EBA5004: Graduate Certificate in Practical Language Processing

Topic: Singapore Advisor - Sentiment Mining on Singapore Attractions Reviews

Team Name/ Number: Group 14

2023-5-21

Group 14

Namrata Thakur

Ouyang Hui

Yashna Gogineni

Fenglei

Content

1. Introduction

The tourism and amusement industry is a significant contributor to Singapore's economy. To maintain its competitiveness, it is essential to understand customer sentiment and preferences regarding attractions. This project aims to perform sentiment mining on Singapore tourist attractions reviews on TripAdvisor, providing insights into customer experiences and preferences.

Through sentiment mining, we can uncover not only the overall sentiment towards specific attractions but also gain a deeper understanding of the factors that contribute to customer satisfaction or dissatisfaction. By examining sentiments at a granular level, such as specific aspects of attractions, amenities, customer service, or cultural experiences, we can identify areas of strength and opportunities for improvement.

For the purpose of this project, we have chosen to examine 20 diverse locations. These venues have been meticulously selected to represent a broad range of categories, ensuring inclusivity and diversity in our analysis. In addition, we incorporated popular places based on their high review numbers. This approach allows us to provide comprehensive insights that encompass both well-known and lesser-known attractions, giving a holistic view of the Singapore amusement landscape.

2. Business Problem

The target business domain we are focusing on is the tourism industry of Singapore. Every year millions of tourists come to Singapore. With the Covid restrictions being relaxed, the international visitors to Singapore are expected to hit 12 million-14 million in 2023. The business value of this project lies in assisting amusement service providers to improve their offerings, identify areas for improvement, and monitor changes in sentiment over time.

3. Related work

Aspect-Based Sentiment Analysis (ABSA) is a specific type of Sentiment Analysis that aims to extract the most important aspects of an entity and predict the polarity of each aspect from the text. ABSA has been used in various domains such as e-commerce, social media, and customer reviews. There are three mainstream methods for ABSA: lexicon-based, traditional machine learning, and deep learning methods (Liu et al., 2020). The survey by Zhang et al. (2022) provides an overview of ABSA tasks, techniques, datasets, and evaluation metrics. The authors discuss the challenges of ABSA such as aspect extraction, aspect sentiment classification, and aspect-based summarization. They also provide a detailed comparison of various ABSA methods such as rule-based methods, machine learning-based methods, and deep learning-based methods. The authors conclude that deep learning-based methods have achieved state-of-the-art performance in ABSA tasks.

4. Dataset

4.1. Data Collection

The dataset used in this project was obtained through web scraping from the Things to Do section of the Tripadvisor website (THE 15 BEST Things to Do in Singapore - 2023 (with Photos), n.d.). For data collection, BeautifulSoup was employed to parse the HTML structure and extract the required data from web pages, while Selenium was used for automating the web scraping process.

A total of 20 diverse and popular places were analysed in this project: Arab Street, Buddha Tooth Relic Temple and Museum, Clarke Quay, Gardens by the Bay, Jurong Bird Park, ArtScience Museum at Marina Bay Sands, Maxwell Food Centre, Merlion Park, Mustafa Centre, National Museum of Singapore, National Orchid Garden, Night Safari, Orchard Road, River Wonders, Sands Skypark Observation Deck, Singapore Flyer, Singapore River, Singapore Zoo, Singapore Botanic Gardens, and Singapore Mass Rapid Transit (SMRT). These carefully selected attractions offer a comprehensive representation of Singapore's rich and varied amusement scene, ensuring a diverse and insightful analysis.

The dataset utilized in this project consists of the following attributes:

- Place: The name of the attraction that was reviewed.
- Reviewer: The name of the user who provided the review.
- Reviewer_location: The origin or location of the reviewer.
- Reviewer_contribution: The reviewer's activity and contributions on the Tripadvisor platform.
- Review_rating: The rating given to the attraction on a scale of 1 to 5, with each number corresponding to terrible, bad, average, good, and excellent respectively.
- Review_type: The type of visit, such as family, friends, or couples.
- Review_date: The date when the review was posted.
- Review_title: The title or summary of the review as provided by Tripadvisor.
- Review_text: The plain text content of the review itself.

place	reviewer	reviewer_location	reviewer_contributions	review_rating	review_type	review_date	review_title	review_text
Arab Street	bob2bkk	Bangkok, Thailand	5907	3		March 18, 2023	Sultan Mosque and quaint sho	Arab Street is a colorful and interesting neighborhood area to take a stroll. I
Arab Street	TassieTravellers99	Launceston, Australia	3172	3		March 6, 2023	Taste of the middle east	Located near Bugis is the Arab Quarter with its brightly painted buildings and
Arab Street	paulj	United Kingdom	99	5		March 5, 2023	Lovely place to spend a couple	A great place to visit, so many restaurants, and fabulous textile shops plenty
Arab Street	Lakehouse15	Byron Bay, Australia	138	4		January 11, 2023	Good for a walk	A very colourful place to walk through with the aroma of spices and coffee su
Arab Street	DuncanTCH	East Greenbush, NY	1183	4		November 9, 2022	Beautiful street that reminds	The Arab neighborhood has plenty of activity and interesting things go on aro
Arab Street	Sarah L	London, UK	2835	4		November 6, 2022	Good for an amble	A popular street for tourists filled with many light and lantern shops, carpet s
Arab Street	Matt S	Adelaide, Australia	22	1	Couples	October 26, 2022	Avoid, avoid, avoid!	Avoid Arab street if you (or someone in your party) is a wheelchair user. It is
Arab Street	Barry C	Perth, Australia	82	4		October 26, 2022	A taste of the middle east in S	Capadocia Turkish restaurant was great and offered fine, freshly made food.
Arab Street	Lisa K	Greater Adelaide, Aus	59	5	Family	October 10, 2022	Lovely spot	Lovely picturesque spot to take a break. Nice little shops. Tarik on the corner
Arab Street	Explore525662	Bengaluru, India	5	5	Couples	September 2, 2022	Capadocia Restaurant Turkish	Cappadocia Turkish restaurant is a lovely place in Arab street ..great food and
Arab Street	Pragya	Mumbai, India	62	4	Friends	June 10, 2022	Lovely for a casual dinner	Great for dinner. If you want to take a break from the lovely Asian food, you s
Arab Street	G2Kpauly	Melbourne, Australia	1784	4	Solo	January 12, 2022	Ok for a walk down	I walked down this street and didn't find it that exciting. Sure it's ok to see al
Arab Street	linfame	Singapore, Singapore	608	5		April 27, 2021	places to sketch in singapore	because of covid I thought it was appropriate to show placesin singapore... w
Arab Street	Ray	Staines, UK	786	5		April 15, 2021	Arab Street	Arab Street is part of the Muslim Quarter. Singapore's founder Sir Stamford
Arab Street	Chelsea B	Milton Keynes, UK	1963	4		November 13, 2020	Arab Street	Loved this area, full of street art and trendy boutiques, it was wonderful to w
Arab Street	MLSingapore	Singapore, Singapore	217	1		November 6, 2020	Sightseeing is fine,just don't b	The Arab Street is terrible in term of shopping for middle east or Turkish ligh
Arab Street	Thomas V	Oakland, CA	14870	4		October 31, 2020	A Special Neighborhood	This is a unique experience, an ethnic area in this very Chinese city. So come
Arab Street	Aslam_Sherif	London, UK	752	5		May 21, 2020	It feels like you are in Middle E	When you walk through the streets it feels like you are in middle east. Sultar
Arab Street	Grover R	Pensacola, FL	17681	5		April 8, 2020	Shopping - rugs and fabrics	Arab Street is an entire street of shops. It is more fabric and rugs as you can
Arab Street	Luvingmywine	San Diego, CA	200	5		March 29, 2020	Easy to spend lots of money h	We went here twice in one week to enjoy the food (there was a Lebanese re
Arab Street	patrickperbkk	Perth, Australia	1220	5		March 28, 2020	Great night life hang out!	My Singapore friend introduced Arab street few years ago, since then, I revis
Arab Street	WeiHuang91	Singapore, Singapore	699	4		March 24, 2020	Great place to immerse yourse	Very interesting place with plenty of sights and food options around here. Sh
Arab Street	Karin H	Minneapolis, MN	620	5		March 21, 2020	Lots of little shops and restaur	A fun place to walk around and get lunch. Lots of little souvenir shops selling
Arab Street	Hello in 10 Languag	England, UK	179	4		March 18, 2020	Not Just Arab Street but the Si	I spent a week in Singapore, whilst I did not stay in the area I made the effor
Arab Street	Sharon H	Las Vegas, NV	2079	5		March 16, 2020	Fascinating area	We enjoyed wandering through the streets of this area, exploring all the sma
Arab Street	macedonboy	Glasgow, UK	165955	3		March 10, 2020	More for the streets and landr	Arab Street is a long street in the neighbourhood of Kampong Glam, which u
Arab Street	Geoff-SW4	London, UK	149	3		February 25, 2020	Really felt like a tourist trap	The street itself is lovely, with well preserved shop houses and a fair bit of at
Arab Street	fornells22	Greater London, UK	23	5	Couples	February 19, 2020	Great place to walk around	We stayed on Fraser Street nearby for a holiday. Loved being close to this are
Arab Street	Lindsey F	Peterborough, UK	103	5		February 15, 2020	Wonderful for textile purchase	What a beautiful place if you want any household textiles from fabric to carp
Arab Street	Bassam A	Jeddah, Saudi Arabia	2	4		February 11, 2020	Many good restaurants and sh	There are many good restaurants and shops, also there is historical mosque.
Arab Street	TopKingofKings	Nottingham, UK	550	4		February 11, 2020	Cool street to look around	The area full of Muslim restaurants and stores with Masjid Sultan right in fro

Figure: Snippet of dataset

4.2. Data Preprocessing

The text data was to be processed before we move into predictive analytics. The steps involved in the preprocessing is tokenisation, replacing contractions, case lowering, removing stop words, removing special characters, lemmatization using POS tags.

Tokenisation: Split the text into individual words or tokens. Tokenization is the process of breaking down a text into smaller units, such as words or subwords, which can be further processed or analyzed.

Lowercasing: Convert all text to lowercase. This helps ensure that the same words are treated consistently, regardless of their capitalization.

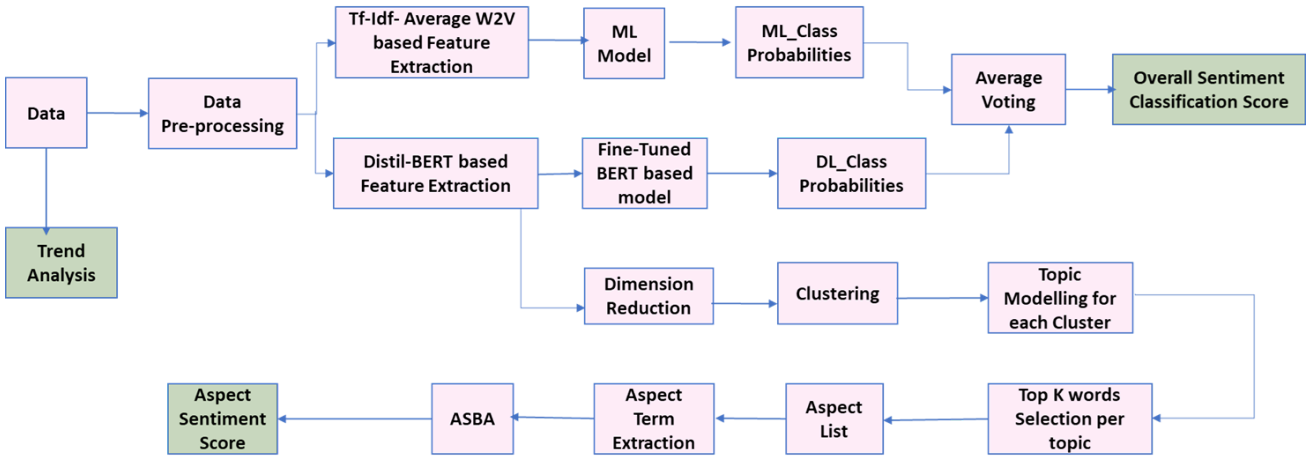
Stop word removal: Remove commonly occurring words that do not carry much significance, such as "a," "an," "the," or "and." These words, known as stop words, are typically filtered out because they occur frequently and do not contribute much to the overall meaning of the text.

Removing special characters: Removing special characters or symbols that may not be relevant or could cause issues during analysis or modeling.

Replacing contractions: This involves expanding abbreviated words into their full forms.

Lemmatization: Lemmatization is the process of reducing words to their base or dictionary form, known as the lemma.We can improve the accuracy of lemmatization by considering the part-of-speech (POS) tags associated with each word. POS tagging is where we assign grammatical tags to words in a sentence, such as noun, verb, adjective, etc.

5. Proposed approaches



We have 3 final outputs namely Overall Sentiment Classification score, Aspect based Sentiment Score and Trend Analysis on the Raw data. We are training or fine - tuning each module separately to get these final 3 outcomes.

5.1. Overall Sentiment Mining

5.1.1. Feature extraction

Machine Learning Based:

The features extraction step is also another important before we move into predictive analytics. The feature extraction was carried out using tfidf-weighted average-word2vec.

TF-IDF refers to Term Frequency - Inverse Document Frequency (TF-IDF). Term Frequency: TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document. Inverse Document Frequency: IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (e.g., a, the, and).The TF-IDF of a term is calculated by multiplying TF and IDF scores.

Word2vec will help get the embeddings associated to each unique word. Word embeddings is a technique where individual words are transformed into a numerical representation of the word (a vector). Where each word is mapped to one vector, this vector is then learned in a way which resembles a neural network. The vectors try to capture various characteristics of that word with regard to the overall text. These characteristics can include the semantic relationship of the word, definitions, context, etc.

TF-IDF (Term Frequency-Inverse Document Frequency) weighted average Word2Vec is a technique that combines the Word2Vec word embeddings with TF-IDF weighting to represent text data in a numerical vector format as shown in the diagram below.

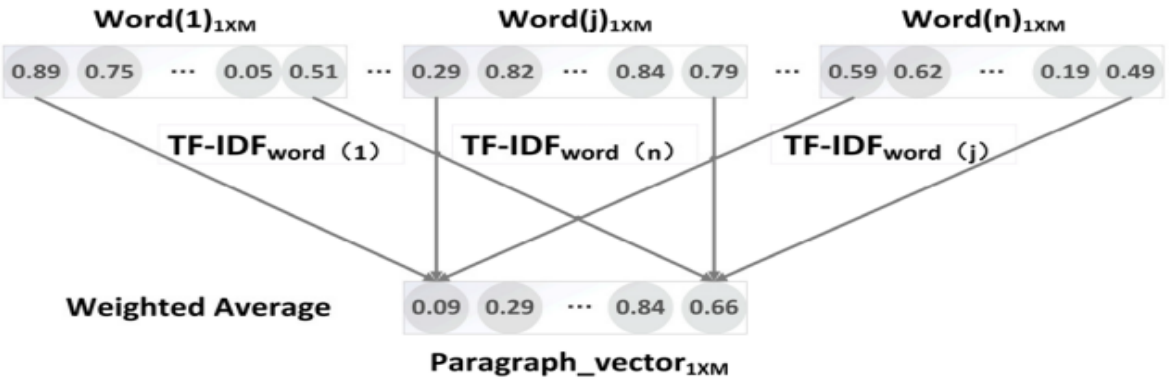


Figure: tfidf-weightedaverage-word2vec

Deep Learning Based:

For the Fine Tuned Bert Model, we need to create different type of embeddings. To use a pre-trained BERT model, we need to convert the input data into an appropriate format so that each sentence can be sent to the pre-trained model to obtain the corresponding embedding. A single vector representing the whole input sentence is needed to be fed to a classifier. In BERT, the decision is that the hidden state of the first token is taken to represent the whole sentence. To achieve this, an additional token [CLS] has to be added manually to the input sentence. The following steps are performed to create the BERT embedding vector for each review:

1. Tokenization: breaking down of the sentence into tokens
2. Adding the [CLS] token at the beginning of the sentence
3. Adding the [SEP] token at the end of the sentence
4. Padding the sentence with [PAD] tokens so that the total length equals to the maximum length
5. Converting each token into their corresponding IDs in the model

The max length of each vector will be 512 dimension. We converted the target labels in Long Tensor, a format needed for the BERT fine tuning. Using the resultant input ids, attention masks and the target label we create the dataloader for the train, cross-validation and test. The train-crossvalidation-test split is considered as 66% - 16% - 18%. With this the dataloaders are ready for training.

5.1.2. Ensemble model

5.1.3. Machine Learning model

The overall steps involved in the Machin Learning pipeline are as follows:

1. Data Pre-processing
2. Feature Extraction
3. Splitting the data i.e train-test split
4. Sampling Methods
    a. Oversampling the minority class
    b. Undersampling the majority class
5. Stratified K-fold cross validation and Model Selection
6. Hyperparameter Tuning
7. Model Evaluation on test set

Data Pre-processing and Feature Extraction The data pre-processing and feature extraction was carried out as mentioned above to convert the text data into numerical representation which further used for creating the predictive models. Predictive analytics was carried out using Machine Learning model for the overall sentiment mining. The labelled ratings have a range from 1-5, with 1 indicating the lowest rating and 5 indicating to hightest rating. The rating 1-5 can be interpreted as the sentiment i.e 1: terrible 2: bad 3: neutral 4: good 5:excellent.

The dataset is heavily imbalanced because most of the ratings belonged to ratings class 5 which can be seen from the diagram below. Hence used oversampling technique to overcome this challenge.

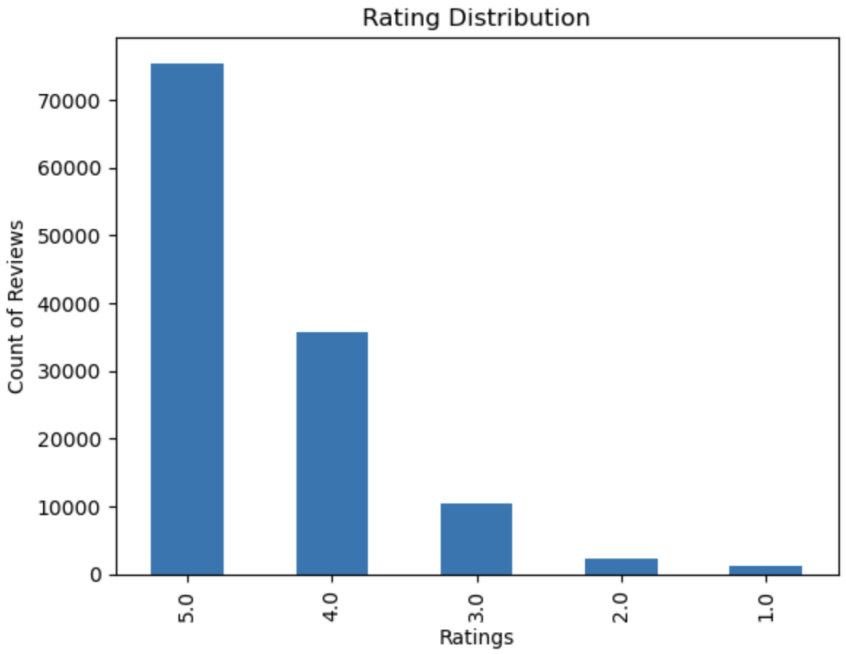


Figure: Dataset Count of Reviwes vs Rating

TrainTest Split Data is split in to 3 parts training, validation and test in a ratio of 60:20:20.. Training set is further split in to 5 folds to select the model using stratified Cross validation strategy.

Sampling methods To deal with data imbalance issue, minority class has been upsampled to increase the availability of reviews and downsampled the majority class. The techniques used in upsampling the minority class was SMOTE, ADASYN and Random Over Sampling. *SMOTE* stands for Synthetic Minority Over-sampling Technique. SMOTE works by synthesizing new examples of the minority class. It randomly selects an instance from the minority class and finds its k nearest neighbors. It then selects one of the neighbors and generates a synthetic instance by linearly interpolating between the selected instance and its neighbor. This process is repeated for a specified number of times, creating multiple synthetic instances. *ADASYN* stands for Adaptive Synthetic Sampling. It generates synthetic examples for minority class samples by introducing random perturbations based on the density distribution of the data. This helps to balance the class distribution and improve the performance of classifiers in handling imbalanced datasets. *Random Over Sampling* randomly duplicates or replicates samples from the minority class to match the number of instances in the majority class. This helps to balance the class distribution and can improve the performance of classifiers in handling imbalanced datasets. The technique used is undersampling the minority class was Random Under sampling. This technique in which the minority class instances are randomly replicated to increase their representation in the dataset. This helps to balance the class distribution by artificially inflating the number of minority class samples, potentially improving the performance of classifiers in handling imbalanced datasets.Apart from the sampling techniques class weight balancing was used. Class weight balancing is a technique used in machine learning to address class imbalance by assigning different weights to the classes during model training. It involves assigning higher weights to the minority class samples and lower weights to the majority class samples, which helps the model to pay more attention to the minority class during training. This helps to mitigate the bias towards the majority class and improves the model's ability to learn from the minority class instances, leading to better performance on imbalanced datasets.

Stratified K fold Cross Validation and Model Selection was used for selecting the ML model to predict the rating given by the user. Stratified K-fold cross validation method is considered so that it maintains the same distribution of classes across different folds. It involves dividing the dataset into k subsets, ensuring that each fold contains a proportional representation of the different classes. This technique helps mitigate the risk of biased performance estimates, especially when dealing with imbalanced datasets. By training and testing the model on various folds, it provides a more robust assessment of its generalization ability. The final performance is typically computed by averaging the evaluation metrics across all folds. In this use case data was split in to 5 folds. The different Machine Learning algorithms used were Logistic Regression, Randomforest, XGB i.e Extreme Gradient Boosting, and LGBM i.e Light Gradient boosting machine. ML model performing better consistently across all the 5 folds used as the final best performing model. Since, the dataset is highly imbalanced macro Averaged F1-Score used as the evaluation metric to validate the model performance. It is the harmonic mean of precision and recall, providing a balanced measure of both metrics. *Extreme Gradient Boosting* (XGB) performed best i.e highest F1-score among all the models where the minority class was oversampled using SMOTE and majority class was undersampled using Random Under sampling.

Hyperparameter Tuning Parameters such as max depth, col sample by tree, n_estimators, and regression alpha were tuned using the Bayesian optimisation algorithm.

Best Model evaluation The results of the best performing model on train and test set are summaried below. The confusion matrix is displayed below: Ratings were relabelled from 0 to 4 for training model.

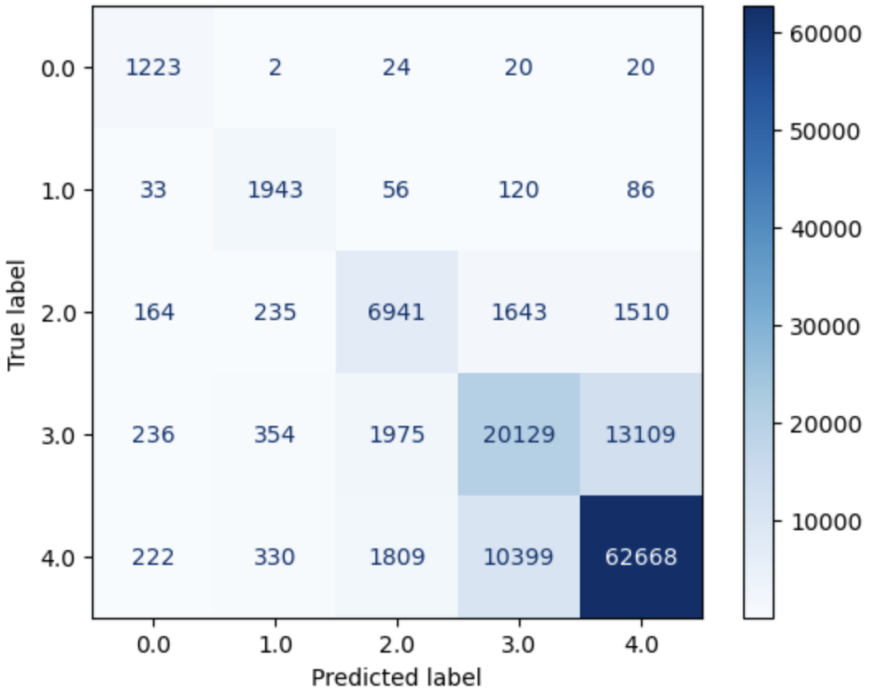


Figure: Train set confusion matrix using XGB

The confusion matrix of the test set is displayed below:

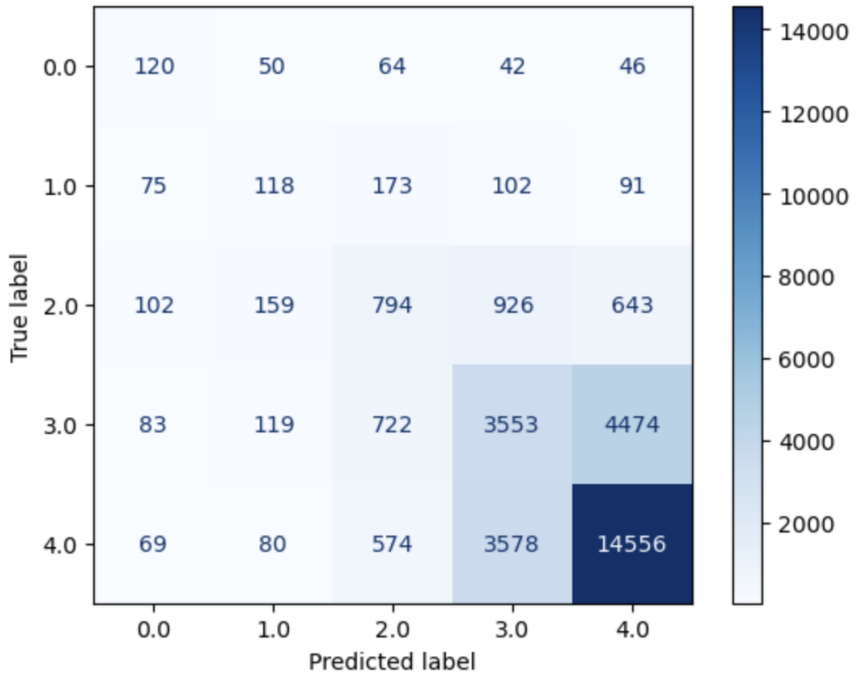


Figure: Confusion matrix of train set using XGB

5.1.4. Deep Learning model

We tried with two approaches to fine tune the bert model. The first approach is to fine-tune the entire BERT model with the new dataset. This approach takes quite long time to train as we are updating the model weights of the entire BERT. The second approach is to use the last hidden state of the '[CLS]' token vector as the input to our custom classifier classifier. [CLS] stands for classification. It is added at the beginning because the training tasks here is sentence classification. This '[CLS]' token vector can encapsulate an aggregated representation of the sentence, thereby providing a comprehensive view of the context. We take the last hidden state of each token vector and compute their average to produce a pooled output. This method aims to leverage the individual nuances captured by each token while ensuring the computational feasibility through averaging. The custom classifier that we used on top of the 'CLS' output is described as below:

```
(classifier): Sequential(
    (0): Linear(in_features=768, out_features=200,bias=True)
    (1): Relu()
    (2): Dropout(p=0.25,inplace=False)
    (3): Linear(in_features=200, out_features=200,bias=True)
    (4): Relu()
    (5): Dropout(p=0.25,inplace=False)
    (6): Linear(in_features=200, out_features=200,bias=True)
    (7): Relu()
    (8): Dropout(p=0.25,inplace=False)
    (9): Linear(in_features=200, out_features=5,bias=True)
)
```

We used the 768 dimension output of the BERT hidden layer and 3 fully connected layers of 200 neurons with Dropout. We tried different architecture using different number of layers, neurons and dropout value. These parameter numbers are the best that we saw during our experimentation.

Our dataset is largely imbalance with the minority class having less than 8% data. Fine-tuning a bert model on such imbalance dataset is pretty challenging. We took quite a few measure to improve our results. To mitigate this imbalance and its potential impact on model performance, we used a weighted loss function that assigns more importance to the minor classes. We used focal-loss with defined class weights. The class weights are given in proportion to their data percentage:

{1: 0.566687901777958, 2: 0.33384490646898707, 3: 0.06955464180228181, 4: 0.02032779289975097, 5: 0.009584757051022182}

In addition to the loss function, we also used different learning rates and epoch numbers to see their impact on the model performance. We train our model with three different learning rates (2e-6, 5e-5, 3e-5) to investigate the influence of learning rate on training efficacy. We didnt experiment with batch size because a larger batch was getting into the problem of memory shortage on Colab Pro.

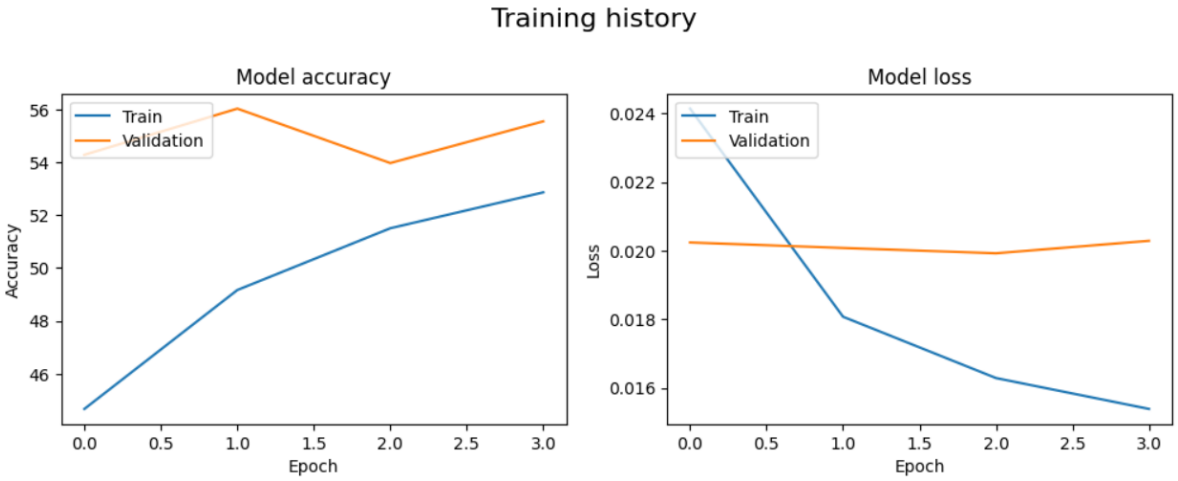


Fig: Loss and Accuracy Curve

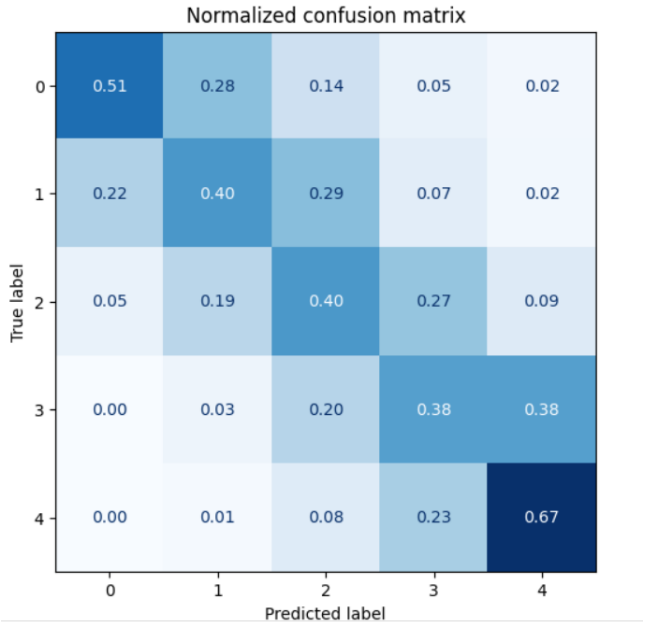


Fig: Test Confusion Matrix

From the loss and the accuracy curve, it is evident that the model is not facing much overfitting. But the model performance didnt improve even after trying all the techniques mentioned above. The huge data imbalance can be a major reason. We see that the model is sometimes not able to differentiate between Rating 4 (given as 3) and Rating 5 (given as 4). Apart from these two classes, the model is not much confused.

5.1.5. Ensemble model

Once we have the outputs of the Machine Learning and Fine tune Bert Model, we ensemble these outputs using a soft voting technique to get the final overall sentiment rating. The ensembling helps to mitigate cases where one model gives high probability to a wrong class and the other model gives high probability to correct class and very low probability to wrong class. To design the ensemble model, we took the class probabilities for each review from each model and averaged it out to give the final sentiment rating. Each amusement has its own overall sentiment score distribution along with exact count of the scores. An example of the overall sentiment scores of amusement **'Arab Street'** is given below:

place	Ensemble_Ratin	Count
Arab Street	4	547
Arab Street	5	187
Arab Street	3	137
Arab Street	2	26
Arab Street	1	9

With this the overall sentiment mining pipeline is finished.

5.2. Aspect Mining

For the Aspect Based Sentiment Mining, we first need to determine the aspects of the each sentences. We want to make sure that similar reviews having similar topics are clustered together so that we can find topics within these clusters.There are several rule based methods using dependency parsing. But considering the volume of our data, this process is extremely tedious and time taking. So, we followed a different approach for Aspect Mining. We first extracted BERT embeddings for the sentences, then used UMAP to reduce the dimension of the embeddings to 10. Further, applied KMeans on the reduced BERT embeddings to get clusters on which topic modelling is applied. After that, for each amusement and for each cluster, we select the top 30 words from each topics. These words are then summarised manually to give the aspect word and the corresponding 30 words for that aspect is the aspect terms. The details of this process is mentioned in the below segments.

5.2.1. Clustering

BERT Embeddings are created in the way described in the Deep Learning Based Feature Extraction section. We have 512 dimension embeddings from bert. Since KMeans doesnt perform very well on extremely high dimension data, so we first decided to reduce the dimensions of the embeddings from 512 to 10 using UMAP. Out of the few dimensionality reduction algorithms, UMAP is arguably the best performing as it keeps a significant portion of the high-dimensional local structure in lower dimensionality. We experimented with the best dimension size and selected 10 as it gave the best clusters. We used the neighbourhood size as 30 and metric as cosine similarity to do the dimension reduction.

After standardizing the reduced umap embeddings using MaxAbsScaler, we proceeded to build the clustering model. We selected KMeans as the clustering algorithm because the NMF (Non Negative Matrix Factorisation) topic modelling doesnt work too well with density based clusters. We hyper-parameter tuned the number of clusters and checked the Silhoutte Coefficient for every cluster. Best cluster number is decided to be 4 following the SilhouetteVisualizer and KElbowVisualizer visualizations. The visualizations are mentioned below:

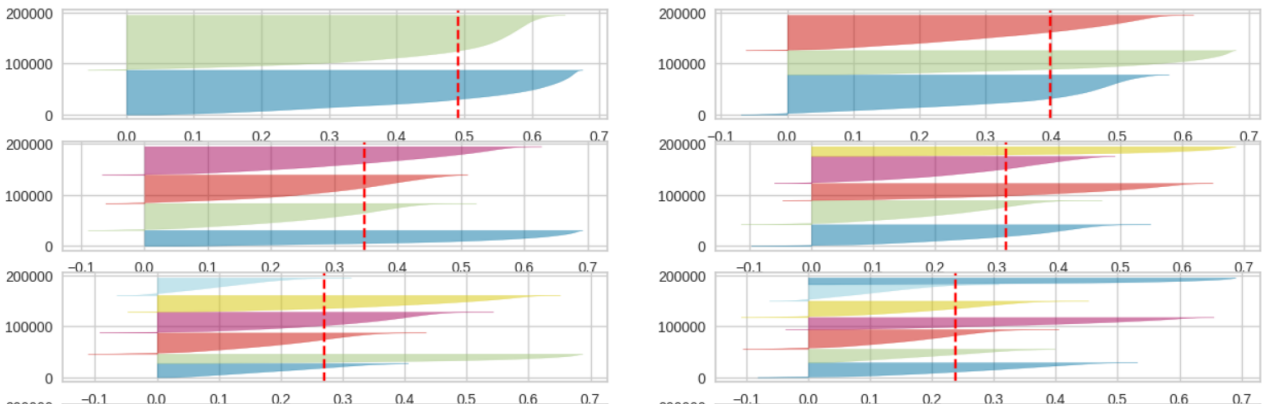


Fig: Silhouette Visualization of the clusters

We considered cluster 3 and cluster 4 but later we realised in cluster 3, two clusters are heavily overlapped. So we decided to go with cluster 4. Same is given by the KElbowVisualizer as mentioned below:

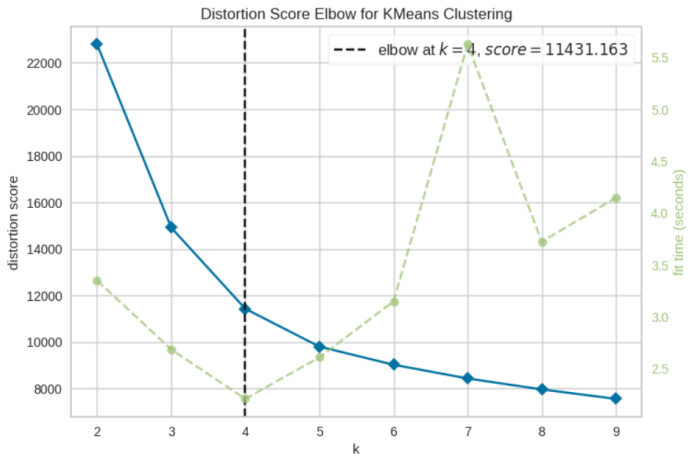


Fig:KElbowVisualizer of the clusters.

To visualize the clusters even better, we reduced the dimension to 2 and then plotted the data. The UMAP reduced dimension data is shown in the figure below:

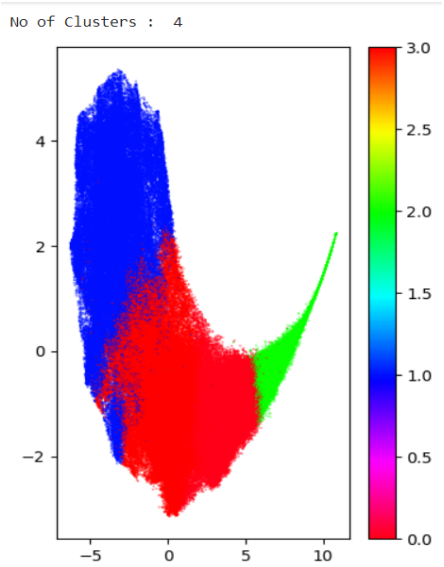


Fig: Umap Visualization of KMeans clusters

From the visualization we can see that 3 clusters are nicely separated but 1 cluster is entirely overlapped. Since this is the best results that we got, we decided to proceed with this and not consider the overlapped cluster to determine the aspect words. We then append the cluster number for each review in the dataset.

The cluster sizes are 46160, 65710, 25306, 58530 for all four clusters. Once the clusters are created, we proceed to the topic modelling part to get the final aspect terms list.

5.2.2. Topic modelling

To get the aspect terms or the topic words for each topic we would need to maintain the original vocabulary of the dataset (i.e. the words of the reviews). Since the embeddings used for clustering are entirely different from the original vocabulary words, we cannot use them in the topic modelling module. So, in order to create the vectorised dataset for each cluster, we first filtered out all the reviews corresponding to each cluster number. We followed all the preprocessing steps as mentioned previously. In addition to that, we did lemmatization and considered to keep words having pos tags as the noun and noun phrases. This decision is taken considering the fact that aspects are generally noun terms only.

Once the reviews are processed according to the requirements of the topic modelling process, we proceed to first hyper-parameter tune the topic numbers. The only parameter that is required is the number of components i.e. the number of topics we want. This is the most crucial step in the whole topic modeling process and will greatly affect how good the final topics are. To evaluate the best number of topics, we can use the coherence score. There are a few different types of coherence score with the two most popular being c_v and u_mass. c_v is more accurate while u_mass is faster. We'll be using c_v here which ranges from 0 to 1 with 1 being perfectly coherent topics. We'll use gensim to get the best number of topics with the coherence score and then use that number of topics for the sklearn implementation of NMF. Sklearn's implementation of NMF is considered better because it can use tf-idf weights which work better as opposed to just the raw counts of words which gensim's implementation is only able to use.

Once we have the best number of topics from the gensim's module, we do the tf-idf vectorization using only unigram features. We used max features as 10,000 so that very common words are not captured. To improve this feature set, we further used ChiSquare and selected best 3000 features for every cluster. Once we have the tf-idf vectorised dataset with 3000 features, we standardised it using StandardScaler. On this standardised dataset, we used the previously chosen best topic number for this cluster and applied it on the SKlearn's NMF.

For every topic, we considered the top 30 words having higher probability as representative of that topic. These topic words are also called aspect terms. The resultant dataset snapshot is mentioned below:

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10	...	Word 21	Word 22	Word 23	Word 24	Word 25	Word 26	Word 27	Word 28	Word 29	Word 30
Topic 1	bird	kid	tram	park	enclosure	safari	show	breakfast	orangutan	elephant	...	habitat	utans	lion	night	ride	monkey	feeding	experience	parrot	creature
Topic 2	food	street	souvenir	market	shop	price	item	clothes	town	bargain	...	everything	chinatown	shirt	jewellery	gift	product	grocery	india	mosque	trinket
Topic 3	view	sand	pool	hotel	flyer	infinity	marina	deck	city	observation	...	merlion	time	sunset	minute	picture	drink	floor	bridge	ferris	experience
Topic 4	tour	guide	history	bunker	battlebox	surrender	fort	jeff	knowledge	decision	...	room	singapore	invasion	information	film	fall	james	buff	passion	group
Topic 5	system	train	card	station	travel	pas	transport	bus	machine	deposit	...	convenient	cash	mass	metro	purchase	destination	transportation	mode	link	pass
Topic 6	rice	chicken	hawker	maxwell	tian	porridge	stall	food	juice	hainanese	...	fritter	rojak	char	dumpling	pork	congee	tien	carrot	sauce	plate

Fig: Topic Representaion for one cluster

The above steps are followed for each of the four clusters. Below table describes the topic numbers per cluster:

Cluster Index	Topic Numbers
0	15
1	20
2	15
3	25

Once we have the topic words for each topic of each cluster, we need to do one final step to get the aspect terms corresponding to each amusement. We save the corresponding topic number for every sentence in a column and append the column with the main dataset. Now, for each review, we have its cluster index number as well its topic number.

To get the aspect words and their corresponding aspect terms, we first filter out the reviews for a particular amusement. Then we do an additional filtering based on cluster number. Suppose the amusement name is Singapore Zoo. We first get all the reviews for Singapore Zoo. We filter them according to the unique clusters we have for the reviews of Singapore Zoo. For each cluster, we check 3 topics having highest counts. Now, for each topic we check the top 30 words. So an amusement can have a maximum of 12 topics. Out of these 12 topics (from all clusters), we select top 3-4 topics depending on the quality and relevance of the aspect terms with the amusements. These 3-4 topics are thus the aspects for the amusement (eg. Singapore Zoo) and the corresponding topic words are aspect terms. Sometimes, the aspects for some amusement is common with many other amusements like Transport and Cleanliness. Following the above approach, we managed to get aspects very specific to an amusement as well as global level (common) aspects.

This process is followed for all 20 amusements. Finally we have around 120 aspects that can describe all the 20 amusements quite well. We manually assigned each list of aspects a sensible and meaningful name to understand different aspects within the reviews. A snapshot of the aspect terms and summarised aspects corresponding to amusement '**Arab Street**' is given.

Arab Street	Culture & Religion	'street', 'lane', 'muslim', 'soak', 'sultan', 'neighbourhood', 'haji', 'pray', 'mosque', 'hindu', 'halal', 'quarter', 'shorter', 'faith', 'community'
	Shopping	'shisha', 'textile', 'ware', 'silk', 'shop', 'rug', 'mood', 'smoking', 'fabric', 'score', 'shophouses', 'smoke', 'boutiques', 'souvenirs', 'merchandise'
	Food	restuarant', 'pub', 'dining', 'food', 'cuisine', 'meal', 'dish', 'eatery', 'delicacy'

Table: Summarised Aspects for Arab Street

As is evident, the aspect term extraction is performing quite well. Using these aspect terms, we would now continue to do the next procedure of Aspect-Based Sentiment Analysis.

5.3. Aspect-Based Sentiment Analysis

5.3.1. Data Preprocessing

Upon obtaining a list of aspects and their corresponding aspect terms through topic modelling, we proceed with Aspect-Based Sentiment Analysis (ABSA). Initially, we scan each review sentence for mentions of the identified aspects. For instance, a sentence containing animal-related words like "elephant", "birds", or "butterflies" would be associated with the "Animals" aspect. However, we observe instances where a single sentence addresses multiple aspects. From our analysis, 6244 out of the total 74238 sentences mention multiple aspects.

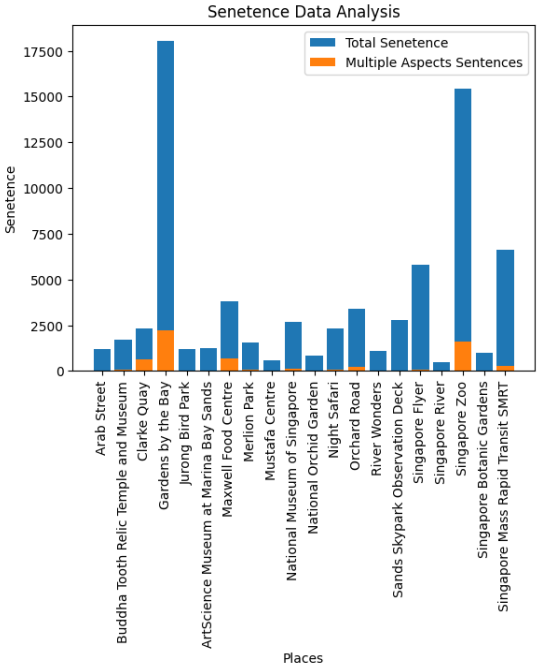


Fig: Sentence Data Analysis

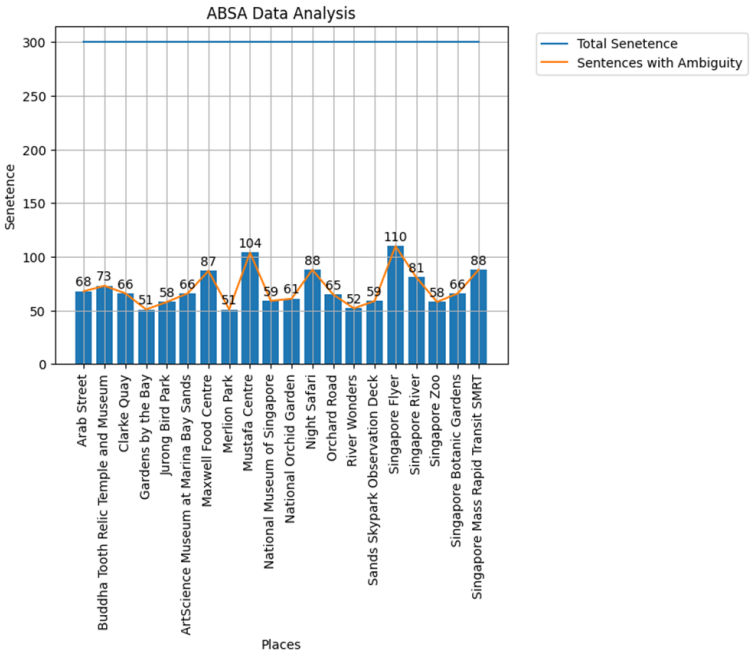


Fig: ABAS Data Analysis

In order to overcome the challenge of multiple aspects within a single sentence, we employ an open-source model called Span-ASTE which is acclaimed for its robust performance in Aspect Term Extraction and Opinion Term Extraction tasks Xu et al. (2021). We leverage Span-ASTE to extract and predict the sentiment of sentences that mention multiple aspects. These predictions are subsequently integrated with the sentiment labels predicted for sentences addressing a single aspect. This combined approach enhances the overall performance of our ABSA system.

Following this, we proceed to train a model with the specific aim of predicting sentiments in sentences that address a single aspect. We generate pairs of aspects and their corresponding aspect-specific sentences. Each pair is then labelled with its sentiment, forming the foundation for training our sentiment prediction model. For model training and evaluation, we randomly select 300 sample sentences from each place, accumulating a total of 6000 labelled sentences.

5.3.2. Data Labelling

For data labeling, we utilized three pre-trained sentiment classification models from Hugging Face. These included "*cardiffnlp/twitter-roberta-base-sentiment-latest*", which is a RoBERTa-base model trained on ~124M tweets and fine-tuned for sentiment analysis; "*cardiffnlp/twitter-xlm-roberta-base-sentiment*", a multilingual XLM-roBERTa-base model trained on ~198M tweets and fine-tuned for sentiment analysis in multiple languages; and "finiteautomata/bertweet-base-sentiment-analysis", a RoBERTa model trained on the SemEval 2017 corpus, consisting of around ~40k tweets.

We derived labels by feeding the models with pairs of aspects and aspect-specific sentences, and then asking them to classify the sentiment as either positive, negative, or neutral. However, we encountered a challenge where these pre-trained models, having been trained on generic datasets, didn't always generate accurate labels for our data which was specific to the amusement industry. Consequently, we aggregated the results from these three different sentiment classification models to identify sentences with ambiguous sentiment. An ambiguous sentiment is defined as a case where the labels generated by the three models were not in agreement. We discovered that 15% of the data was characterised by ambiguous sentiment.

Initially, our plan was to distribute the task of labelling these ambiguous sentences equally among our team members, with a 10% overlap to verify agreement. However, due to time constraints, we decided to use GPT-3.5 to help us label the rest of the data. We validated this approach by randomly assigning each team member 50 ambiguous sentences and comparing the results generated by GPT-3.5. We found that GPT-3.5 correctly labelled 90% of the data, leading us to decide to use it for the remainder of our data labelling task.

5.3.3. ABSA Model Training

This study proposes a comprehensive method to train an Aspect-Based Sentiment Analysis (ABSA) model using the pre-trained BERT 'bert-base-uncased' as a feature extractor. The BERT model is pre-trained on a vast corpus in an unsupervised manner, enabling it to acquire generic representations of tokens from extensive textual data (Merchant et al., 2020). Leveraging BERT's exceptional ability to comprehend sentence context, we employ it as a foundational component in our ABSA model. Upon this, we construct a custom classifier designed to further refine our sentiment analysis.

```
(classifier): Sequential(
  (0): Linear(in_features=768, out_features=100, bias=True)
  (1): ReLU()
  (2): Dropout(p=0.25, inplace=False)
  (3): Linear(in_features=100, out_features=100, bias=True)
  (4): ReLU()
  (5): Dropout(p=0.25, inplace=False)
  (6): Linear(in_features=100, out_features=3, bias=True)
)
```

Fig: Structure of Classifier

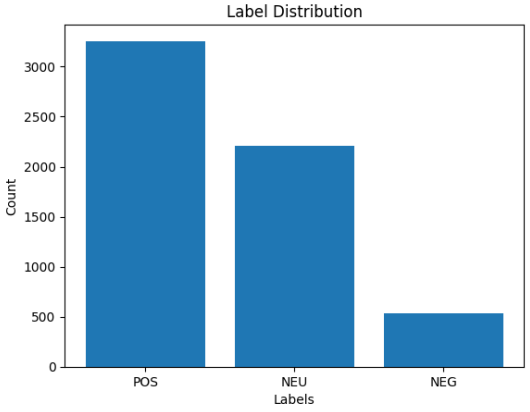


Fig: Class Distribution

We examine two methods of feature extraction to assess their respective influences on model performance. The first method utilises the last hidden state of the '[CLS]' token vector as the input to our classifier. This '[CLS]' token vector stands out due to its ability to encapsulate an aggregate

representation of the sentence, thereby providing a comprehensive view of the context. The second method involves the usage of all token vectors within a sentence. We take the last hidden state of each token vector and compute their average to produce a pooled output. This method aims to leverage the individual nuances captured by each token while ensuring the computational feasibility through averaging.

An important consideration in our study is the imbalanced nature of our dataset. Particularly, the class representing negative sentiment is significantly underrepresented, accounting for only 8% of the data. To mitigate this imbalance and its potential impact on model performance, we adopt a weighted loss function that assigns more importance to the minor classes. We examine two variants of focal loss: one with a fixed alpha value of 0.25, and another with alpha set to the inverse class distribution. In addition, we also employ cross-entropy loss as a baseline for comparison.

In addition to the loss function, our study also examines the impact of various hyperparameters on the model's performance. We train our model with three different learning rates (1e-4, 5e-5, 3e-5) to investigate the influence of learning rate on training efficacy. We also experiment with different batch sizes, including 16, 32, and 64, to understand their effect on learning. Each configuration is trained over three epochs to ensure robust comparison. To prevent overfitting observed when fine-tuning with BERT, all configurations in our study are trained for three epochs, which ensures effective learning while maintaining the model's generalisation ability.

5.4. Review Analysis

5.4.1. Review Sentiment Analysis

Our sentiment analysis aims to evaluate the sentiment and rating distribution across various time periods, providing insights into all reviews, recent two-year reviews, and recent five-year reviews. In addition to overall sentiment analysis, we delve into aspect-based sentiment analysis, allowing us to understand the sentiment associated with specific aspects or features of the attractions.

To gain a deeper understanding of visitor demographics and preferences, we focus on the composition of reviewers. By categorising them as either foreign or local visitors, we can identify the sentiment trends and rating distributions for each group. Furthermore, we analyse the accompanying group types, such as couples, families, singles, friends, or business travellers, to investigate the overall sentiment for each visitor group.

5.4.2. Trend Analysis

We analyse trends in the ratings of amusement attractions and the number of reviews over time to understand how visitor perceptions and experiences may have evolved.

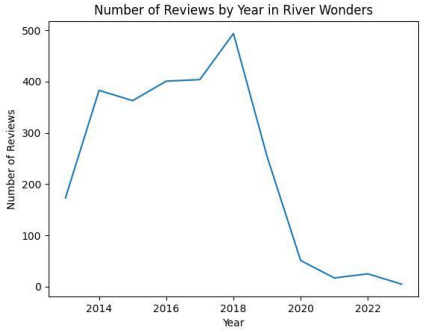


Fig: Trend of Number of Reviews by Year in River Wonders

5.4.3. Feature Analysis

In addition, we conduct feature analysis on each amusement attraction to extract valuable insights. This analysis enables us to generate informative tags that highlight important aspects of the attractions. For example, we identify the primary source of tourists for a specific attraction, such as "local" or "foreigner," and determine if the attraction is highly rated by that particular group. This information helps us classify the attraction as a recommended destination for the respective group of people. We also analyse specific aspects that receive high ratings, such as "Culture & Religion" or "Transportation & Accessibility." By providing these tags, users can easily find attractions that align with their preferences.

5.5. Application Development

5.5.1. Website Development

The web application has been developed to provide users with the ability to visualise review analysis. It leverages React as the framework to implement two primary features:

1. **Search and Visualised Review Analysis**: The application enables users to search for a specific place and view the review analysis pertaining to that place. The review analysis is then presented in a visually intuitive manner, allowing users to comprehend and interpret the information of an amusement effectively.

2. **Place Recommendation Based on User Requirements**: Another significant feature of the application is its capability to suggest places based on user requirements. Users can input their preferences and specifications, and the application will provide relevant recommendations accordingly. This functionality enhances user convenience by offering tailored suggestions that align with their specific needs and preferences.

5.5.2. Chatbot Development

A chatbot has been developed to facilitate natural language interactions with users and recommend suitable places. The chatbot's behaviour is predefined based on different scenarios, outlined in the following table.

Action	Description	Slot
Greeting	Greeting the system (E.g., “hi”).	None
AskService	When user asks about your capability, introduce yourself (E.g., "What can you do?").	None
AskPlace	When user asks for information about a specific place (E.g., “Give me more information about Singapore Zoo.”).	(Place, Value)
RequirPlace	When user describes the type of place they want to visit, suggest a place (E.g., “Find me a place with local food.”).	(Requirement, Value)
Error	When the user asks something irrelevant to places of tourist attraction or out of the bot's scope (E.g., “What will the weather be like tomorrow?”).	None
NotFound	When the bot cannot find the place asked by the user (E.g., “Give me more information about Toronto.”).	(Place, Value)

Table: Predefined Action List

To determine the appropriate action, GPT-3.5 is utilised to understand the user's intent and classify the action type. If no slot extraction is required, a response is randomly selected from a set of predefined responses. However, if slot values need to be extracted, GPT-3.5 is employed to assist in extracting the necessary information. For instance, when a user expresses their requirements using natural language (e.g., "Find me a place suitable for family outings"), the chatbot first utilises GPT-3.5 to identify the action type as "RequirPlace". Then the chatbot fetches GPT-3.5 again to find places that align with the given requirement and provides the user with a list of suitable places along with links to pages containing detailed information about each place. In the event that a specific place is not found within our dataset, the chatbot apologises using a predefined response.

6. Experimental Results

6.1. Aspect-Based Sentiment Analysis

Batch Size	Input Feature	Loss Function	Learning Rate	POS Accuracy	NEU Accuracy	NEG Accuracy	Average Accuracy
16	Averaging the vectors of all the tokens	Focal Loss (alpha = 0.25)	1e-4	0.74	0.82	0.86	0.84
			5e-5	**0.77**	**0.82**	**0.89**	**0.86**
			3e-5	0.65	0.84	0.91	0.87
32	Averaging the vectors of all the tokens	Focal Loss (alpha = 0.25)	1e-4	0.66	0.84	0.9	0.86
			5e-5	0.65	0.79	0.93	0.86
			3e-5	0.65	0.83	0.91	0.86
64	Averaging the vectors of all the tokens	Focal Loss (alpha = 0.25)	1e-4	0.61	0.82	0.81	0.80
			5e-5	0.67	0.81	0.82	0.81
			3e-5	0.60	0.84	0.82	0.81
		Focal Loss (alpha = inverse class distribution)	1e-4	0.59	0.84	0.77	0.78
			5e-5	0.64	0.84	0.87	0.83
			3e-5	0.66	0.85	0.82	0.81
		Cross-Entropy Loss (weights = inverse class distribution)	1e-4	0.54	0.8	0.87	0.81
			5e-5	0.60	0.79	0.88	0.82
			3e-5	0.52	0.81	0.9	0.82
	[CLS] token Vector	Focal Loss (alpha = 0.25)	1e-4	0.45	0.81	0.9	0.81
			5e-5	0.52	0.83	0.88	0.82
			3e-5	0.46	0.83	0.89	0.82

Table: Experiments Results for ABSA Model

To evaluate the performance of our ABSA model, we employed a confusion matrix to measure prediction accuracy for each class and calculate the average accuracy on testing data. Our experimental results indicated that the highest average accuracy of 0.87 was achieved with a batch size of 16, a learning rate of 3e-5, and the utilisation of Focal Loss (alpha = 0.25) as the loss function.

We found that averaging the vectors of all tokens as the input feature and using Focal Loss with alpha = 0.25 consistently yielded higher average accuracies compared to other approaches. Specifically, averaging the token vectors outperformed using the [CLS] token vector as the input feature. Moreover, employing Focal Loss with alpha = 0.25 demonstrated improved prediction accuracy for the minority class compared to using Focal Loss (alpha = inverse class distribution) and Cross-Entropy Loss (weights = inverse class distribution). This suggests that averaging the token vectors may be a more effective approach for the input feature, and Focal Loss with alpha = 0.25 may be a preferable loss function when fine-tuning BERT with an imbalanced dataset. Furthermore, a study by Nayak (2022) supports the notion that Focal Loss is a superior choice compared to Cross-Entropy Loss.

Regarding learning rates, the results were not as consistent across different batch sizes and input features. However, in most cases, learning rates of 5e-5 or 3e-5 outperformed a learning rate of 1e-4 in terms of average accuracy. For batch sizes, the trend observed in the table suggests a decrease in average accuracy as the batch size increases from 16 to 64. Consistently, the highest average accuracies were achieved with batch sizes of 16. It is worth noting that larger batch sizes, as suggested by Masters

Luschi (2018), tend to result in lower accuracy but faster training epochs, which aligns with our findings.

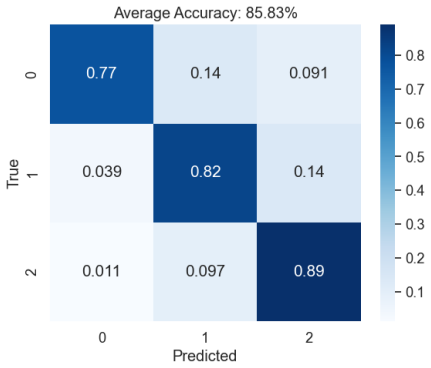


Fig: Confusion Matrix for Final ABSA Model

Our final ABSA model was designed to achieve a balance between accurate prediction of the minor class and high average accuracy. The model trained using a batch size of 16, a learning rate of 5e-5, and Focal Loss with alpha = 0.25 as the loss function is selected as our final ABSA model. This model attained a score of 0.77 for the negative sentiment class and an overall accuracy of 0.86.

7. Conclusions

In conclusion, our project focuses on sentiment mining in the tourism industry of Singapore, specifically analyzing tourist attraction reviews on TripAdvisor. By utilizing aspect-based sentiment analysis (ABSA) techniques, we aim to extract essential aspects and predict the sentiment polarity associated with each aspect. This approach allows us to gain valuable insights into customer experiences and preferences, enabling amusement service providers to enhance their offerings, identify areas for improvement, and monitor changes in sentiment over time. With ABSA being a well-established and effective method in sentiment analysis, we can leverage its capabilities to provide comprehensive and accurate sentiment analysis for the Singapore amusement landscape.

8. Acknowledgements

8.1. ChatGPT Usage

ChatGPT is a tool developed by Open AI. It is based on the GPT-3.5 architecture, which stands for "Generative Pre-trained Transformer 3.5." GPT-3.5 is a state-of-the-art model for natural language processing and generation tasks.

For the purpose of this project, GPT-3.5 has been utilised to develop a chatbot capable of understanding user intents and extracting slot values, enhancing user interactions. GPT-3.5 has also been leveraged for parts of data labelling. Additionally, ChatGPT, based on GPT-3.5, has been employed for report writing, generating comprehensive and insightful reports based on project findings.

With respect to our project, ChatGPT was useful to help provide small snippets of code if we were stuck at any point. We cannot completely rely on ChatGPT's code. Modifications were made according to our usecase. Previously, The search engine Google was helpful to answer our queries. But, this would run through a set of pre-existing web pages and give us results. ChatGPT is a novel innovation because it is able to understand our particular usecase and make suggestions accordingly. We have never encountered a tool before where it is able to understand the context like a human and modify the answers accordingly. During our in-class PLP assignments, the teachers allowed us to use ChatGPT with a disclaimer that we need to accordingly modify ChatGPT's response based on our domain understanding.

In a general context, ChatGPT acts as a starting point for a quick read on any topic. It is useful for summarization of long documents when we want to understand the gist of the article, sending personalised messages to others. Such personalisation was never possible with a tool before. The bot is 24* 7 available, hence we can avail it's services at any time. Also, it responds immediately without any delay.

It is important to note that ChatGPT is not always correct. But, it is definitely a useful tool which will aid us in our task but should be used with human judgement.

9. References

Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *ArXiv:1804.07612 [Cs, Stat]*. https://arxiv.org/abs/1804.07612

Merchant, A., Rahimtoroghi, E., Pavlick, E., & Tenney, I. (2020, April 29). *What Happens To BERT Embeddings During Fine-tuning?* ArXiv.org. https://doi.org/10.48550/arXiv.2004.14448

Nayak, R. (2022, April 28). *Focal Loss : A better alternative for Cross-Entropy*. Medium. https://towardsdatascience.com/focal-loss-a-better-alternative-for-cross-entropy-1d073d92d075

Xu, L., Chia, Y. K., & Bing, L. (2021, August 1). *Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction*. ACLWeb; Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.367

Liu, Haoyue & Chatterjee, Ishani & Zhou, Mengchu & Lu, Sean & Abusorrah, Abdullah. (2020). *Aspect-Based Sentiment Analysis: A Survey of Deep Learning Methods.* IEEE Transactions on Computational Social Systems. PP. 1-18. 10.1109/TCSS.2020.3033302.

Zhang, W., Li, X., Deng, Y., Bing, L., & Lam, W. (2022). A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges. *ArXiv:2203.01054 [Cs]*. https://arxiv.org/abs/2203.01054

Lu Xu, Yew Ken Chia, Lidong Bing(2021)Learning Span-Level Interactions for Aspect Sentiment Triplet Extraction.https://aclanthology.org/2021.acl-long.367/

Grootendorst, M. (2022, March 11). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv.org. https://arxiv.org/abs/2203.05794

UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction — umap 0.5 documentation. (n.d.). https://umap-learn.readthedocs.io/en/latest/

Text classification (huggingface.co)