# ANALYSIS OF DISNEYLAND REVIEWS USING NLP

RAJAT RAUTELA SAURABH BHADALE

UNDER THE SUPERVISION OF SOURAV SAHA

LILY POLEKAD

# SUNANDAN DIVATIA SCHOOL OF SCIENCE

SVKM’S Narsee Monjee Institute of Management Studies (Deemed-To-Be University)

Vile Parle West, Mumbai, Maharashtra 400056 APRIL, 2021

**Acknowledgement**

We would like to express our warmest thank of gratitude to our supervisors Mr. Sourav Saha and Ms. Lily Polekad for providing guidance throughout this project. Their invaluable guidance and advice helped us to carry out the project successfully.

We would like to thank our course coordinator Dr. Pradnya Khandeparkar for providing necessary facilities required for completion of this project.

We would also like to extend our gratitude to the whole Department of Statistics for providing us the opportunity to carry out the project ‘Analysis of Disneyland Reviews using NLP’.

Finally, with great support of our team members and excellent guidance of mentors we were able to complete our project within the specified time.

# ABSTRACT

Sentiment Analysis or Opinion Mining is the computational study of opinions or emotions towards aspects or things. The aspects are nothing but attributes or components of the individuals, events, topics, products and organizations. Sentiment Analysis has been an active research area in Web Mining and Natural Language Processing (NLP) in recent years. Online reviews are now popularly used for judging quality of product or service and influence decision making of users while selecting a product or service. In this project we attempt to analyse reviews for Disneyland Parks in three locations: Anaheim (California), Paris (France) and Hong Kong. We will attempt to find relevant business insights that will in turn help improve public perception of the company and their attendance.

# CONTENT

|  |  |
| --- | --- |
| **Sr. No** | **Topic** |
| **1.** | Glossary |
| **2.** | Introduction Rationale and Scope Objectives  Literature Reviews |
| **3** | Data Collection  Data Description |
| **5** | Methodology   * 1. Data cleaning   2. Data Pre-Processing   3. Vectorization   4. Classification   5. Association Rule Mining   6. Polarity and Subjectivity   7. Outcomes |
| **6** | Conclusion  Business Insights |
| **7** | Limitations  Future Scope of the work |
| **8** | References |

**GLOSSARY**

# Recall/Sensitivity

Recall is the number of True Positives (TP) divided by the Total Number of True Positives (TP) and False Negatives (FN). It tells us that of all the actual positive examples out there, how many of them have been correctly predicted to be positive.

Recall =

TP TP + FN

# Specificity

Specificity is the number of True Negatives (TN) divided by the Total Number of True Negatives (TN) and False Positives (FP). It tells us that of all the actual negative examples out there, how many of them have been correctly predicted to be negative.

Specificity =

# Precision

TN TN + FP

Precision is the number of True Positives (TP) divided by the Total Number of True Positives (TP) and False Positives (FP). It tells us that of all the positive predictions that have been made, how many of them are truly positive.

Precision =

# F1 score

TP TP + FP

To evaluate model performance comprehensively, we should examine **both** precision and recall. The F1 score serves as a helpful metric that considers both of them. It is the harmonic mean of precision and recall, that provides a more balanced summarization of model performance.

F1 score = 2 × Precision×Recall

OR F1 score = TP

Precision+Recall

1 ( )

TP+2× FP+FN

# Support

Support refers to the number of actual occurrences of the class in the dataset.

# Macro average

The macro-averaged F1 score (or macro F1 score) is computed by taking the arithmetic mean (aka unweighted mean) of all the per-class F1 scores

This method treats all classes equally regardless of their support values.

# Weighted average

The weighted-average F1 score is calculated by taking the mean of all per-class F1 scores while considering each class’s support.

The ‘weight’ essentially refers to the proportion of each class’s support relative to the sum of all support values.

# Micro average / Accuracy

Micro averaging computes a global average F1 score by counting the sums of the True Positives (TP), False Negatives (FN), and False Positives (FP).

The reason that accuracy and micro-averaging are the same is that micro-averaging essentially computes the proportion of correctly classified observations out of all observations. This definition is in fact what we use to calculate overall accuracy.

Accuracy =

TP + TN

TP + FP + TN + FN

Note: These results mean that in multi-class classification cases where each observation has a single label, the micro-F1, micro-precision, micro-recall, and accuracy share the same value and hence only needs to display a single accuracy value.

# Hyperparameter tuning

When creating a machine learning model, we are presented with design choices as to how to define our model architecture. Often times, we don't immediately know what the optimal model architecture should be for a given model, and thus we'd like to be able to explore a range of possibilities.

In machine learning, ideally, we would like the machine to perform exploration to select the optimal model architecture automatically. Parameters which define the model architecture are referred to as hyperparameters and thus this process of searching for the ideal model architecture is referred to as hyperparameter tuning.

# INTRODUCTION

The Walt Disney Company was founded in 1923 by brothers Walt Disney and Roy O. Disney. Today, its headquarters are found in Burbank, California. Disney is made up of five major segments, including Walt Disney Studios, Parks and Resorts, Disney Consumer Products, Media Networks, and Disney Interactive. Disney’s theme parks and cruise line are maintained under the Parks and Resorts division. In Florida,

Disney’s Magic Kingdom was the most visited amusement park in the world in 2019, with approximately

20.96 million attendees. Disney emphasizes an image campaign that advertises Disney World as the “Happiest Place on Earth”, spending 1.78 billion U.S. dollars on advertising and marketing campaigns in 2017. In Europe, the company’s revenue totaled 7.03 billion U.S. dollars. However, its media networks, as well as parks and resorts, generated a significant portion of its total revenue at 50.87 billion U.S. dollars and 16.55 billion U.S. dollars, respectively in 2021.

In this project, we try to classify customer sentiments regarding their experience at the three Disneyland locations (California, Paris, and Hong Kong) using Sentiment Analysis and provide business insights to different branches by analysing the negatively classified reviews further and help Disney determine visitor pain points and improve park experience.

# RATIONALE AND SCOPE

Sentiment analysis is a powerful marketing tool that enables product managers to understand customer emotions in their marketing campaigns. It is an important factor when it comes to product and brand recognition, customer loyalty, customer satisfaction, advertising and promotion's success, and product acceptance. Understanding the psychology of consumers can help product managers and customer success managers to alter their product roadmap with greater precision. The term emotion-based marketing is a broad umbrella phrase that encompasses emotional customer responses, such as "positive," "negative," "neutral," "negative," "uptight," "disgust," "frustration," and others. Understanding the psychology of customer responses can also increase product and brand recall.

When it comes to creating an amazing customer experience, all companies can learn from the Happiest Place on Earth. With its magical and personalized approach to customer experience, Disney and its theme parks have created a passionately loyal fan base, welcoming 157 million visitors in 2018 with an amazing 70% return rate of first-time guests.

As for the scope of the project, we limit ourselves to statistical and machine learning models.

# OBJECTIVES

In this project we aim to:

* Identify key problem areas across three Disneyland locations (California, Paris, and Hong Kong).
* To help the business implement the suggestions by identifying the customer pain points
* **Project Goal**: To compare customer sentiment regarding attractions at using sentiment analysis. To identify keywords that can help Disney understand the visitor pain points and improve overall park experience.

# LITERATURE REVIEW

In [1] the paper elaborately discusses two supervised machine learning algorithms: K-Nearest Neighbour(K- NN) and Naïve Bayes’ and compares their overall accuracy, precisions as well as recall values. It was seen that in case of movie reviews Naïve Bayes’ gave far better results than K-NN but for hotel reviews these algorithms gave lesser, almost same accuracies.

In [2] Natural Language Processing for sentiment analysis is utilized based on six machine learning algorithms like Logistic Regression, Support Vector Classifier, Random Forest, k-NN. Further, Recurrent Neural Networks and LSTMs leverage the power of deep learning to achieve higher accuracy and better results

In [3] the sentiment that refers to the specific subject are detected using Natural Language Processing techniques. To classify sentiments, it consists of three main steps; subjectivity classification, semantic association and polarity classification.

# DATA COLLECTION

We used a dataset found in Kaggle containing reviews from TripAdvisor about three Disneyland branch locations: California, Paris, Hong Kong. This data is based on reviews obtained starting from 2009 – 2019.

# DATA DESCRIPTION

There are 42,000 reviews from trip advisor in this dataset.

There are 19,406 reviews about California, 13,630 about Paris

9,620 about Hong Kong.

There are following 6 variables:

1. Review\_ID: unique id given to each review
2. Rating: ranging from 1 (unsatisfied) to 5 (satisfied)
3. Year\_Month: when the reviewer visited the theme park
4. Reviewer\_Location: country of origin of visitor
5. Review\_Text: comments made by visitor
6. Disneyland\_Branch: location of Disneyland Park

# DATA PRE-PROCESSING

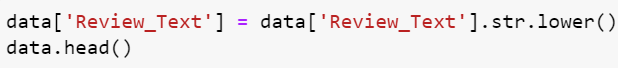
**Original Dataset:**



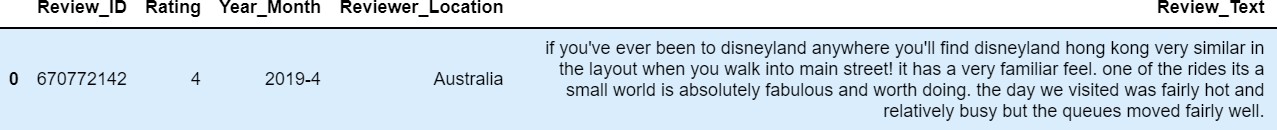
# Following are the steps to treat the textual review data:

1. **Lowering the text:**

One of the most common pre-processing steps where the text is converted into the same case preferably lower case.



# OUTPUT:



We can see that compared to original data in the “Review\_text” column all the first letters of the words have been converted to lower case, this is done because Python reads “If” and “if” as 2 different words (due to different ASCII values), even though they have the same meaning.

# White Space Removal:

White space is any section of a document that is unused or space around an object. This is also problematic because whitespace has an ASCII value of it’s own which in turn causes for example, “if” and “ if” to be considered as two distinct words.



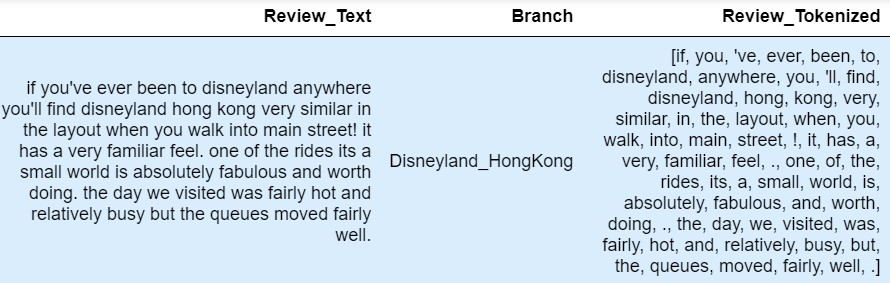
It’s very hard to notice this change and this is just an intermediary step, so showing output for this step won’t be very helpful.

# Tokenization:

In this step, the text is split into smaller units, for example, paragraphs are broken down into sentences and sentences are broken down in words. We can use either sentence tokenization or word tokenization based on our problem statement.



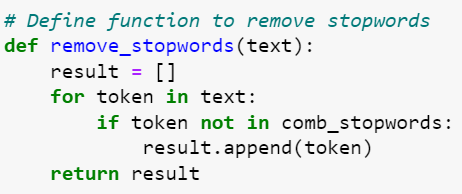
# OUTPUT:



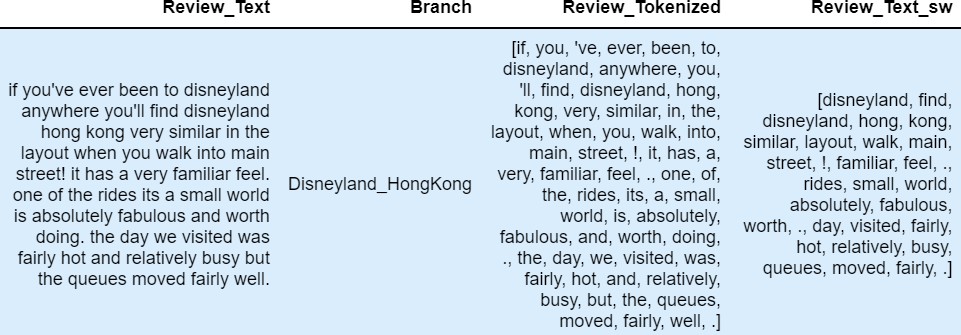
We can see that the sentences have been broken down into individuals words some of which are meaningful and some of which are not that meaningful (you’ve = “you”,“’ve”).

# Stop word removal:

Stopwords are the commonly used words and are removed from the text as they do not add any value to the analysis. These words carry less or no meaning.



# OUTPUT:

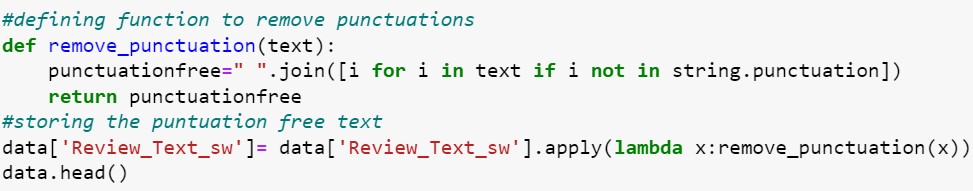


There are many stopword dictionaries in Python, for this process we have joined the SpaCy and NLTK sets of Stopwords, in order to create a more exhaustive dictionary.

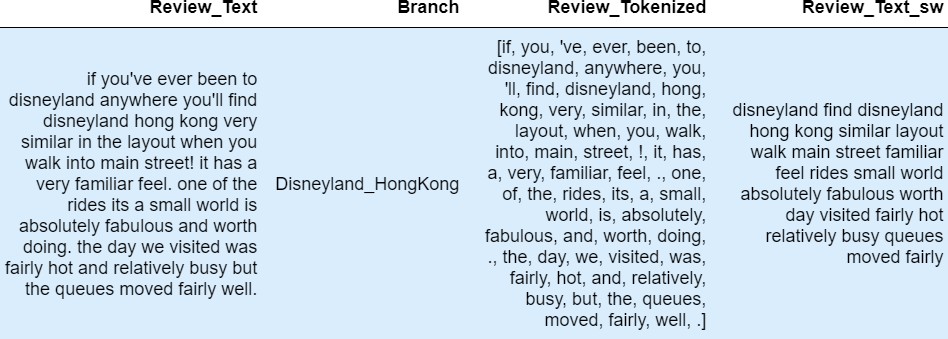
We can see in the “Review\_text\_sw” column that only meaningful words remaining in the list. There are still some punctuations that are not desirable, which are removed in the next step.

# Punctuation Removal:

In this step, all the punctuations from the text are removed. string library of Python contains some pre- defined list of punctuations such as **‘!”#$%&'()\*+,-./:;?@[\]^\_`{|}~’**



# OUTPUT:



We can see that the punctuations has been removed when we compare Review\_Tokenized with Review\_Text\_sw

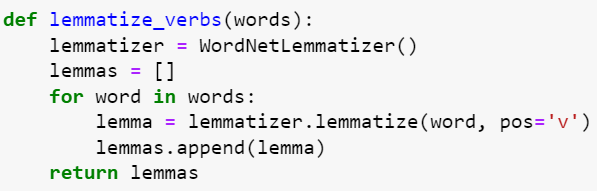
# Stemming:

Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. However, sometimes the words lose their meaning after stemming. For example: It will reduce flies to fli; which has no meaning.

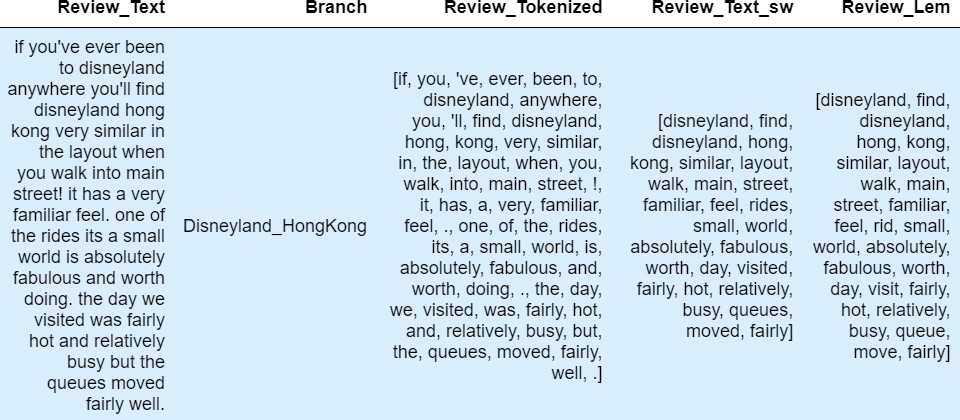
OR

# Lemmatization:

It stems the word but makes sure that it does not lose its meaning. Lemmatization has a pre-defined dictionary that stores the context of words and checks the word in the dictionary while diminishing.

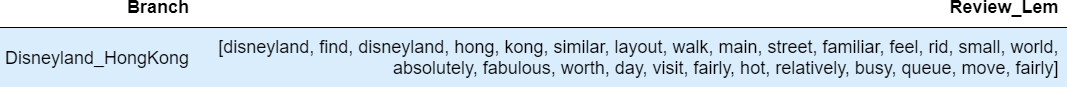


# OUTPUT:



After completing the text pre-processing, further we formed the word clouds using our pre-processed text.

# FINAL DATA:



**METHODOLOGY**

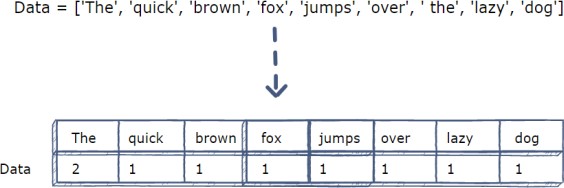
# Feature engineering

* 1. **Count Vectorization:**

After tokenization, the words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called feature extraction (or vectorization).

This method creates vectors that have a dimensionality equal to the size of our vocabulary, and if the text data features that vocab word, we will put a one in that dimension. Every time we encounter that word again, we will increase the count, leaving 0s everywhere we did not find the word even once.

Example:



# TF-IDF:

TF-IDF stands for “Term Frequency — Inverse Document Frequency”. This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

Basically, TF-IDF = Term Frequency (TF) \* Inverse Document Frequency (IDF)

**Term Frequency:** This measures the frequency of a word in a document. This highly depends on the length of the document and the generality of the word, as more general words will appear more often across a longer document, so (say) if we have a short document (1,000 words) and a long document (100,000 words), we implicitly assign more weight to longer document words, when in reality we don’t have an idea as to if which of the two documents is more important. To combat this problem, we perform normalization on the frequency value, by dividing the frequency of each word with the total number of words in the document.

Finally, we need to vectorize the document. When we plan to vectorize documents, we cannot just consider the words that are present in any particular document. If we do that, then the vector length will be different for all the documents we have, and it will not be feasible to compute their similarity. So, what we do is that we vectorize the documents on the vocab. Vocab are the list of all possible words in the corpus. Corpus basically means the total document set that we have.

TF is individual to each document and word; hence we can formulate TF as follows: tf(t,d) = count of t in d / number of words in d

**Document Frequency:** This measures the importance of documents in a whole set of the corpus. This is very similar to TF but the only difference is that TF is the frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term is present in the document at least once, we do not need to know the number of times the term is present.

df(t) = occurrence of t in N documents

To keep this also in a range, we normalize by dividing by the total number of documents. Our main goal is to know the informativeness of a term, and DF is the exact inverse of it.

**Inverse Document Frequency:** IDF is the inverse of the document frequency which measures the informativeness of term t. This finally gives what we want, a relative weightage.

idf(t) = N/df(t)

# Word2Vec (Word Embeddings)

Word Embedding is a set of language modeling techniques for mapping words to a vector of numbers. It’s just a fancy way of saying a numeric vector represents a word.

Word2Vec creates vectors of the words that are distributed numerical representations of word features – these word features could comprise of words that represent the context of the individual words present in our vocabulary. Word embeddings eventually help in establishing the association of a word with another similar meaning word through the created vectors.

Even though Word2Vec is an unsupervised model where we can give a corpus without any label information and the model can create dense word embeddings, Word2Vec internally leverages a supervised classification model to get these embeddings from the corpus.

The CBOW (Continuous Bag of Words) architecture comprises a deep learning classification model in which we take in context words as input, X, and try to predict our target word, Y.

# Sentiment Polarity and Subjectivity

Sentiment polarity for an element defines the orientation of the expressed sentiment, i.e., it determines if the text expresses the positive, negative or neutral sentiment of the user about the entity in consideration. The magnitude of the Polarity value defines the intensity of the particular sentiment. The value lies between [-1, 1].

Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1].

# Classification Models

* 1. **Naïve-Bayes**

This algorithm is called “Naïve” because it makes a naive assumption that each feature is independent of other features which is not true in real life.

As for the “Bayes” part, it refers to the statistician and philosopher, Thomas Bayes and the theorem named after him, Bayes’ theorem, which is the base for Naive Bayes Algorithm.

Therefore, Naïve Bayes algorithm can be defined as a supervised classification algorithm which is based on Bayes theorem with an assumption of independence among features.

Let’s look at the equation for Bayes’ Theorem,

P(A|B) =

P(B|A) × P(A) P(B)

where,

* P(A|B) is the probability of hypothesis A given the data B. This is called the posterior probability.
* P(B|A) is the probability of data B given that the hypothesis A was true.
* P(A) is the probability of hypothesis A being true (regardless of the data). This is called the prior probability of A.
* P(B) is the probability of the data (regardless of the hypothesis).

# How does Naive Bayes Algorithm work?

Let us take an example to understand how does Naive Bayes Algorithm work.

A step-by-step high-level overview of the algorithm (without any involved mathematics):

* For each class calculate the probability of the given instance belonging to it.
* After calculation for all the classes, we check all the calculated values and select the largest value.
* The largest value (highest probability) is selected because it implies that it has the highest probability to actually belong to that class. So, this class is selected.

# Complement Naïve Bayes

Complement Naïve Bayes is somewhat an adaptation of the standard Multinomial Naïve Bayes algorithm. Multinomial Naïve Bayes does not perform very well on imbalanced datasets. Imbalanced datasets are datasets where the number of examples of some class is higher than the number of examples belonging to other classes. This means that the distribution of examples is not uniform. This type of dataset can be difficult to work with as a model may easily overfit this data in favour of the class with a greater number of examples.

# How CNB works:

Complement Naive Bayes is particularly suited to work with imbalanced datasets. In complement Naive Bayes, instead of calculating the probability of an item belonging to a certain class, we calculate the probability of the item belonging to all the classes. This is the literal meaning of the word, complement and hence is called Complement Naive Bayes.

A step-by-step high-level overview of the algorithm (without any involved mathematics):

* + - For each class calculate the probability of the given instance not belonging to it.
    - After calculation for all the classes, we check all the calculated values and select the smallest value.
    - The smallest value (lowest probability) is selected because it is the lowest probability that it is NOT that particular class. This implies that it has the highest probability to actually belong to that class. So, this class is selected.

Note: We don’t select the one with the highest value because we are calculating the complement of the probability. The one with the highest value is least likely to be the class that item belongs to.

# Random Forest:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

# Working of Random Forest Algorithm

Before understanding the working of the random forest, we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model.

# Bagging

Bagging, also known as Bootstrap Aggregation is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as row sampling. This step of row sampling with replacement is called bootstrap. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as aggregation.

# Steps involved in Random Forest algorithm:

**Step 1**: In Random Forest n number of random records are taken from the data set having k number of records.

**Step 2**: Individual decision trees are constructed for each sample.

**Step 3**: Each decision tree will generate an output.

**Step 4**: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

# XG Boost

XG Boost is an ensemble learning method. Sometimes, it may not be sufficient to rely upon the results of just one machine learning model. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. The resultant is a single model which gives the aggregated output from several models.

The models that form the ensemble, also known as base learners, could be either from the same learning algorithm or different learning algorithms. Bagging and boosting are two widely used ensemble learners. Though these two techniques can be used with several statistical models, the most predominant usage has been with decision trees.

# Boosting

In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals.

The base learners in boosting are weak learners in which the bias is high, and the predictive power is just a tad better than random guessing. Each of these weak learners contributes some vital information for prediction, enabling the boosting technique to produce a strong learner by effectively combining these weak learners. The final strong learner brings down both the bias and the variance.

In contrast to bagging techniques like Random Forest, in which trees are grown to their maximum extent, boosting makes use of trees with fewer splits. Such small trees, which are not very deep, are highly interpretable. Parameters like the number of trees or iterations, the rate at which the gradient boosting learns, and the depth of the tree, could be optimally selected through validation techniques like k-fold cross validation. Having a large number of trees might lead to overfitting. So, it is necessary to carefully choose the stopping criteria for boosting.

The boosting ensemble technique consists of three simple steps:

* An initial model F0 is defined to predict the target variable y. This model will be associated with a residual (y – F0)
* A new model h1 is fit to the residuals from the previous step
* Now, F0 and h1 are combined to give F1, the boosted version of F0. The mean squared error from F1 will be lower than that from F0: 𝐹1(𝑥) = 𝐹0(𝑥) + ℎ1(𝑥)

To improve the performance of F1, we could model after the residuals of F1 and create a new model F2:

𝐹2(𝑥) = 𝐹1(𝑥) + ℎ2(𝑥)

This can be done for ‘m’ iterations, until residuals have been minimized as much as possible: 𝐹𝑚(𝑥) =

𝐹𝑚−1(𝑥) + ℎ𝑚(𝑥)

Here, the additive learners do not disturb the functions created in the previous steps. Instead, they impart information of their own to bring down the errors.

# Logistic Regression

It’s a classification algorithm, that is used where the response variable is categorical. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome. The method is relatively robust, flexible and easily used, and it lends itself to a meaningful interpretation. In LR, no assumptions are made regarding the distribution of the explanatory variables.

The LR model can be expressed as:

𝑝

log (1 − 𝑝) = 𝛽0 + 𝛽1𝑥

where,

( ) is the logit (log-odds) function

* 𝑝

1−𝑝

* 𝛽0 is the intercept
* 𝛽1 is the slope
* & 𝑥 is the regressor

Since this is a multiclass dataset, we apply one v/s rest approach to apply binary logistic regression.

# One versus Rest:

One-vs-rest (OvR for short, also referred to as One-vs-All or OvA) is a heuristic method for using binary classification algorithms for multi-class classification.

It involves splitting the multi-class dataset into multiple binary classification problems. A binary classifier is then trained on each binary classification problem and predictions are made using the model that is the most confident.

# Association Rule mining: Apriori Algorithm

Apriori Algorithm is one of the algorithms used for transaction data in Association Rule Learning. It allows us to mine the frequent itemset in order to generate association rule between them.

Principles behind Apriori Algorithm

* Subset of frequent item sets are frequent itemset.
* Superset of infrequent item sets are infrequent itemset.

Apriori Algorithm has three parts:

For a rule X ⇒ Y

* 1. Support = Frequency(X,Y)

N

* 1. Confidence = Frequency(X,Y)

Frequency(X)

* 1. Lift = Support

Support(X)×Support(Y)

# Algorithm of Apriori Algorithm:

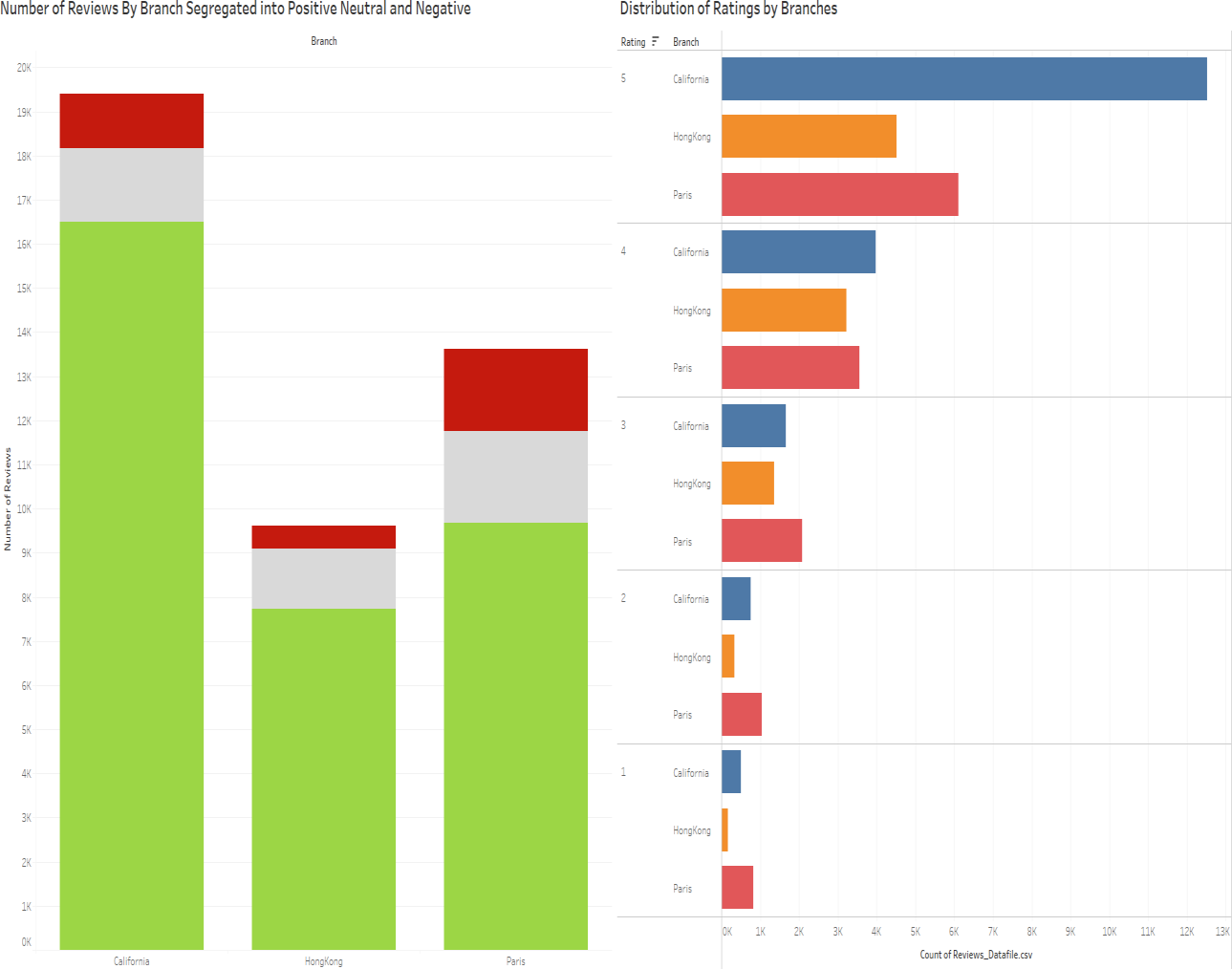
**Step 1**: Set a minimum support and confidence.

**Step 2**: Take all the subset present in the transactions which have higher support than minimum support.

**Step 3**: Take all the rules of these subsets which have higher confidence than minimum confidence.

**Step 4**: Sort the rules by decreasing lift.

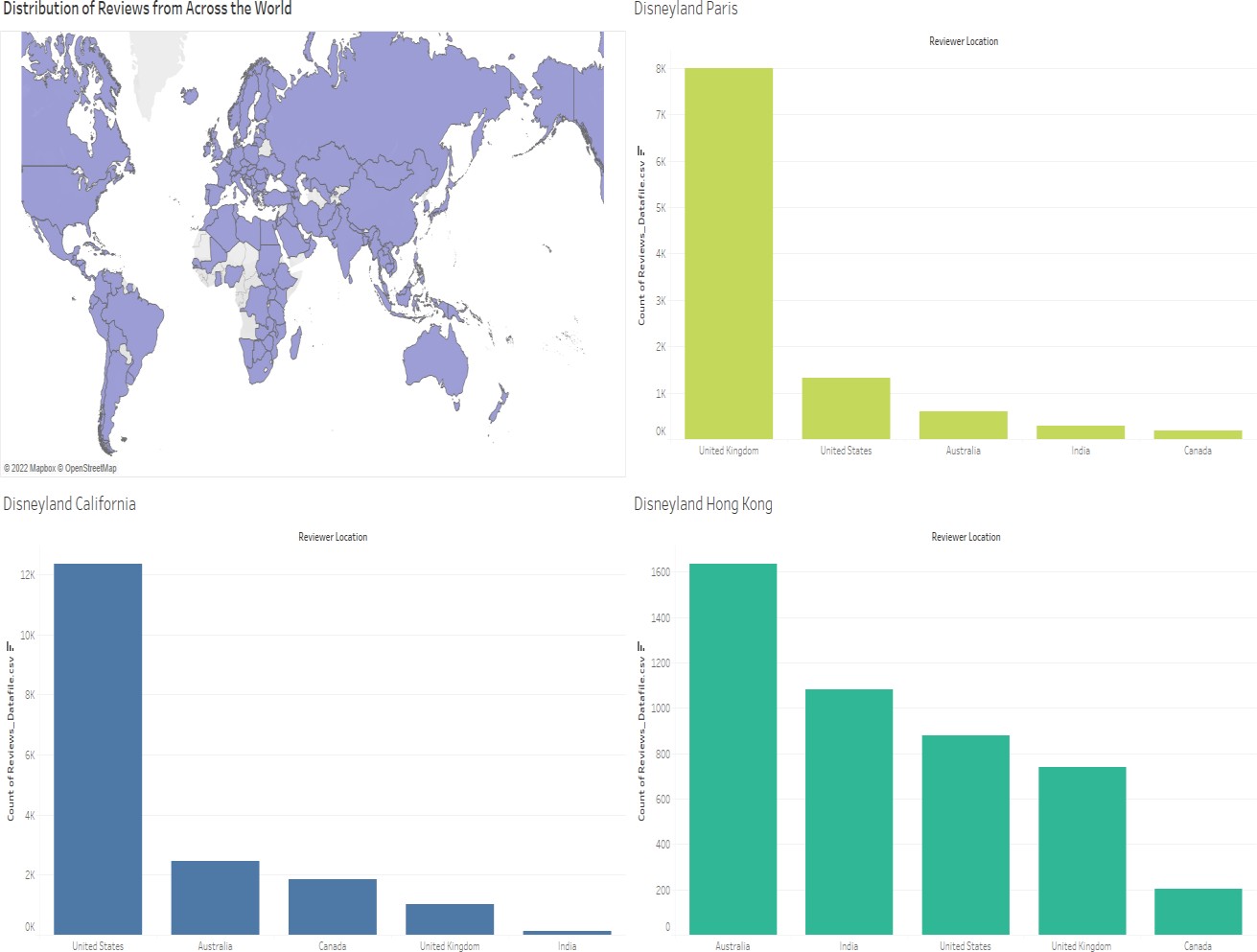
# EXPLORATORY DATA ANALYSIS



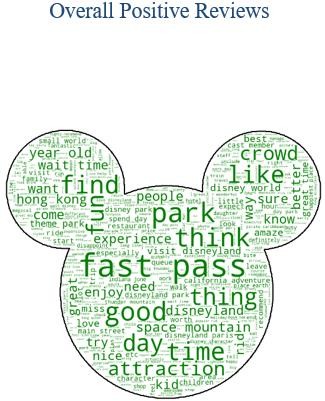
Most of the reviewers in the dataset have reviewed Disneyland California and even though the number of reviews for Disneyland California are the highest, we observe that the number of negative reviews (Ratings: 1 and 2) are higher for Disneyland Paris.



The highest number of reviews are from the years of 2014-2016. In 2015, Disneyland California celebrated its 60th anniversary from July 2015 till September 2016 which might be the reason for the higher number of visits in California and hence the higher number of reviews from that branch, during that period. From the month wise distribution of negative reviews, we observe that for most of the years the negative reviews are for the months of July to September. This might be because of summer vacations for these countries happen during that period, which results in more visits and hence more negative reviews.



People have visited Disneyland parks from all around the world the top 5 reviewers’ location for Disneyland Paris are from UK, USA, Australia, India and Canada. For Disneyland California, most of the reviewers are from USA, customers from UK, Australia, Canada and India are also frequent reviewers for Disneyland California. For Disneyland Hong Kong, the most frequent reviewers are from Australia and India due to the close vicinity of these countries, people from USA are 3rd highest reviewers (hence visitors) even though they have two of the biggest Disneylands in California and Florida.







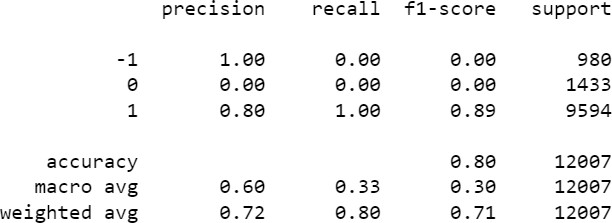
In general, the reviews from all three locations mention fast passes, lines or queues, visitor demographic, rides and attractions.

The word cloud of California has words which are more common with the word cloud of all the positive reviews.

While, word cloud of Paris has words which are common with the word cloud of all the negative reviews.

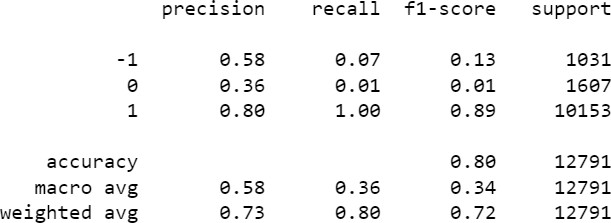
# OUTPUT

**Naïve Bayes**



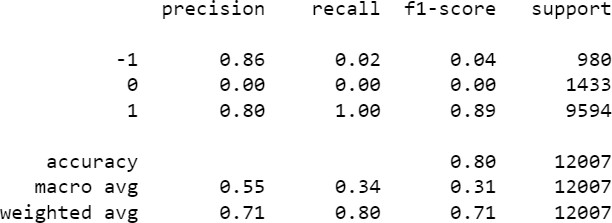
In this model we have good precision scores for -1 and 1, but when we look at the f1-score(s) we see that the f1-score for negative and neutral reviews is 0.00, as the recall for both cases is 0.00. Which means it does not properly classify actual negative, which is more important for us from a business standpoint.

# Complement Naïve Bayes



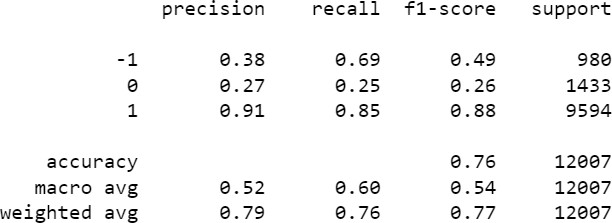
In this model we see that the f1-score(s) have improved compared to Naïve Bayes but they are still very low.

# Random Forest (before hyperparameter tuning):



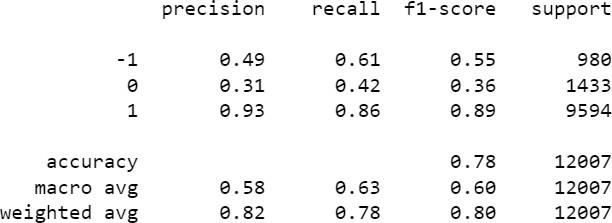
In this model, we see that this model is a step backwards on the CNB model. Recall scores and by their effect the f1-score(s) have worsened quite a bit. We can apply hyperparameter tuning in order to improve these scores.

# Random Forest (after hyperparameter tuning):



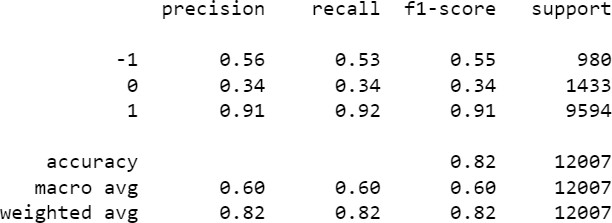
After applying hyperparameter tuning, we see major improvements in the Random Forest model. Here we can observe that the weighted average f1-score has increased to 0.77 as compared to 0.71 (in Random Forest before hyperparameter tuning) and 0.72 (in the CNB Model).

# XG Boost (before hyperparameter tuning)



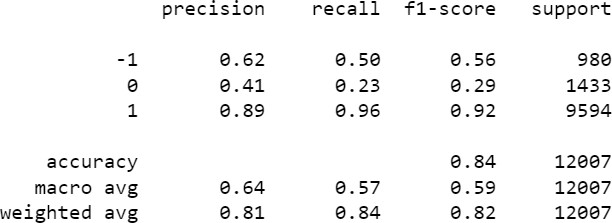
In this model we can see that the f1-score for negative reviews has increased even further compared to Random Forest, which means that XG Boost is doing a better job of classifying negative reviews as compared to Random Forest. We will apply hyperparameter tuning in order to increase these scores even further.

# XG Boost (after hyperparameter tuning)



After applying hyperparameter tuning on the XG Boost model, we see marginal improvements on accuracy and weighted average scores

# Logistic Regression



Logistic Regression model works just as well as XG Boost after hyperparameter tuning, but uses considerably less computational power.

# Apriori Algorithm

For the association rules, we keep the minimum support at 0.05 and the minimum lift at 1.2 and obtain the following results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item 1** | **Item 2** | **Support** | **Confidence** | **Lift** | **Frequency** |
| United Kingdom | Disneyland Paris | 0.274258 | 0.875855 | 1.710352 | 896 |
| United States | Disneyland California | 0.230181 | 0.757301 | 2.179826 | 752 |
| August | Disneyland Paris | 0.130701 | 0.648712 | 1.266791 | 277 |

Applying the Apriori algorithm we see that the itemset {United Kingdom, Disneyland Paris} has a support of 0.2742 indicating that 27.42% of the negative reviews come from this itemset. Another 13.07 % of the negative reviews involve the itemset {August, Disneyland Paris} and another 23.01% of the negative reviews involve the itemset {United States, Disneyland California}.

From the confidence values we see that the conditional probability of occurrence of Disneyland Paris given that the reviewer is from the United Kingdom is 0.8758 and we also see that while looking at the negative reviews that the reviewer is **1.71 times more likely** to give a negative review about Disneyland Paris given that he is from the United Kingdom.

We also see that the conditional probability of occurrence of Disneyland California given that the reviewer is from the United States is 0.757 and we also see that while looking at the negative reviews that the reviewer is **2.17 times more likely** to give a negative review about Disneyland Paris given that he is from the United States.

We also see that the conditional probability of occurrence of Disneyland Paris given that the review is made in the month of August is 0.648 and we also see that while looking at the negative reviews that the reviewer is **1.266 times more likely** to give a negative review about Disneyland Paris given that the review is made in the month of August.

We ask the business to identify the possible specific reasons for the high amount of negative reviews from the groups namely {United Kingdom, Disneyland Paris}, {August, Disneyland Paris} and {United States, Disneyland California}. By going through a few reviews we observe that the keywords like “Queue”, “Waiting”, “Expensive” occur frequently which were a few of the common words observed in the word clouds.

The most frequently occurring problems include:

* Long Waiting time at each ride causing frustration among the visitors.
* Overpricing
* Overcrowding during peak times
* Long waiting times for washrooms
* Non Co-operative Staff

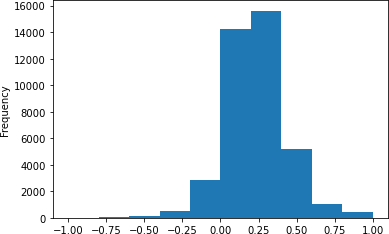
# BUSINESS INSIGHTS

**Methodology**

POLARITY AND SUBJECTIVITY:

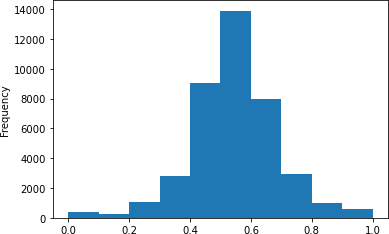
Using the textblob module, we calculate the polarity and subjectivity for the reviews.

By plotting the histogram of the polarity, we notice that most polarity scores are distributed between moderate magnitudes of polarity. However, we are more interested in the higher magnitudes of polarity, which signifies a stronger sentiment.



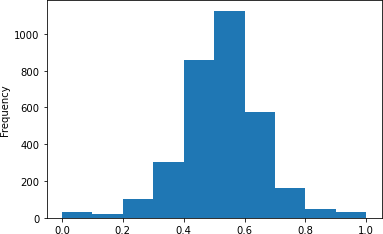
Similarly, we plot the histogram for Subjectivity scores for reviews. From the histogram, we notice that the distribution of subjectivity scores is centered at 0.5. This implies that the reviews in our dataset are both subjective and objective in nature. Objective reviews are the ones that do not emit any sentiment or feeling but are factual in nature. Subjective reviews are the personal opinions and are useful for analysing the sentiments.

We consider the values above 0.4 to be subjective enough for our analysis



ANALYSIS FOR BUSINESS INSIGHTS

Let us now look at the distribution of negative reviews based on subjectivity.



We observe that a large number of negative reviews have a subjectivity score of more than 0.4. So, it is easy to conclude that the negative reviews are more of personal opinions based on their experience rather than some factual statements.

Now, looking from a business perspective, the reviews with a more negative polarity score are more important to identify the pain points of the visitors.

So, we consider the reviews that has polarity scores of less than (-0.6). From such reviews, we pick out the keywords to determine the major issues mentioned by the visitors for the three branches of Disneyland.

Some of the major keywords are: overprice, queue, wait, disability, filthy, stale, malfunction.

Now, we find the ‘most similar’ terms to these words from the context of the reviews by fitting Word2Vec model and using the ‘most\_similar()’ function of word2vec. Using this function, we get different words that either have same meaning to the keywords mentioned above or are associated based on the context in the reviews.

The word ‘stale’ has been associated with words like ‘yoghurts’, ‘buns’, ‘stew’. So, this could mean that the food they might have been served was stale.

The word ‘filthy’ has been associated with words like ‘bathrooms’, ‘dirty’, ‘litter’, ‘urinal’. This could imply that the reviewer must have faced issues related to littering or dirty bathrooms.

Similarly, for all the other keywords, either synonyms or other associated words were found using Word2vec.

Now, using Count Vectorizer, we find the frequency with which the keywords along with their associated similar words have occurred in the reviews. This would help us determine which of the keyword has the major pain point associated with it.

We determine the frequencies of the keywords for all the three branches and determine what leads to more negative reviews related to the particular branch.

# Proposed Business Solutions:

Since Paris has the highest proportion of negative reviews among the three branches, and by further analysis on the keywords in negative reviews with high polarity, we propose:

1. Better queue management system at all the branches of Disneyland.
2. Improvement in facilities and assistance provided to specially-abled visitors.
3. Reconsider the allocation of prices to tickets, different rides etc.
4. Encourage the visitors to keep the premises and toilets clean.

# CONCLUSIONS

* + We find that Disneyland Paris is not functioning as well as it’s California and Hong Kong counterpart.
  + Best performing models were XG Boost and Logistic Regression. They both performed equally well, but Logistic Regression is computationally cheaper.
  + Taking proper steps and systematically improving the sectors that we identified as problematic, will result in positive visitor feedback and increased attendance in the upcoming years.

# LIMITATIONS/ CHALLENGES

* + There are about 2 lac reviews available online for various Disneyland branches but since we used a dataset of scraped reviews from Kaggle, we were able to work with 42,632 reviews only.
  + We did not have access to visitor reviews for competitors of Disney parks such as Universal Studios and Legoland.
  + The knowledge of visitor sentiments for these parks could have helped in providing better suggestions regarding pricing and customer service at Disney parks.

# FUTURE SCOPE

* + The scope of this project can be widened further and be carried out by using Deep Learning models, which we were not able to implement given the time constraints.
  + Advanced tools like Topic modelling can be used to get better business insights.
  + The developed Machine Learning model in the project can be deployed and further be used by Disneyland to classify the continuously incoming reviews and use the relevant insights to improve the visitor experience.

# BIBLIOGRAPHY

1. Dey, L.; Chakraborty, S.; Biswas, A.; Bose, B.; Tiwari, S. (2016) “Sentiment analysis of review datasets using naive bayes and k-nn classiﬁer.”
2. M. Kanakaraj and R. M. R. Guddeti, "Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques."
3. Chong, W.Y.; Selvaretnam, B; Soon, L (2014), “Natural Language Processing for Sentiment Analysis”