Disney’s Reviews Analysis Report

Team Member: Yilin Ye, Tianying Zhao, Xin Li

Executive Summary

This project explores all possible relationships in the Disneyland review dataset from various aspects of exploratory data analysis and data visualization, text classification, and sentiment analysis. In this project, we performed preliminary data cleaning and visualization via Pandas and other regular python packages; in addition to that, we implemented TF-IDF on tokenized reviews to find the association between reviewers' rating scores and their reviews via logistic regression. Finally, we built a pipeline via Pyspark and used sentiment analysis to extract their review attitudes from the reviews and explore the relationship between the reviewers' ratings and the true attitudes behind their reviews, which adds a touch of humanity beyond the regression model.

Motivation & Goal

With the development of the online platforms, customers could do more than just basic interactions with the businesses, but adding reviews and ratings after the using experience. While the comment section is beneficial for the customers for they could get more information before they buy the product to avoid unnecessary dissatisfaction or scams, it is also beneficial for the business owners that they could get great suggestions from the reviews. However, most of the time customers will provide both their comments and their rating scores while those two sections are providing the same information: How well does the customer like the product. So this project will explore if there is a correlation between the comments and the rating scores, and if it is possible to use only one to fully represent another.

There are several business researchers who have tried to explore the relationship between the comments and ratings before[1], and here in this project we use data from Kaggle and no papers have conducted the same methods and modelings to the same dataset yet.

This project will contain both data analytics and modelings.

Data

This Disneyland reviews dataset is a public dataset from Kaggle, which contains 42,000 reviews across three Disneyland branches (Paris, California, and Hong Kong). This dataset contains six columns: Review_ID, Rating, Year_Month, Reviewer_Location, Review_Text, Disneyland_Branch. Review ID is the primary key of this dataset, and each review corresponds to a unique Review ID, Rating and Review_Text are customer ratings and reviews for each visit.

The Disneyland branch column tells us which Disneyland, reviewer_loccation tells us where the reviewer is from. And Year_Month tells us when Disneyland got the review.

Exploratory Data Analysis

We first checked the percent of missing data. We found that only the column named Year_Month has missing values which account for 6.13%. After dropping the duplicated rows based on Review_ID, our data decreased from 42,656 to 42,636.

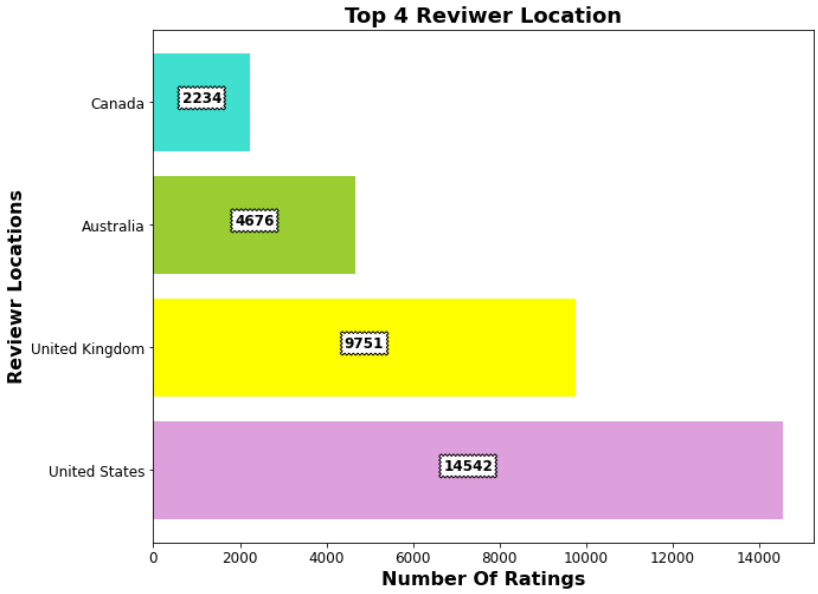


Figure 1

Then we checked which reviewers would like to leave a rating. As shown in Figure 1, we can see that visitors from the United States would like to leave ratings most, then people from the United Kingdom also liked to leave ratings. Then visitors from Australia and Canada also would like to do that.

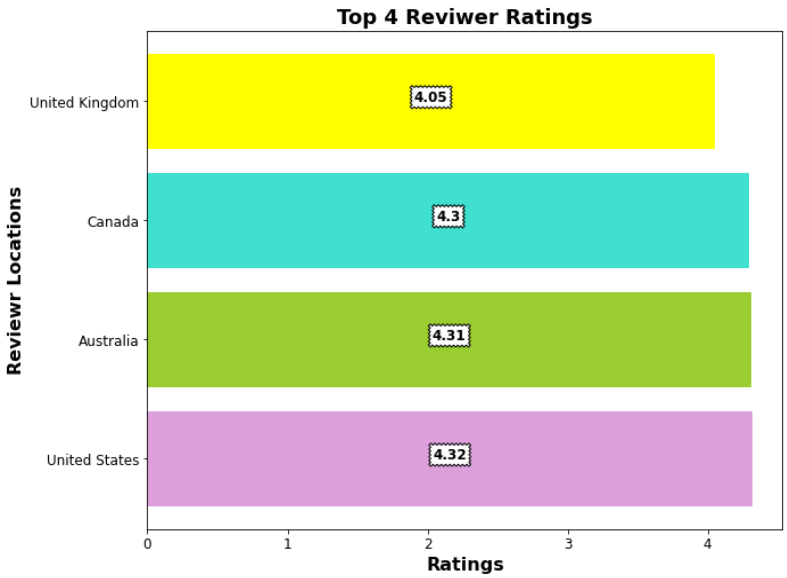


Figure 2

We also checked their average ratings shown as figure 2. We can see that people from the US gave the highest rating. People from the UK gave the lowest rating compared with the other 3 countries. Overall, the average ratings across these 4 countries were very close.

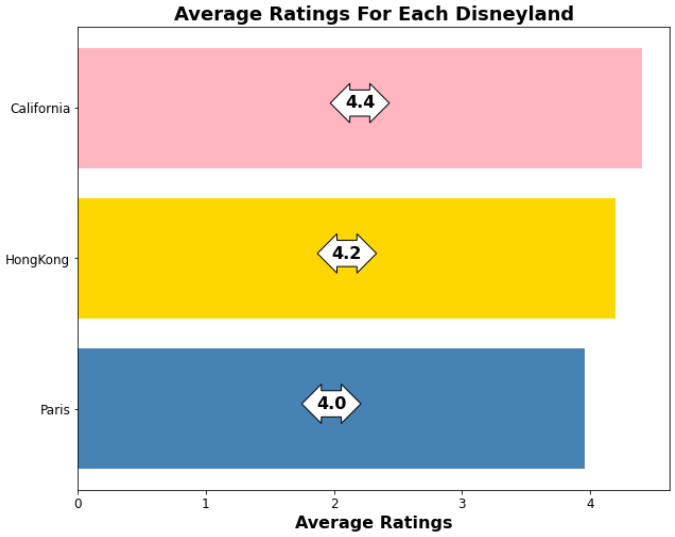


Figure 3

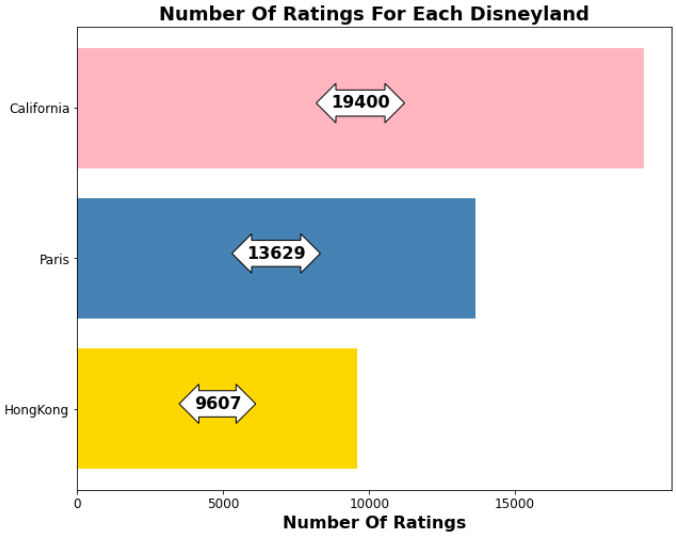


Figure 4

We plotted a bar chart for checking the average ratings for each park. From figure 3, we can see that the results were very consistent with Figure 4. California park had higher average ratings than the other 2 parks. The branch in Hong Kong was average and the branch in Paris was the last one among the 3 parks. However, if we take a look at figure 4, we can see that the number

of comments was different across the three branches. California park had twice the number of ratings as the Hong Kong branch had.

Figure 5

Figure 6

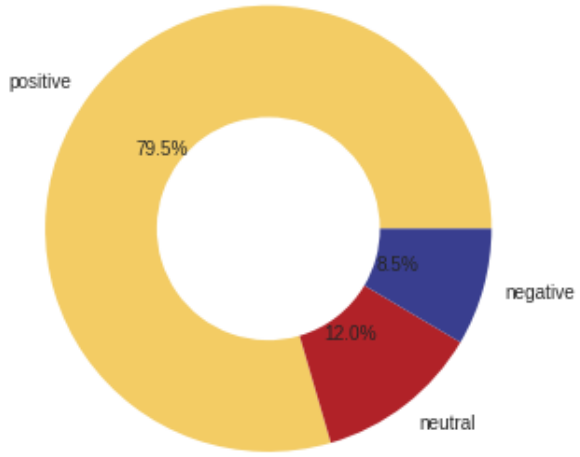
Finally, we plotted the most frequent words in each rating score. From Figure 5 and 6 we can see that for positive comments, reviewers mentioned Disneyland, rides, amazing and great. For negative comments, they mentioned minutes, staff, queue, and ticket. We can speculate that what annoyed people was a large number of visitors, long queues and too much waiting time. And it is interesting that what impressed the visitors, which was the rides, also made people wait for a long time and then become impatient and leave negative reviews.

Text Classification & Logistic Regression

After an initial exploratory data analysis, we can see that the ratings range from 0 to 5, but most of the keywords are the same for each rating score. Therefore, in this section, we decided to use a classification model to classify the ratings by reviews to see if there is a deterministic relationship between ratings and reviews.

Methodology

Initially, we tried a multi-category classification where we divided the ratings into 3 different categories, with ratings 1 to 2 being negative, 3 being neutral, and 4 to 5 being positive. However, as shown in Figure 7, 79.5% of the ratings were positive, 12% were neutral, and only 8.5% were negative. This percentage of the dataset is very unbalanced, and classification is even more difficult when most of the categories are minority. On top of that, we are not familiar with text classification and its associated natural language model. Therefore, it is more practical to combine neutral and negative into a negative category, making the task a binary classification. After that, we can see that even though the percentage of each category is unbalanced, it is still better than the original classification.



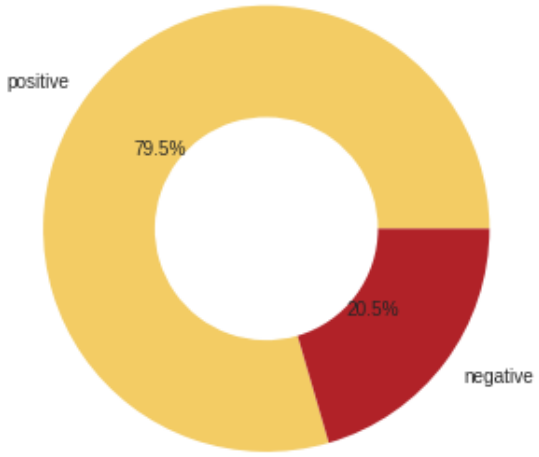


Figure 7

Figure 8

After we labeled the data. We use TF-IDF (Term Frequency, Inverse Document Frequency) to vectorize each comment into a feature vector. By using TF-IDF, we can know what a word does to a sentence and a document, and it also makes the model more robust to misspelled words.The method behind TF-IDF is that we count the words in all comments and divide by the total number of words to get the term frequency. After that, we calculate the inverse document frequency, which is the logarithmic ratio of the count of a word to the number of documents in which the word appears. Then we multiply TF by IDF to get our final value for each vector.
The last step is to classify the ratings by the feature vectors of the reviews. The model we use is the classical logistic regression.

Result

However, the results are frustrating. recall is a good indicator of model performance for unbalanced datasets, and after I set the threshold of the model to 0.01, the recall score is only 0.67 and the accuracy is about 0.75. It is obvious that TF-IDF and the model cannot show the relationship between ratings and reviews. This may be because TF-IDF is only a method to calculate the word frequency and normalize the feature vector of each comment. It does not show the sentiment and emotion behind each word, and much information is missing in this model. The use of SMOTE to up-sample the dataset is also not applicable to our case because the TF-IDF vectors already ignore most of the sentiment and emotion meanings, and up-sampling based on these feature vectors would be even worse.

Sentiment Analysis

With the result from the previous section that TF-IDF cannot fully capture the relationship between the ratings and reviews, we conducted the sentiment analysis on the reviews to see if such a relationship does exist.

Data Preprocessing

Before we feed the raw review data into the sentiment analysis model, we have to transform the data into the desired format. In order to achieve it, the spark pipeline was being used. First, we use tokenizer to transform the reviews into lower case words. Second, we use the StopWordsRemover to eliminate the stop words including short function words or prepositions. Then we concat all the words back to a sentence form and remove all the undesired punctuations. By eliminating the stop words, we can increase the accuracy and the efficiency of the sentiment analysis model.

Modeling

We feed our data into the natural language toolkit package, which is a pre-designed python model for sentiment analysis. The model would return a polarity score ranging from negative 1 to positive 1 describing how negative or positive the sentence is. A score of near 0 means the sentence is neutral.

Result

After we feed the data into the model, we group our data by the ratings to see the polarity score distribution for every different rating.

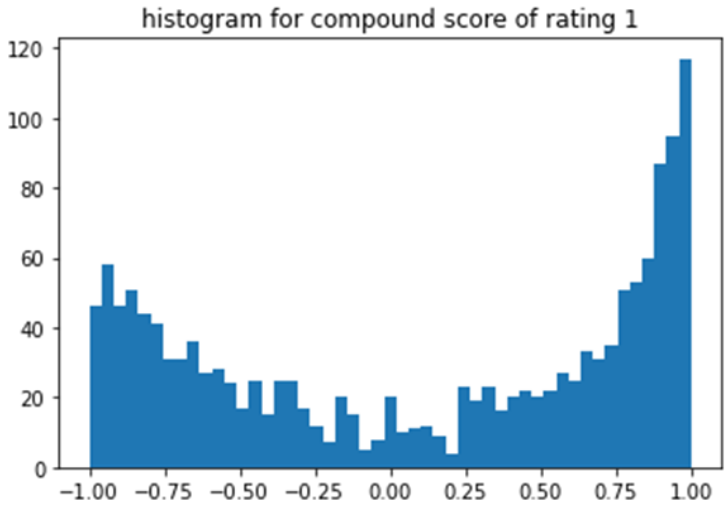


Figure 9

However, from the figure 9 we could see that the distribution of the scores with rating 1 is different from our expectations. What we expect to see for the rating 1 is that most scores should be gathered around negative, while the figure shows that more than half of the scores

are positive, and this could be seen as evidence that the rating scores might not have a strong connection with the review sentiments.

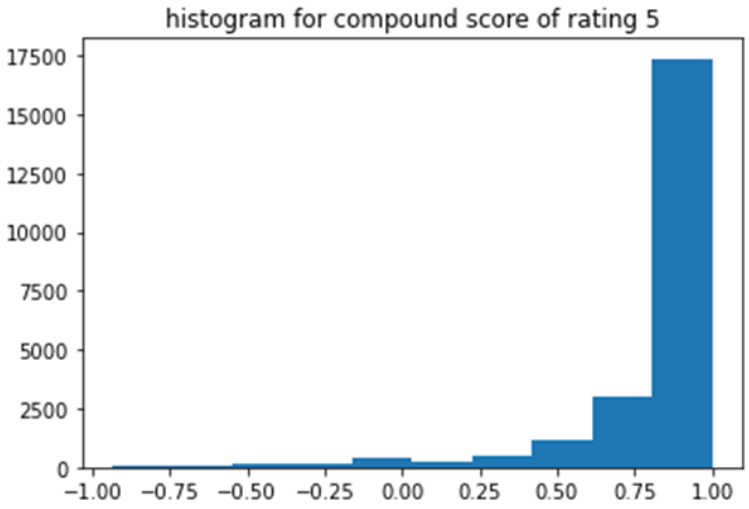


Figure 10

Figure 10 shows the distribution of the scores with rating 5, and we could see that most scores are around 1, which is meeting our expectation that people giving a best score will write a very positive review. However, we can also observe that some negative reviews still exist.

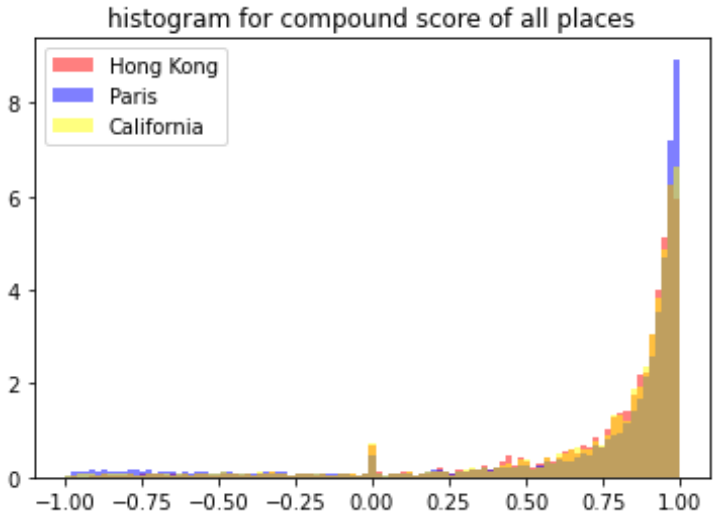


Figure 11

Finally, figure 11 shows the score distribution under three park branches, and we could observe the fact that although all three parks have almost overlapping score distributions, the Paris park is having the most reviews with score 1. Although from the EDA step we have shown that Paris park is having the lowest average ratings, their review sentiments are having the best result. So,

we could conclude that the reviews do not have a very strong connection with the ratings and ratings itself can not represent how well the park operates alone.

Conclusion

From the EDA steps, we had our basic understanding on how the data was distributed. Then, to find the correlation between the ratings and the reviews, we first used TF-IDF technique to vectorize the reviews and fed it into the logistic regression model. However, since the recall is low despite the high accuracy, we can not find strong evidence to say the ratings and the reviews are strongly related. We then conducted the sentiment analysis based on the NLTK package and found out that the sentiment score for the reviews are not distributed the same as the rating; in fact more than half of the reviews in rating 1 is having a positive sentiment. In all, we conclude that the ratings for Disneyland park from our dataset can not fully represent the rating scores, and they do not have a strong relationship.

Future steps

In order to further understand the customer reviews, we extracted some reviews from the top scores in rating 1 and the lowest scores in rating 5, and we found out that those reviews are actual complaints from loyal customers. So for the future study, extracting those reviews based on the sentiment scores might help the business owner get valuable suggestions.

References

[1]: Ramachandran, Rahul, et al. “Exploring the Relationship between Emotionality and Product Star Ratings in Online Reviews.” *IIMB Management Review*, Elsevier, 9 Dec. 2021, https://www.sciencedirect.com/science/article/pii/S0970389621001178.

Github Repo Link

https://github.com/Yilin-Ye/big_data_disney_review