus Water & Theme Park |
| 3.98572685 | 3.92632 | Helix | Liseberg |
| 3.88738009 | 3.89189 | Twister II | Elitch Gardens |
| 3.86956736 | 3.89189 | Goliath | Six Flags Magic Mountain |
| 3.80918747 | 3.89189 | Boardwalk Bullet | Kemah Boardwalk |
| 3.76643647 | 3.89189 | Wild Thing | Valleyfair! |
| 3.85695476 | 3.87296 | Fahrenheit | Hersheypark |
| 3.78135111 | 3.82578 | Steel Force | Dorney Park & Wildwater Kingdom |
| 3.77830418 | 3.82578 | Mamba | Worlds of Fun |
| 3.70485405 | 3.82578 | Manta | SeaWorld San Diego |
| 3.37392181 | 3.82578 | Mine Blower | Fun Spot America |
| 3.9989899 | 3.79237 | Whizzer | Six Flags Great America |
| 3.99476547 | 3.79237 | Racer | Kings Island |
| 3.92311009 | 3.79237 | Goudurix | Parc Asterix |
| 3.8972642 | 3.79237 | Texas Tornado | Wonderland Amusement Park |
| 3.87610825 | 3.79237 | New Revolution | Six Flags Magic Mountain |
| 3.84300038 | 3.79237 | Silver Bullet | Knott's Berry Farm |
| 3.80907643 | 3.79237 | Abismo | Parque de Atracciones de Madrid |
| 3.79424031 | 3.79237 | MP-Xpress | Movie Park Germany |
| 3.78204959 | 3.79237 | Patriot | Worlds of Fun |
| 3.78186354 | 3.79237 | Limit | Heide-Park Soltau |
| 3.77483958 | 3.79237 | Star Wars Hyperspace Mountain: Rebel Mission | Disneyland Resort Paris - Disneyland Park |
| 3.75956044 | 3.79237 | Eejanaika | Fuji-Q Highland |
| 3.75922168 | 3.79237 | Coaster Thrill Ride | Puyallup Fair |
| 3.75276808 | 3.79237 | Colorado Adventure | Phantasialand |
| 3.70124053 | 3.79237 | Ninja | Six Flags Magic Mountain |
| 3.70098774 | 3.79237 | Swarm | Thorpe Park |
| 3.69578099 | 3.79237 | Fluch von Novgorod | Hansa Park |
| 3.67404604 | 3.79237 | Rougarou | Cedar Point |
| 3.6288221 | 3.79237 | Judge Roy Scream | Six Flags Over Texas |
| 3.58222864 | 3.79237 | Fly the Great Nor'Easter | Morey's Piers |
| 3.5705784 | 3.79237 | Blue Streak | Cedar Point |
| 3.56054789 | 3.79237 | Cannibal | Lagoon |
| 3.53119815 | 3.79237 | Joker | Six Flags Great America |
| 3.52241667 | 3.79237 | Hydra the Revenge | Dorney Park & Wildwater Kingdom |
| 3.49472305 | 3.79237 | Gemini | Cedar Point |
| 3.49396758 | 3.79237 | Half Pipe | Elitch Gardens |
| 3.46828099 | 3.79237 | Boomerang | Parque de la Costa |
| 3.46054963 | 3.79237 | Boomerang | Fantasilandia |
| 3.44523483 | 3.79237 | Viper | Six Flags Magic Mountain |
| 3.40939578 | 3.79237 | Piraten | Djurs Sommerland |
| 3.40209304 | 3.79237 | Blue Hawk | Six Flags Over Georgia |
| 3.34500379 | 3.79237 | Flight of Fear | Kings Dominion |
| 3.31721989 | 3.79237 | Boomerang | Six Flags Mexico |
| 3.3032695 | 3.79237 | Dragon Khan | PortAventura Park |
| 3.2698137 | 3.79237 | Dragon's Run | Dragon Park |
| 3.19965029 | 3.79237 | Nemesis | Alton Towers |
| 3.13088125 | 3.79237 | Red Force | Ferrari Land |
| 3.12496189 | 3.79237 | Journey to Atlantis | SeaWorld San Antonio |
| 3.07643364 | 3.79237 | Boomerang | Six Flags St. Louis |
| 3.05096659 | 3.79237 | Boomerang | Worlds of Fun |
| 3.05096659 | 3.79237 | Flashback | Six Flags New England |
| 3.04870246 | 3.79237 | Boomerang | Elitch Gardens |
| 3.04870246 | 3.79237 | Boomerang | Six Flags Fiesta Texas |
| 3.04640557 | 3.79237 | Boomerang Coast to Coaster | Six Flags Discovery Kingdom |
| 2.98699938 | 3.79237 | Wicked Twister | Cedar Point |
| 2.96901875 | 3.79237 | Sidewinder | Elitch Gardens |
| 2.9513695 | 3.79237 | Superman: Escape from Krypton | Six Flags Magic Mountain |
| 2.85092248 | 3.79237 | Batman: Arkham Asylum | Parque Warner Madrid |
| 2.81943908 | 3.79237 | Mr. Freeze Reverse Blast | Six Flags Over Texas |
| 2.81943908 | 3.79237 | Mr. Freeze Reverse Blast | Six Flags St. Louis |
| 2.75562501 | 3.79237 | Montana Rusa | La Feria Chapultpec |
| 2.66974378 | 3.79237 | Grizzly | California's Great America |
| 2.48120258 | 3.79237 | Colossus | Thorpe Park |
| 2.32223402 | 3.79237 | Steel Dragon 2000 | Nagashima Spa Land |
| 2.29092273 | 3.79237 | Backlot Stunt Coaster | Kings Island |
| 2.26884184 | 3.79237 | Invertigo | Kings Island |
| 2.24668903 | 3.79237 | Goliath | Six Flags New England |
| 2.17482557 | 3.79237 | V2: Vertical Velocity | Six Flags Discovery Kingdom |
| 2.16322141 | 3.79237 | Vertical Velocity | Six Flags Great America |
| 2.16269415 | 3.79237 | Steel Venom | Valleyfair! |
| 2.02858192 | 3.79237 | Montezooma's Revenge | Knott's Berry Farm |
| 1.85671929 | 3.79237 | Tornado | Bosque Magico |
| 1.58447029 | 3.79237 | Wild Thing | Wild Waves Theme Park |
| 3.96986348 | 0.0274 | Big Loop | Heide-Park Resort |
| 2.73507206 | 0.0274 | SpeedSnake FREE | Fort Fun Abenteuerland |
| 2.73011225 | 0.0274 | Super Tornado | Zoo Safari- und Hollywoodpark Stukenbrock |
| 4.03905686 | 0.0265 | Furius Baco | PortAventura Park |
| 3.13239752 | 0.0253 | Atlantica SuperSplash | Europa Park |
| 2.70278975 | 0.0253 | Stunt Fall | Parque Warner Madrid |
| 3.91773025 | 0.0194 | Crazy Bird | Happy Valley |
| 3.78511248 | 0.0194 | Phaethon | Gyeongju World |
| 3.62715414 | 0.0194 | HeiBe Fahrt | Wild- und Freizeitpark Klotten/Cochem |
| 3.62005421 | 0.0194 | Temple of the Night Hawk | Phantasialand |
| 3.39217977 | 0.0194 | Sky Wheel | Skyline Park |
| 3.33571713 | 0.0194 | Sky Scream | Holiday Park |
| 3.13906223 | 0.0194 | Snow Mountain Flying Dragon | Happy Valley |
| 2.85719893 | 0.0194 | Boomerang | Walibi Rhone-Alpes |
| 2.84003369 | 0.0194 | Boomerang | Freizeit-Land Geiselwind |
| 2.5176192 | 0.0194 | Winjas | Phantasialand |
| 2.3926823 | 0.0194 | 10 Inversion Roller Coaster | Chimelong Paradise |
| 2.37226978 | 0.0194 | Stealth | Thorpe Park |
| 2.19203442 | 0.0194 | Takabisha | Fuji-Q Highland |
| 1.99046582 | 0.0194 | Force One | Schwaben Park |
|  |  |  |  |
|  |  | Bullet Coaster | Happy Valley |
|  |  | Cedar Creek Mine Ride | Cedar Point |
|  |  | Corkscrew | Cedar Point |
|  |  | Happy Angel | Wanda Theme Park |
|  |  | Journey to Atlantis | SeaWorld San Diego |
|  |  | OCT Thrust SSC1000 | Happy Valley |
|  |  | Terminator Salvation: The Coaster | Six Flags Magic Mountain |