

Team SegFault

Data Visualization and Insight gathering

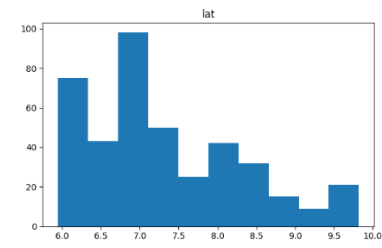
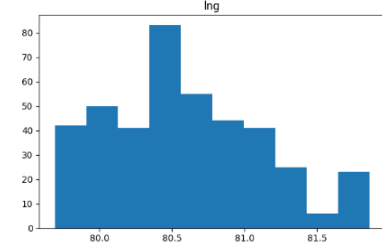
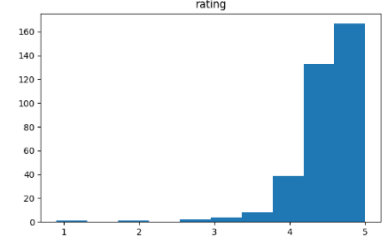
Locations dataframe

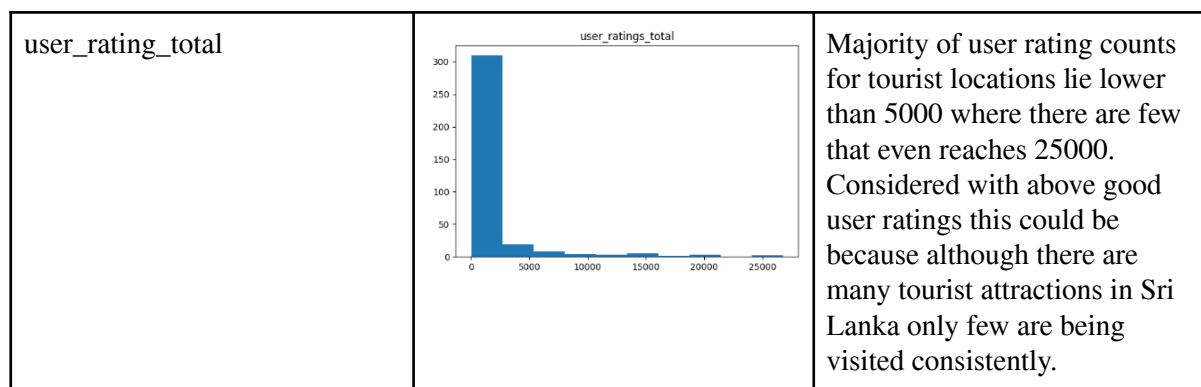
Searching for null valued features for future preprocessing

We checked each feature for null value percentage to scope the future preprocessing plan which is stated under Data Preprocessing. Following were the percentages that we found.

name	0.000000
lat	0.243309
lng	0.243309
formatted_address	0.000000
rating	13.625304
user_ratings_total	13.625304
latest_reviews	0.000000

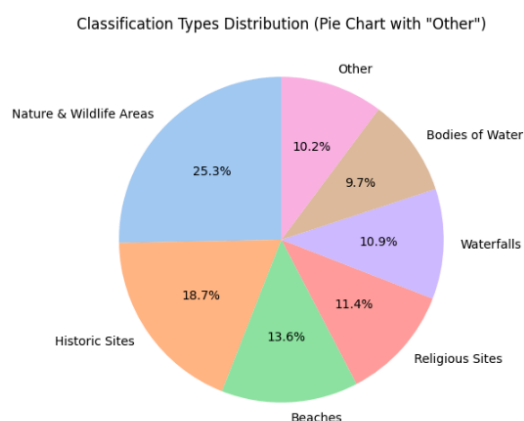
Histograms for numerical features

Feature	Histogram	Observations and Insights
lat (Latitude)		1. Majority of the tourist locations are centralized in the mid section of the island. This could be due to the diverse types of tourist locations and experiences can be found in this area of the country.
lng (Longitude)		2. Very small amount of tourist attractions in the northern part of the country. This could be due to the dry climate and the small surface area of the northern part of the country
rating		Majority of tourist locations have received user reviews that are higher than 4. This could be because their are diverse tourist experiences found in Sri Lanka.



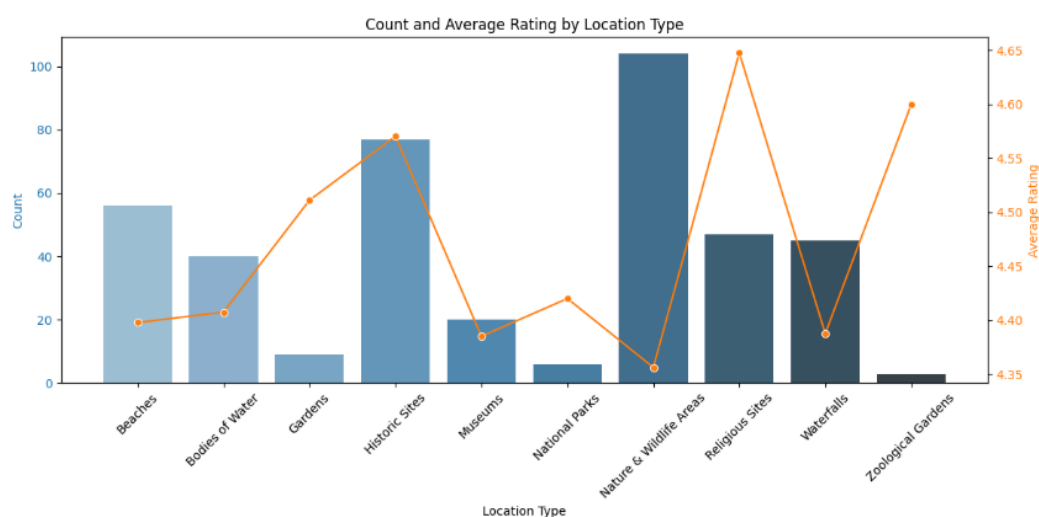
Percentages of tourist attraction types

With the help of a reference data set which had the tourist attractions classified (methodology mentioned below) we classified the tourist locations in the data set given to us to get an idea on the tourist location type percentages in the data set.



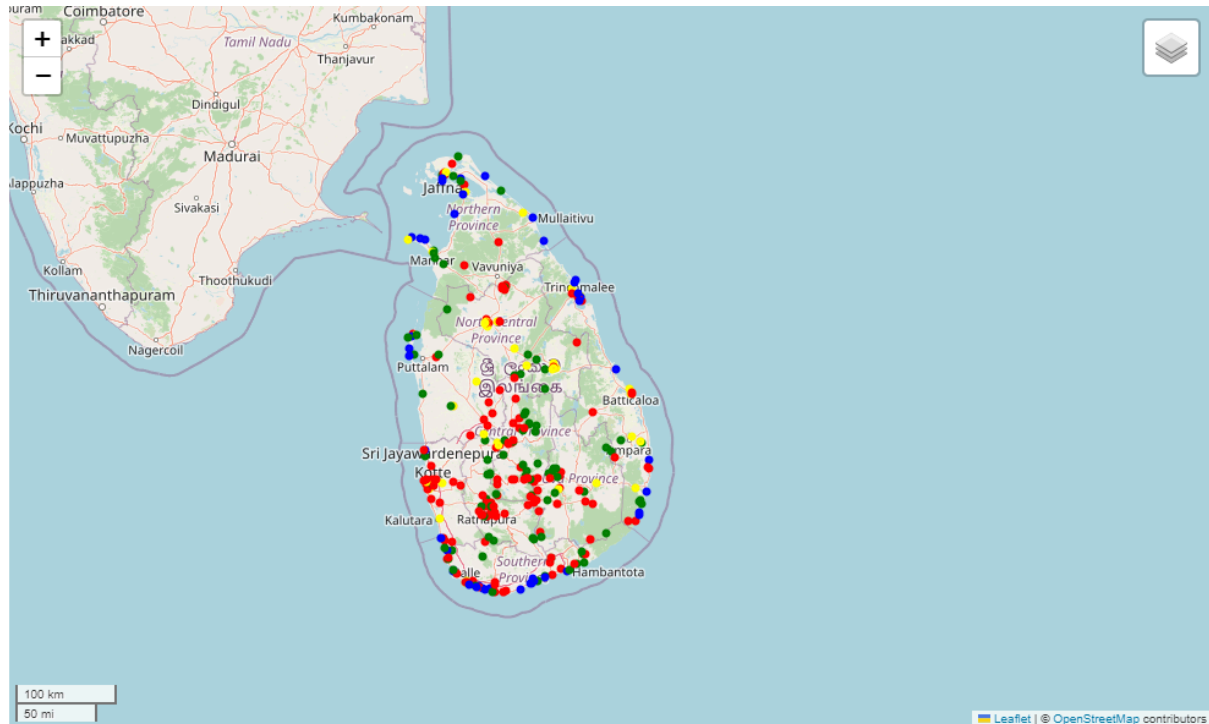
Classified tourist attractions and their average rating

Averaging the location classes rating values can provide misleading ideas due to the count of each location class and their respective rating counts. We graphed the following bar graph and the line graph to get an idea on the ratings of each location along with their location counts.



Viewing datapoints (attraction locations) in a map

By using python folium library we visualized all the given attraction locations in a map to get an idea on potential preprocessing that should be done in the longitude and latitude values in the data set. But all the data points were annotated within the island and did not seem to have an necessity of preprocessing the longitude and latitude values



Beaches: Blue

Nature and Wild life locations: Green

Historical sites: Yellow

Others: Red

Visitors Dataframe

Visualizing top activities that people like

68 unique activities that people like to do were found in the activity list of dataset. Following are the top 10 activities that people like to do and their counts

cultural festivals	496
yoga retreats	479
cultural experiences	479
river cruises	473
historic walks	473
amusement parks	472
water parks	470
outdoor adventures	469

scuba diving	469
hot air ballooning	467

Visualizing top places that people have in their bucket list

158 unique places were found that people have added to their bucket lists. Following are the top 10 locations that people have in their bucket list and their counts.

Sigiriya	1325
Kandy	1166
Trincomalee	1153
Mirissa Beach	1091
Ella	1049
Yala National Park	872
Anuradhapura	837
Hikkaduwa	827
Pigeon Island	802
Bentota River	802

Visualizing inconsistencies in naming places in Location data-set and Visitor data-set(bucket list)

54 locations in bucket list are not there in the due to not being in the original dataset and naming differences.

```
[ 'Batatotalena (Batadombalena) Cave' 'Galle Fort' 'Passikuda Beach'
  'Polonaruwa' 'Ella' 'Bentota River' 'Water World Lanka' 'Knuckles'
  'Belilena Caves' 'Negambo' 'Ambalangoda Mask Workshop' 'Bentota'
  'Unawatuna Lagoon' 'Horton Plains' 'Nallur Kandaswamy Kovil'
  'Museum of Modern and Contemporary Art' 'Victoria Golf Club'
  'Kitugala Forest' 'Ella Rock' 'Colombo Port City'
  'Hikkaduwa Coral Sanctuary' 'Mahalenama Cave' 'Excel World'
  'Galle City Tour' 'Ritigala' 'Wavulpone Cave' 'Riverstone Gap'
  'Laxapana Falls' 'Anawilundawa Wetlands' 'Mahapelessa Hot Springs'
  'Madu River' 'Udawalawe' 'Hiriketiya' 'Jungle beach' 'Hatton'
  'Kanniya Hot Springs' 'Weligama Beach' 'Ambuluwawa Tower' 'Kandy Temple'
  'Batadombalena Craft Centre' 'Ella Gap' 'Kithulgala' 'St Clair's Falls'
  'Vaddha Village Camping' 'Anuradapura' 'Ratnapura Gem Museum'
  'Baker's Falls' 'Folk Museum' 'Royal Botanical Gardens, Peradeniya'
  'Perl Bay' 'Arankelle Forest Monastery' 'Kosgoda Turtle Hatchery'
  'Bentota Beach' ]
```

Data Preprocessing

Locations dataframe

Cleaning the values that were added due to encoding inconsistencies.

Due to the inconsistencies of encodings of the dataset, many data points had terms such as 'ÃfÆ'Ã,Â'. By analyzing the data points it was clear to us that most of these Jargon had patterns.

ÃfÃçÃ,â,¬Ã,â,,ç was one of them which was a result of encoding the apostrophe. We replaced these values and removed the rest of these characters with blanks.

Extracting location names from the user reviews using GenAI

The provided data set contained several location names that were unclear due to encoding errors and data entry errors. To impute these location names we used the Gemini generative AI model through **google.generativeai** library and extracted location names from the user reviews. Through this we were able to clarify uncertain location names of the data set.

Dealing with Null values

In the given dataset null values were only found in features, rating and user_ratings_total with only one exception. Data point referred to leisure world had Longitude and Latitude as null values which we imputed using the realistic values found through google maps. For preprocessing the datapoints with null ratings we separated the null rated and rated datapoints into two dataframes and preprocessed separately. How ratings were imputed in null-rated data frame is given below.

Removing Redundant Locations

Finding redundant data points

Due to the multiple ways that data location names could be represented, we analyzed the location names for similarities in their values. After finding similar entries we analyzed these repeated terms in location names to find redundant data points.

Removing redundant data points

In data points that were repeated we kept the feature values of the mostly reviewed datapoint and removed other datapoints. Before removing these repeating data points we added the review counts and replaced the review value with the averaged value of all the repeating data point's review values.

Generating Reviews Based on Recent User Feedback

During our analysis, we noticed that some places lacked user reviews or had invalid ones, where ratings fell outside the valid range of 1 to 5. To address this issue, we took a simple yet effective approach.

We fine-tuned an [SBERT](#) model to extract the sentiment from each review, calculating an average sentiment score for each location. We generated representative, synthetic review scores for places without valid reviews by scaling the sentiment scores within the standard 0 to 5 rating range. These generated scores filled in the gaps, ensuring every location had a reliable review score for more accurate insights.

Reference dataset preprocessing

To get an idea about datasets in the domain and to improve the provided dataset we used a reference data-set on Sri Lanka's tourism locations. We had to preprocess this dataset to make it understandable as it contained singular user reviews as singular datapoints.

Grouping similar location data points: We grouped the datapoints relevant to the same location and averaged the user reviews.

Visitors Dataframe

Separating the activity feature and the bucket list feature

Originally in the dataset activity *Bucket list destinations Sri Lanka* and the *Preferred Activities* features had their multiple values as lists in 2 features. We separated these lists as 5 location features for the bucket list location and 3 activity features for the preferred activity feature.

Dealing with inconsistencies in location naming in bucket list with the location data set

We extracted the inconsistent location names with the relevant location names given in the preprocessed location data set using Levenshtein library (able to find two most similar elements between 2 lists of strings). But this method causes misconceptions of some location names as shown in the Preprocessing_Visitor_Dataset notebook. This issue is addressed withing our model description.

Our Solution

To provide personalized travel recommendations, we developed a hybrid approach that combines **content-based filtering** and **collaborative filtering** to match user preferences with the best travel destinations. Here's how our solution works:

1. **Content-Based Filtering on User Preferences and Reviews**

We analyze user preferences and location reviews by utilizing a fine-tuned sentence transformer model. This allows us to compute text embeddings and measure similarities between user preferences and location reviews using cosine similarity. The places with the highest similarity scores are recommended to the user. The recommended places are taken out to the next stage of the pipeline

2. **Collaborative Filtering Based on User Preferences**

By analyzing user reviews, we group users with similar preferences and suggest locations from the bucket lists of similar users. This method leverages the assumption that users with similar tastes tend to choose similar destinations.

3. **Collaborative Filtering Based on Bucket List Items**

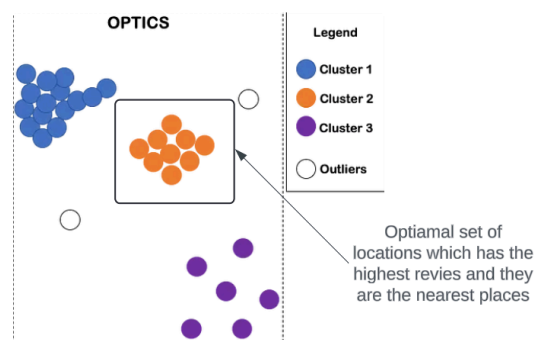
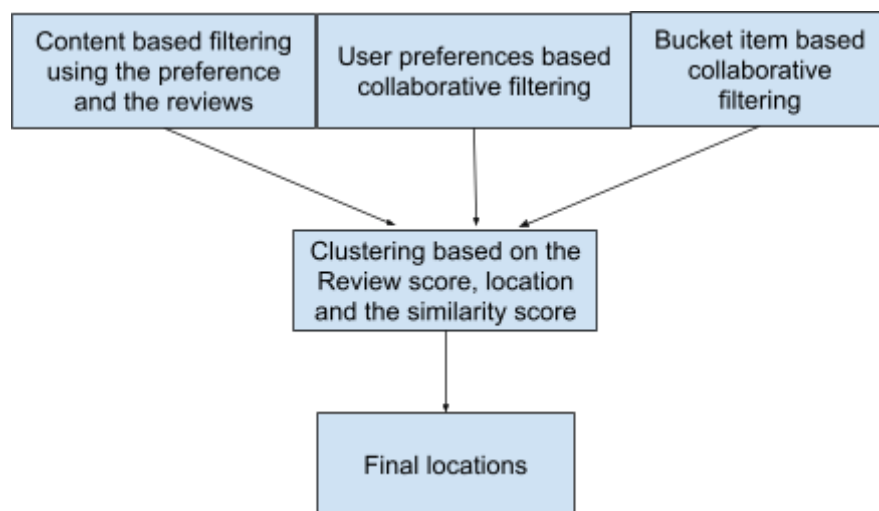
We also compared location review embeddings across different users' bucket lists and taken the similarity between the bucket list items. By finding similar locations between bucket lists, we can recommend new travel destinations based on overlapping interests.

We have taken the suggestions of these stages in to the next satage

Clustering-Based Filtering for Final Destination Recommendations

To provide a seamless travel experience, it's essential that the recommended destinations not only align with the user's preferences but are also well-reputed and located near each other. Suggesting distant locations from opposite ends of Sri Lanka would be impractical.

We propose a final clustering-based filtering layer to optimize the destination selection. By utilizing the [OPTICS clustering algorithm](#), we ensure that the user is presented with the nearest five locations that have the highest percentage of positive reviews. This approach guarantees that users receive tailored recommendations that meet their preferences and provide an efficient travel route.



This algorithm is robust for the outliers as well so this was the right choice for our implementation of the recommendation engine.

Evaluation Metrics That we have used for the Model.

To enhance our model's accuracy, we evaluated its performance using users' bucket list details. While the bucket list plays a vital role in predicting their preferred locations, we propose a method to further validate the model's effectiveness by analyzing the influence of these bucket list items.

Recall @K

Recall@K gives a measure of how many of the relevant items are present in top K out of all the relevant items, where K is the number of recommendations generated for a user. For example, In our case we are building a travel place recommender system where we recommend locations for every user. If a user has selected 5 places, and our recommendation list has 3 of them (out of the 10 recommendations), the Recall@10 for a user is calculated as $3/5 = 0.6$. We have taken the average across all users for evaluation.

$$\text{Recall@}K = \frac{\text{Number of Relevant Items in Top } K}{\text{Total Number of Relevant Items}}$$

Precision @K

Precision@K gives a measure of “out of K” items recommended to a user and how many are relevant, where K is the number of recommendations generated for a user..

For a recommendation system where we recommend 10 Travel destinations for every user. If a user has selected 5 places and we can predict 3 out of them (3 movies are present in our recommendation list) then our Precision@10 is 3/10.

$$\text{Precision@}K = \frac{\text{Number of Relevant Items in Top } K}{K}$$

F1 @K

$$\text{F1@}K = \frac{2 \times \text{Precision@}K \times \text{Recall@}K}{\text{Precision@}K + \text{Recall@}K}$$

F1 Score is a combination of Precision and Recall using harmonic mean. This is the same as the regular F1 Score and does not differ in the context of the recommendation systems. The harmonic mean nature makes sure if either Precision or Recall has a really high value, then it

does not dominate the score. F1 Score has a high value when both precision and recall values are close to 1.

Custom Similarity Metric using the vector Embeddings

In this method what we did was that we have used the review embeddings from the locations of the users bucket list and we have taken the review embeddings of the predicted locations and we have measured the cosine similarity between the locations

The Reasons behind selecting this Evaluation metrics

1. **Precision@K:**

- For a travel recommendation system, it's important to ensure that the destinations recommended are highly relevant to the user's preferences (e.g., bucket list items). Precision@K helps determine the accuracy of the model by evaluating if the top K suggestions align with what the user would likely choose, directly assessing recommendation quality.

2. **Recall@K:**

- Captures how well the model retrieves all relevant items from the user's preferences within the top K recommendations. In a travel recommendation context, recall is crucial to ensure the model isn't missing out on key locations that match the user's desires. Recall@K focuses on the model's ability to present a comprehensive set of relevant suggestions, which enhances user satisfaction by covering a wide range of desired destinations.

3. **F1@K:**

- Combines precision and recall into a balanced score, indicating how well the model balances between not only returning relevant recommendations (precision) but also capturing all relevant suggestions (recall). Travel preferences often require both precision (to show high-quality destinations) and recall (to avoid overlooking important locations). F1@K provides a harmonic mean of both metrics, offering a single metric to gauge the overall effectiveness of the recommendation model.

Together, these metrics ensure the travel recommendation system effectively presents both relevant and comprehensive destination suggestions, aligning with the goal of personalizing user experiences based on their preferences.

Results From our Model

```
User ID: 9991
Activity List: ['botanical gardens', 'elephant rides', 'cultural festivals']
Recommended places for the new user based on activity: {'Kandy', 'Nallur Kandaswamy Devasthanam'}
All places recommended: ['Kandy', 'Nallur Kandaswamy Devasthanam']
Bucket Listed Places: ['Pinnawala', 'Hakgala Botanical Garden', 'Udawalawe', 'Seethawaka Wet Zone Botanical Gardens', 'Dry Zone Botanic Gardens, Hambantota']
Recommended items for the new user based on bucket_list: set()
['Seethawaka Wet Zone Botanical Gardens', 'Hakgala Botanical Garden', 'Udawalawe', 'Pinnawala', 'Dry Zone Botanic Gardens, Hambantota']
```

We evaluated our model using **precision@5** for users already present in the dataset, achieving an impressive score of **80%**, which is a strong result for precision at k. Additionally, we employed a cosine similarity measure for our predictions, which yielded a high similarity score of **89%**. The measure for the recall@5 was 85% which give the F1@5 of **0.84**

We can say that our model performed exceptionally well in this scenario for several reasons:

1. High Precision@5 (80%):

- Achieving 80% precision means that out of the top 5 recommendations, 4 out of 5 were highly relevant to the user's preferences. This indicates that our model is effectively predicting destinations that closely match user preferences, particularly for users already present in the dataset.
- Reason: This high precision suggests that the model successfully identifies and ranks relevant travel destinations at the top, ensuring users are presented with destinations they are likely to prefer, which is critical in personalized recommendation systems.

2. Cosine Similarity (89%):

- The cosine similarity measure indicates how closely related a user's preferences are to the recommended destinations. A score of 89% shows that our model is effectively identifying destinations that are very similar to the user's historical preferences.
- Reason: By using cosine similarity, the model can capture the nuanced relationships between user preferences and travel destinations, leading to more personalized and accurate recommendations.

3. High Recall@5 (85%):

- Recall@5 of 85% shows that the model is retrieving most of the relevant destinations within the top 5 recommendations. This ensures that users are not missing out on important travel suggestions that align with their preferences.
- Reason: High recall means the model is comprehensive, ensuring that a wide range of relevant options is presented, enhancing the overall recommendation quality and user satisfaction.

4. F1@5 Score of 0.84:

- The F1@5 score of 0.84 balances both precision and recall, indicating that the model is well-rounded in presenting accurate and comprehensive recommendations.
- Reason: A strong F1 score shows that our model not only excels in making highly relevant predictions but also ensures that it captures most of the relevant destinations, providing an optimal balance between precision and recall.

References

- Reimers, Nils, and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks." *ArXiv*, 2019, /abs/1908.10084. Accessed 13 Sept. 2024.
- Sewwandi, Taniya (2023), "Tourism and Travel Reviews: Sri Lankan Destinations", Mendeley Data, V1, doi: 10.17632/2nbvx5m4hs.1 Accessed 12 Sept 2024
- Ankerst, M., Breunig, M. M., Kriegel, H., & Sander, J. (1999). OPTICS. *ACM SIGMOD Record*, 28(2), 49–60. <https://doi.org/10.1145/304181.304187>