



# LIVE LIKE A LOCAL

Recommendation system for Airbnb Singapore

## Team Members

|              |           |
|--------------|-----------|
| FAN ZHEYUAN  | A0198497B |
| FANG WENLIN  | A0198417R |
| SHEN CHEN    | A0058260J |
| WANG BOCHEN  | A0198493J |
| XIAO JINTING | A0198422Y |
| YANG XIAORUI | A0198456L |

## Contents

|   |           |
|---|-----------|
| <b>1. Executive Summary .....</b>               | <b>3</b>  |
| <b>2. Report Overview .....</b>                 | <b>4</b>  |
| <b>3. Data Preparation .....</b>                | <b>5</b>  |
| <b>3.1 Dataset Descriptions .....</b>           | <b>5</b>  |
| <b>3.2 Data Cleaning .....</b>                  | <b>6</b>  |
| <b>3.3 Data Transformation .....</b>            | <b>6</b>  |
| <b>3.4 Output .....</b>                         | <b>6</b>  |
| <b>4. Exploratory Analysis .....</b>            | <b>7</b>  |
| <b>4.1 Listings.csv .....</b>                   | <b>7</b>  |
| <b>4.2 Reviews.csv .....</b>                    | <b>8</b>  |
| <b>4.3 Implications .....</b>                   | <b>11</b> |
| <b>5. Collaborative Filtering .....</b>         | <b>12</b> |
| <b>5.1 Predict User Ratings .....</b>           | <b>12</b> |
| <b>5.2 Evaluating Recommendations .....</b>     | <b>18</b> |
| <b>6. Content-based Filtering .....</b>         | <b>19</b> |
| <b>6.1 Making up Corpus Object .....</b>        | <b>19</b> |
| <b>6.2 Document Similarity Algorithms .....</b> | <b>19</b> |
| <b>7. Limitations and Conclusion .....</b>      | <b>22</b> |
| <b>7.1 Limitations .....</b>                    | <b>22</b> |
| <b>7.2 Further Improvements .....</b>           | <b>22</b> |
| <b>7.3 Implementation Considerations .....</b>  | <b>22</b> |
| <b>Appendix 1: Algorithms Description .....</b> | <b>23</b> |
| <b>Appendix 2: Topics Description .....</b>     | <b>24</b> |

## List of Figures

|  |    |
|--|----|
| Figure 1 Overview of system development process flow ..... | 4  |
| Figure 2 Listings by region .....                          | 7  |
| Figure 3 Listing by property type .....                    | 7  |
| Figure 4 Criterion for highly rated listings .....         | 8  |
| Figure 5 Popularity of Airbnb.....                         | 8  |
| Figure 6 Sparse distribution of ratings .....              | 9  |
| Figure 7 Distribution of review sentiments .....           | 10 |
| Figure 8 Word cloud for positive vs negative reviews ..... | 10 |
| Figure 9 Plutchik's wheel of emotions .....                | 10 |
| Figure 10 Length of review vs review sentiments.....       | 11 |
| Figure 11 Collaborative filtering process flow .....       | 12 |
| Figure 12 KNN iteration process .....                      | 14 |
| Figure 13 Matrix factorization iteration process .....     | 16 |
| Figure 14 Iteration outcome for Matrix factorization.....  | 17 |
| Figure 15 DTM process flow .....                           | 19 |
| Figure 16 K-means clustering process flow .....            | 19 |
| Figure 17 Topic modelling process flow .....               | 20 |
| Figure 18 Perplexity value per number of topics .....      | 20 |

## List of Tables

|   |    |
|---|----|
| Table 1 Variable descriptions in Listing.csv table .....                              | 5  |
| Table 2 Variable descriptions in Review.csv table .....                               | 5  |
| Table 3 Data cleaning for Listing.csv .....   | 6  |
| Table 4 Data cleaning for Reviews.csv.....  | 6  |
| Table 5 Variable transformation for Listing.csv .....                                 | 6  |
| Table 6 Count of dataset objects before and after data cleaning .....                 | 6  |
| Table 7 qdap sentimental analysis .....   | 9  |
| Table 8 Comparison of “qdap” and “sentimentr” based on % of total words captured..... | 13 |
| Table 9 Comparison of “qdap” and “sentimentr” based on unique words captured .....    | 13 |
| Table 10 Polarity score normalization.....  | 13 |
| Table 11 Comparing user-based and item-based filtering approaches .....               | 15 |
| Table 12 Criterion to evaluate collaborative filtering approaches .....               | 17 |
| Table 13 ALS and SVD model test .....   | 18 |
| Table 14 Evaluation of k-means clustering.....  | 21 |

## 1. Executive Summary

Airbnb, Inc. is an online platform offering short-term lodging and tourism-related services. It serves two types of customers – hosts and guests. Hosts list their properties on the platform, and guests are the consumers of these listings. Airbnb receives commission from each successful booking.

This report looked into **building a recommender system** for Airbnb's primary business – **lodging, in Singapore**. Using collaborative filtering and content-based filtering methods, the recommender system provided personalized listing recommendations to a guest browsing for lodges based on the guests' preferences and behaviors.

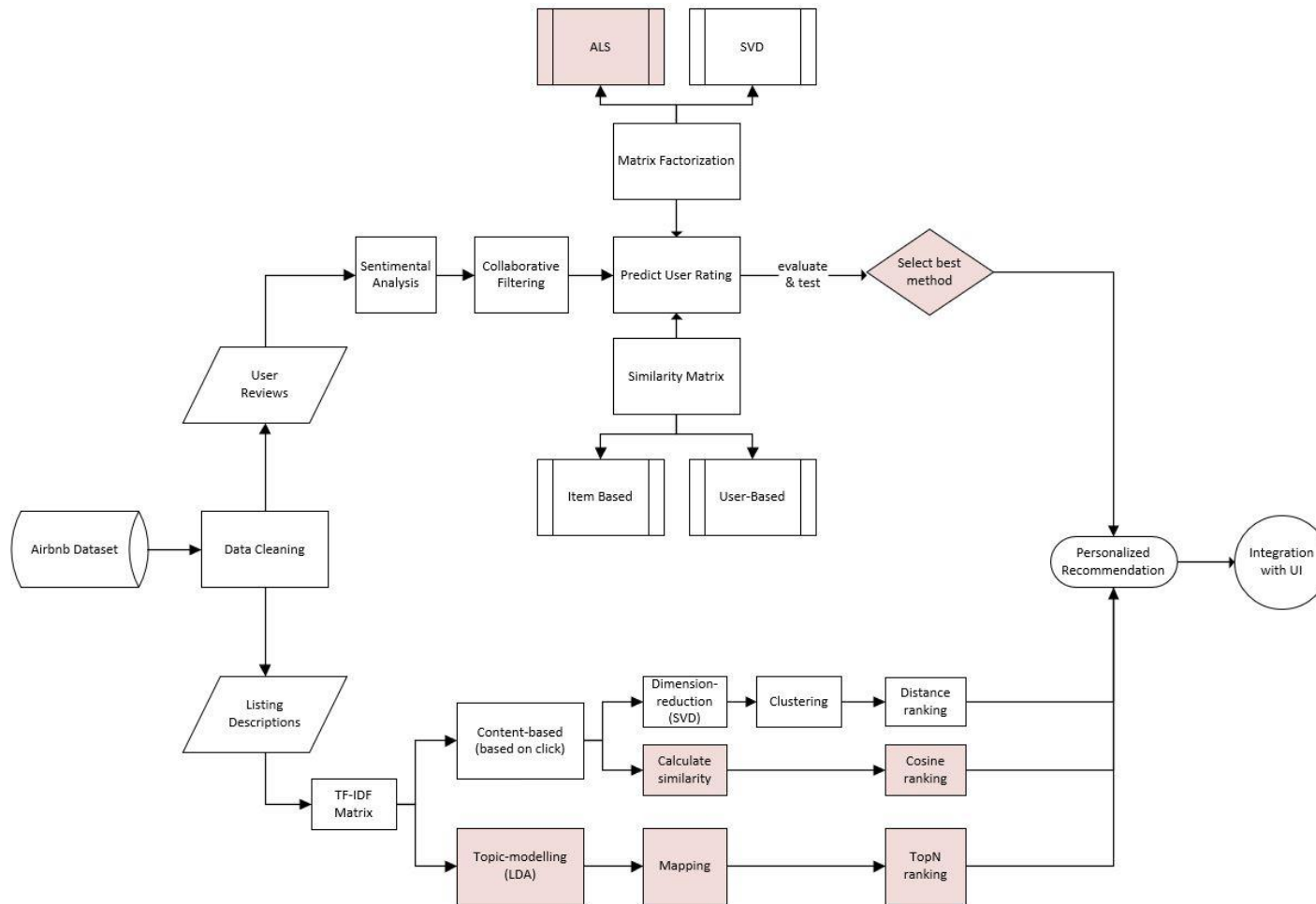
With the recommender system, the following business objective will be fulfilled:

- a) **Increase booking** on the platform by recommending listings that are preferred by the guests
- b) **Address the long tail problem** by including less popular listings in the recommendations

## 2. Report Overview

The analytical process of the recommender system was summarized in Figure 1.

Figure 1 Overview of system development process flow



### 3. Data Preparation

#### 3.1 Dataset Descriptions

We used two tables from the dataset of Singapore from [Inside Airbnb](#) website to build the recommender system:

- 1) **listings** - detailed listings data showing 96 attributes for each of the listings, in which *8 are useful* for our analysis
- 2) **reviews** - detailed reviews from *May'11 to Aug'19* given by the guests with 6 attributes, of which *4 were useful* for our analysis

The descriptions for the selected attributes were in Table 1 & Table 2.

Table 1 Variable descriptions in Listing.csv table

| Listings.csv                |          |   |
|-----------------------------|----------|---|
| Variable                    | Type     | Description   |
| listing_id                  | discrete | Unique identifier of each listing   |
| description                 | textual  | Descriptions of the interior environment (apartment) of each listing  |
| neighborhood_overview       | textual  | Descriptions of the exterior environment (neighborhood) of each listing   |
| transit                     | textual  | Information on access to public transportation, including transit time between nearest station and listing  |
| access                      | textual  | Descriptions of facilities within the neighborhood, and hardware facilities within the house, such as 24-hour hot water supply, air conditioning, wifi, kitchen, etc. |
| neighborhood                | textual  | The general geographical district where the listing belonged to   |
| neighborhood_cleansed       | textual  | Detailed geographical district where the listing belonged to (i.e. distinct district for those classified as "Central Area" in the attribute "neighborhood")          |
| neighborhood_group_cleansed | textual  | The geographical distribution of the listing neighborhood (i.e. North, South, Central, West Region of Singapore etc.)   |

Table 2 Variable descriptions in Review.csv table

| Reviews.csv |          |   |
|-------------|----------|---|
| Variable    | Type     | Description   |
| listing_id  | discrete | Unique identifier of each listing                       |
| id          | discrete | Unique identifier of each review                        |
| reviewer_id | discrete | Unique identifier of each reviewer (a.k.a guest)        |
| comments    | textual  | User <sup>1</sup> reviews for the listing of their stay |

---

<sup>1</sup> The term "user" will be interchangeably used with "guest"

### 3.2 Data Cleaning

We cleaned the data for both tables in the Airbnb Singapore dataset. Cleaning procedures were summarized in Table 3 &

Table 4.

Table 3 Data cleaning for Listing.csv

| Attribute   | Issue                                     | Solution        |
|-------------|---|-----------------|
| description | foreign language (non-English characters) | drop characters |
|             | Special character "€"                     | drop characters |

Table 4 Data cleaning for Reviews.csv

| Variable | Issue  | Solution  |
|----------|--|-----------|
| comments | 73 missing values  | drop rows |
|          | foreign language (non-English and English characters) <sup>2</sup> | drop rows |
|          | system generated comments for cancelled reservations               | drop rows |

### 3.3 Data Transformation

7 variables in Listing.csv table was merged to into a **new column “desc\_all”** to concatenate the descriptive textual information for each Airbnb listing, as shown in Table 5.

Table 5 Variable transformation for Listing.csv

| Listings.csv                |           |              |
|-----------------------------|-----------|--------------|
| Variable                    | Action    | New Variable |
| listing_id                  | No change | listing_id   |
| description                 | merge     | desc_all     |
| neighborhood_overview       |           |              |
| transit                     |           |              |
| access                      |           |              |
| neighborhood                |           |              |
| neighborhood_cleansed       |           |              |
| neighborhood_group_cleansed |           |              |

### 3.4 Output

Table 6 Count of dataset objects before and after data cleaning

|          | No. of rows |         |
|----------|-------------|---------|
|          | Raw         | Cleaned |
| Listings | 7,907       | 7,907   |
| Reviews  | 101,268     | 69,646  |

<sup>2</sup> Foreign language in English alphabets are detected with the “textcat” package

## 4. Exploratory Analysis

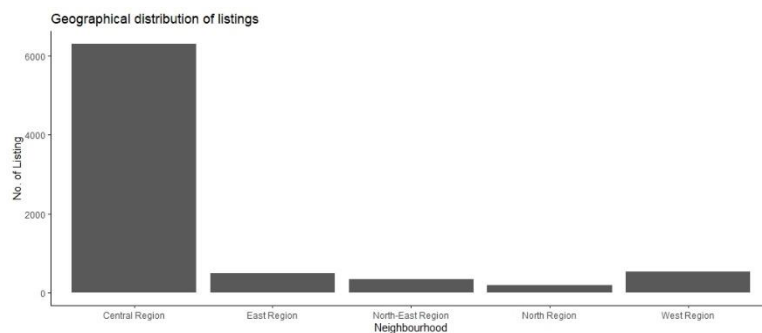
We explored the two tables (Listing.csv & Reviews.csv) to gain **preliminary understanding** of the datasets, as well as to **discover potential issues** that may surface in our attempt to build the recommender system. Findings were summarized in Section 4.3.

### 4.1 Listings.csv

#### 4.1.1 Geographical Popularity

Figure 2 showed the distribution of Airbnb listings by region. The number of listings with location in **central area** (i.e. Orchard, Newton, Queenstown, River Valley, Marine Parade, etc.) dominated over the other neighborhoods plausibly due to the concentration of tourist attractions in central Singapore.

Figure 2 Listings by region

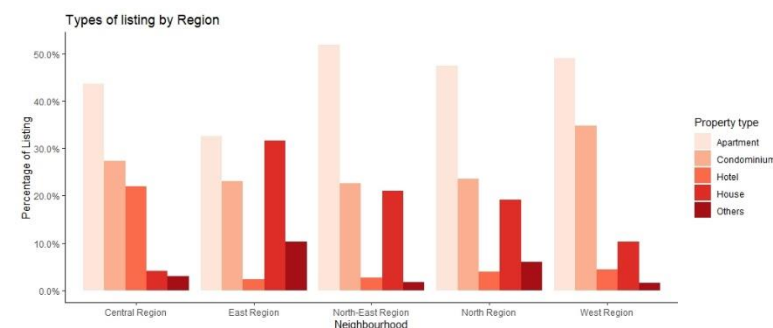


#### 4.1.2 Type of Listing

Figure 3 showed that **condominium was the most popular type of listing** in all region, accounting for more than 50% of total listings. In the central area, hotel (including service apartment, hostel, boutique hotels etc.) were also popular choices.

It is interesting to note that in the East Region, there was a relatively large proportion of property type labelled as “Houses” and “Others”, which includes options like Chalet, campsites, and tents. These may be rental options targeted at the locals looking for a weekend getaway rather than tourists.

Figure 3 Listing by property type





### 4.1.3 Listing descriptions

Figure 4 was a word cloud generated using high frequency words from the descriptions of listings with perfect rating score (100/100). It highlighted **important features** that defined a **popular listing** such as easy access to public transport, centrality of listing locations, availability of facilities, size of listing and stylish interior furnishing.

Figure 4 Criterion for highly rated listings

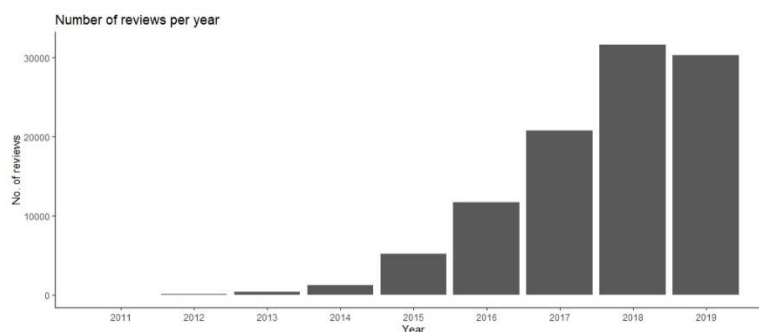


## 4.2 Reviews.csv

#### 4.1.4 Distribution of reviews overtime

Figure 5 showed that the number of reviews increased exponentially over the years from 2011 to 2019 (data up to 3rd quarter), implying the **increasing popularity** of Airbnb rentals in Singapore.

Figure 5 Popularity of Airbnb

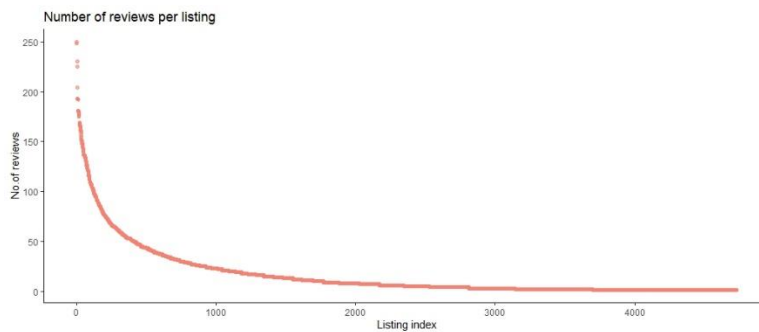


#### 4.1.5 Distribution of reviews per listing

Figure 6 showed the frequency of reviews per listing. As seen by the **long tail**, data points were mostly collected from very popular listings and highly engaged guests. Large amount of less popular listings had very few reviews (~10 reviews/listing).

These **sparse ratings** had implications for our collaborative filtering method as ratings were less predictable for most guests and highly sensitive to an individual who loves obscure listings. Thus, we expected a lot of noise in the algorithm output using the review.csv data.

Figure 6 Sparse distribution of ratings



#### 4.1.6 Review sentiments

Reviews were **largely positive** with an average polarity (positive vs negative expression) of 0.837<sup>3</sup> as shown in Table 7.

Table 7 qdap sentimental analysis

| No. of comments | Total words | Avg polarity | std polarity |
|-----------------|-------------|--------------|--------------|
| 77,289          | 319,657     | 0.837        | 0.526        |

Figure 7 showed that the polarity scores were normally distributed with very few negative sentiments (polarity < 0).

As the contrasting word cloud in Figure 8 shows, most compliments revolved around features as discussed in Section 4.1.3, implying that positivity was derived from the matching of expectations from listing descriptions with actual experience. While complaints were related to topics such as poor hygiene, unfunctional facilities and noisy external environments.

<sup>3</sup> Sentiment analysis using qdap package

Figure 7 Distribution of review sentiments

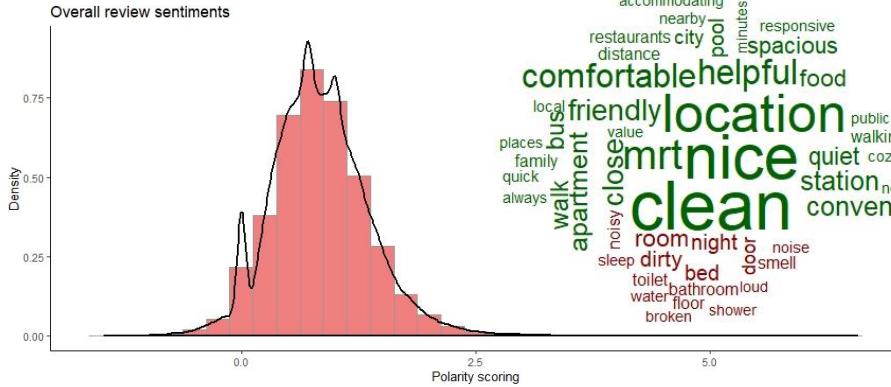


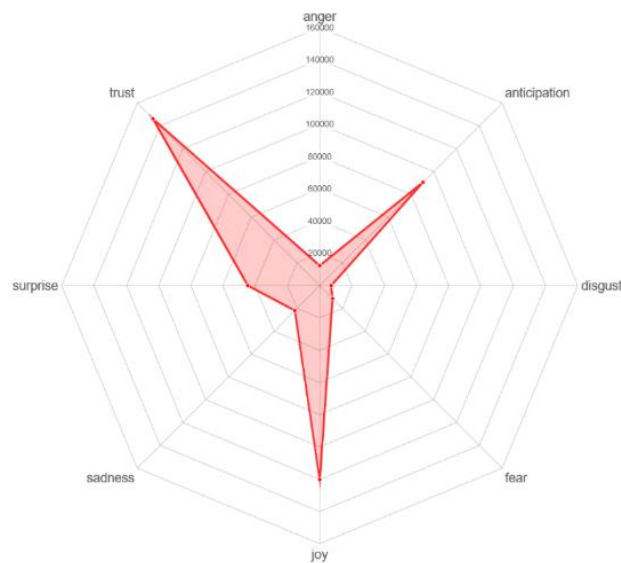
Figure 8 Word cloud for positive vs negative reviews



#### 4.1.7 Review emotions

A further analysis of review emotions using the “NRC” lexicon in Figure 9 showed also positive **reviews with domineering emotional elements** such as trust, anticipation, surprise and joy. Though there are more features, the analysis was less useful for sentimental analysis in comparison to polarity scoring in our case as it was difficult to quantify the various emotions as compared to a straightforward demarcation of positive vs negative.

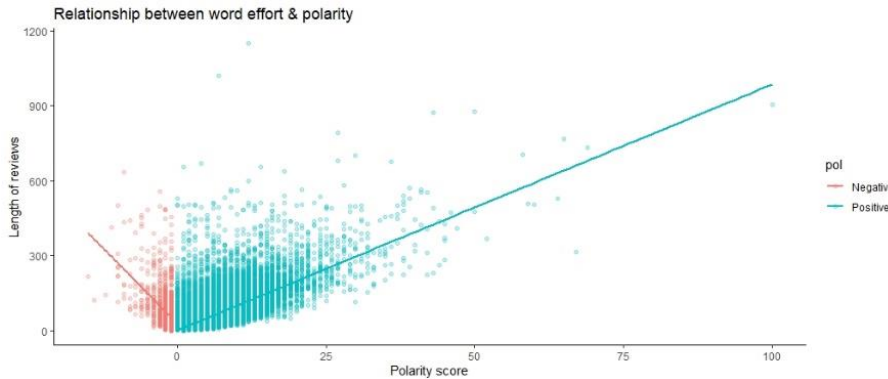
Figure 9 Plutchik's wheel of emotions



#### 4.1.8 Author Effort

Figure 10 showed that guests tend to use more words when they were more passionate, and hence longer reviews. **Length of words were proportionate** to both the negative and positive **polarity scores**. This further emphasized the sparsity problem as discussed in Section 4.1.5.

Figure 10 Length of review vs review sentiments



### 4.3 Implications

The above exploratory analysis revealed several implications on our recommender model summarized as below:

1. **Target audience** – 2 general groups consist of *tourist and locals* who may have specific preference for locations.
2. **Sparsity** – predictive algorithm using user ratings may not work well and produce *biased outcome* favoring the recommendation of popular items in the collaborative filtering process.
3. **Quantify user ratings** – sentimental analysis using *polarity scoring* is more appropriate than emotion classification.
4. **Listing description vs guest reviews** – content-based filtering will produce recommendations that try to make a guess of users' expectations while collaborative filtering will produce recommendations that shows the degree of realization of users' expectations.

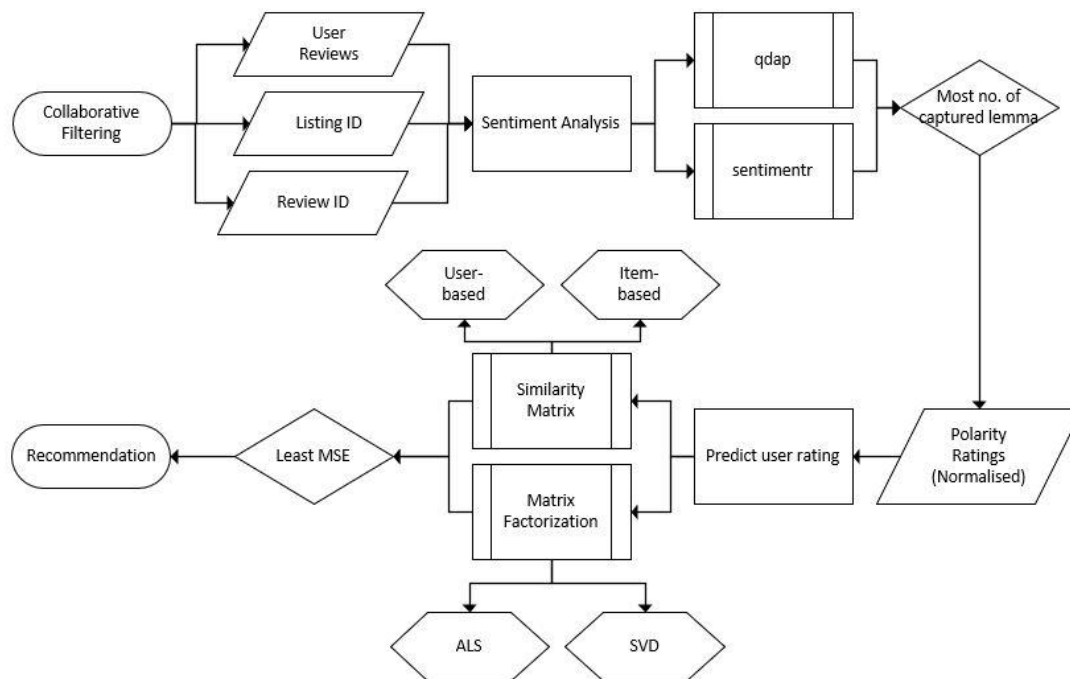
## 5. Collaborative Filtering

Collaborative filtering approach built a predictive model to **predict a user's interest** in a product **based on user's past behaviors** or **similar decisions made by other users**. In Airbnb's case, due to the lack of user scoring, we defined user's past behaviors using **explicit ratings** - Airbnb guests' reviews for each listing.

Firstly, we quantified the reviews using sentimental analysis. This was followed by passing the scoring matrix through several algorithms to predict user's ratings for listings which they did not reviewed on. Finally, recommendation was made using the best predictive algorithm.

The methodology process flow for collaborative filtering was summarized in Figure 11.

Figure 11 Collaborative filtering process flow



### 5.1 Predict User Ratings

#### 5.1.1 Sentiment Analysis

We adopted 3 approaches for the sentiment analysis and reached the following conclusion:

- 1) The final calculation of the polarity score was done using the **default “sentimentr” package dictionary (Jokers & Rinker)** → best method in terms of proportion of words captured (15%, Table 8) due to the inclusion of valence shifters and larger number of build-in lexicons (Table 9)<sup>4</sup>

<sup>4</sup> J&R dictionary is an expansion based on Hu & Liu's that is used as the default dictionary in the “qdap” package

- 2) The polarity score varied from -1 to 1 and we **normalized it to a value of 1 to 5** (Table 10) to make it a more representative rating of how guests would rate each listing in real life situations<sup>5</sup>

Table 8 Comparison of “qdap” and “sentimentr” based on % of total words captured

| Method                   | qdap                   | sentimentr |                  |
|--------------------------|------------------------|------------|------------------|
| Lexicon                  | Hu & Liu               | Hu & Liu   | Jockers & Rinker |
| Positive                 | 329,151                | 338,446    | 425,574          |
| Negative                 | 30,117                 | 33,352     | 54,947           |
| Total words (per review) | 3,193,657 <sup>6</sup> | 3,196,232  | 3,196,155        |
| % captured               | 11%                    | 12%        | 15%              |

Table 9 Comparison of “qdap” and “sentimentr” based on unique words captured

| Method   | qdap     | sentimentr |                  |
|----------|----------|------------|------------------|
| Lexicon  | Hu & Liu | Hu & Liu   | Jockers & Rinker |
| Positive | 1,115    | 1,140      | 2,055            |
| Negative | 1,379    | 1,420      | 2,103            |
| Sum      | 2,494    | 2,560      | 4,158            |

Table 10 Polarity score normalization

| Normalized score | 1                   | 2        | 3       | 4                  | 5             |
|------------------|---------------------|----------|---------|--------------------|---------------|
| Polarity score   | -1 to 0 (exclusive) |          | 0       | 0 (exclusive) to 1 |               |
| Interpretation   | Very Negative       | Negative | Neutral | Positive           | Very Positive |

### 5.1.2 Similarity Matrix Algorithms - KNN

To develop the similarity matrix, we adopted the K nearest neighbor (KNN) approach with both user-based and item-based methodologies. The objective was to predict user ratings for listings which the user did not actually rated on by:

- 1) **User-based** – using the *weighted average ratings* of similar users (i.e. gave similar ratings to common listings) to estimate the rating of the queried user
- 2) **Item-based** – using *similarity score between listings* and previous ratings by the queried user to estimate the rating of the queried user

<sup>5</sup> For users who have reviewed more than once for the same listing, we took the average of normalized polarity score for his reviews as the final rating. There were 821 such cases.

<sup>6</sup> There was a difference in the way that qdap and sentimentr count as a word where a special punctuation symbol is concerned such as apostrophe and backslash, among others. For example, 1' was counted as a word in qdap but ignored in sentimentr. Hence the difference of approximately 2.5k in total word count.

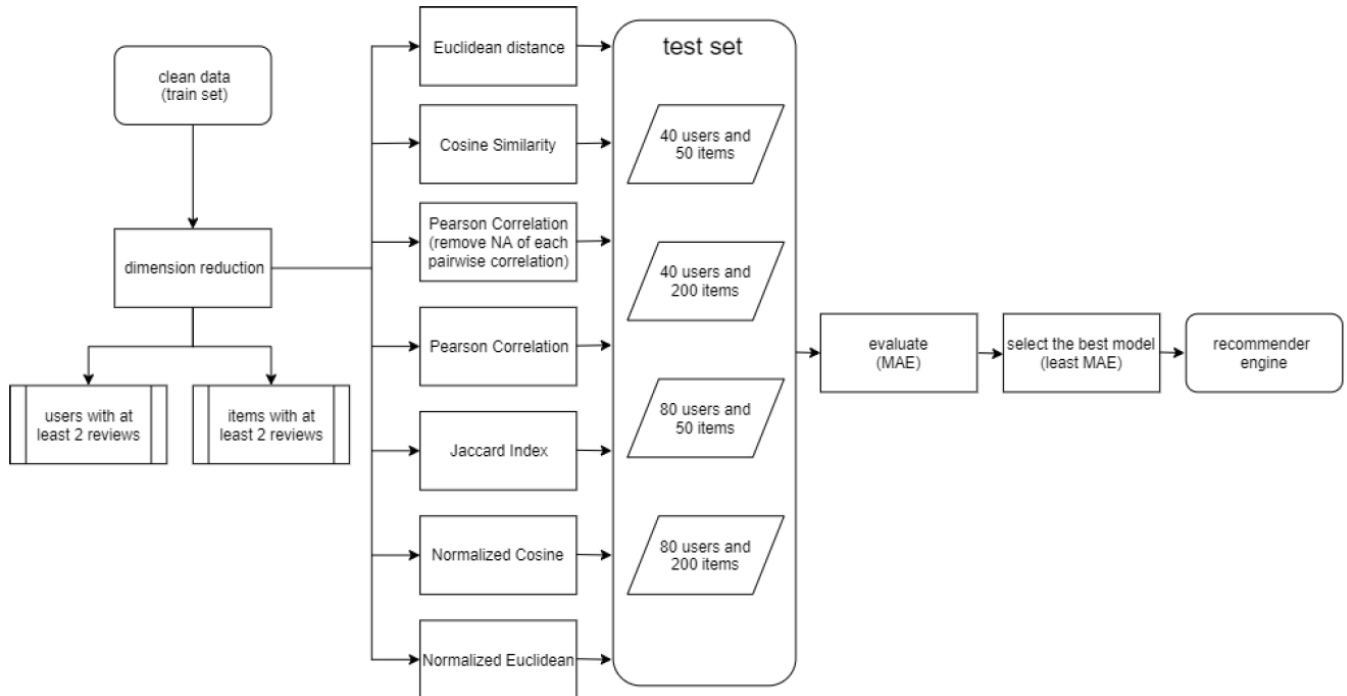
The iteration process described below was also illustrated in Figure 12:

We split the dataset into train and test set. The training set was passed through several algorithms (with custom R functions)<sup>7</sup> and the test set was used to measure the mean average error (MAE) with the combination of following test sets of users and items:

- 1) Number of test set subjects (i.e. user) - 40 vs 80
- 2) Number of matrix items being test (i.e. listing) - 50 vs 200

We then evaluated the outcome for data sparsity. The first iteration produced a sparse matrix, which we did dimension reduction by eliminating inactive users (reviews < 2) and inactive listings (reviews < 2). We arrived at a matrix of 3635 active users and 2425 active listings.

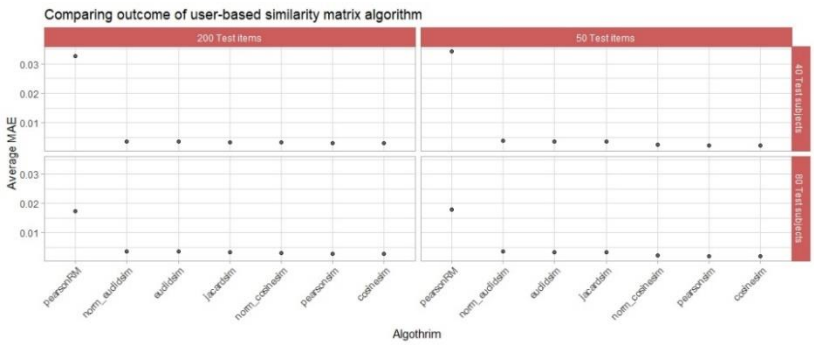
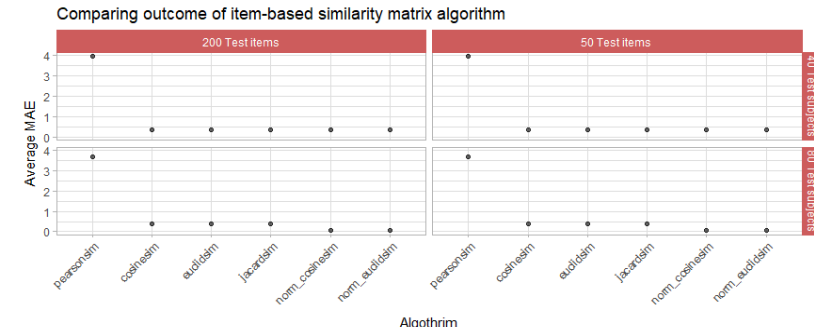
Figure 12 KNN iteration process



<sup>7</sup> See Appendix 1: Algorithms Description for a description of the algorithms used in Table 11

Findings for the iteration process were summarized in Table 11.

Table 11 Comparing user-based and item-based filtering approaches

| Algorithms | Euclidean distance   | Cosine similarity | Pearson correlation | Pearson correlation (Remove NA) | Jaccard index | Normalised Cosine | Normalised Euclidean |
|------------|--|-------------------|---------------------|---------------------------------|---------------|-------------------|----------------------|
| User-based | 0  | 0                 | 0                   | 0                               | 0             | 0                 | 0                    |
|            |  <p>Comparing outcome of user-based similarity matrix algorithm</p> <p>Evaluation: <u>Cosine similarity</u> consistently gives the least MAE under all parameters<sup>8</sup></p>  |                   |                     |                                 |               |                   |                      |
| Algorithms | Euclidean distance   | Cosine similarity | Pearson correlation | Pearson correlation (remove NA) | Jaccard index | Normalised Cosine | Normalised Euclidean |
| Item-based | 0  | 0                 | 0                   | X                               | 0             | 0                 | 0                    |
|            |  <p>Comparing outcome of item-based similarity matrix algorithm</p> <p>Evaluation: <u>Normalised Euclidean</u> consistently gives the least MAE under all parameters</p> <p>Note: Pearson Correlation in this case is not meaningful for comparing item-item similarity</p> |                   |                     |                                 |               |                   |                      |

<sup>8</sup> Pearson correlation gave the same output with Cosine similarity on all test scenarios but the later was preferred as the former output (correlation) might not be applicable since rows with missing values were omitted from the calculation of vector correlations



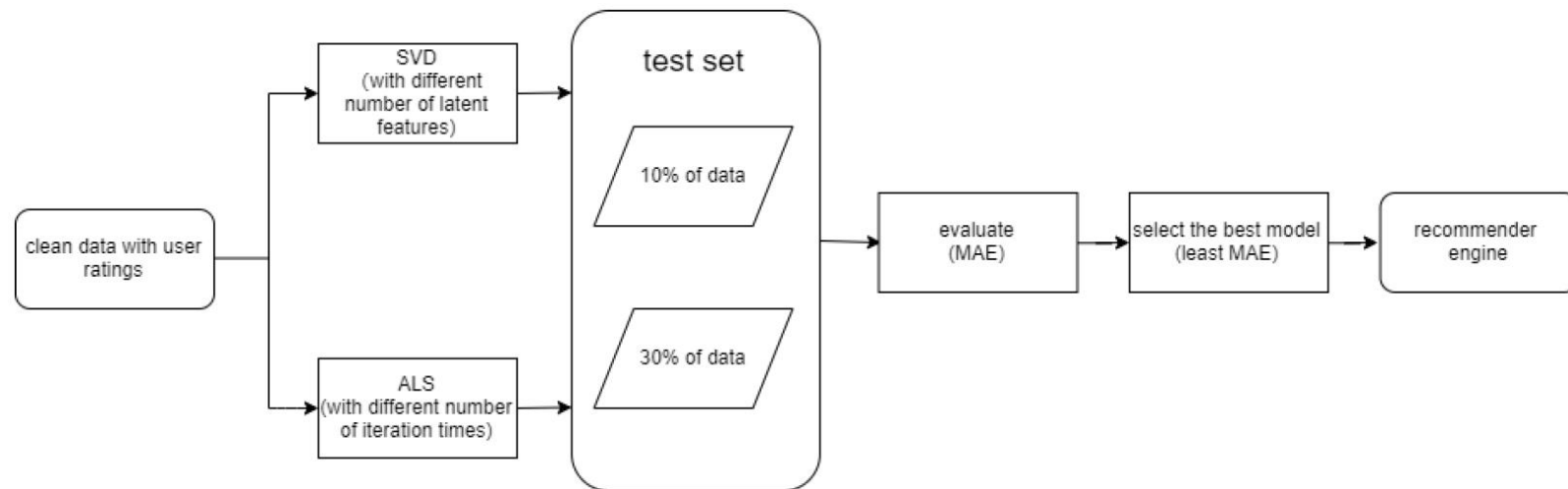
### 5.1.3 Matrix Factorization

This method found **user and items dependencies** in the matrix with latent factors (i.e. features) by **minimizing the root mean square error (RMSE)** to approximate the rating of the queried user. We used 2 approaches – alternative least square (ALS) and singular value decomposition (SVD) - to factorize the matrix.

The iteration process described below was also illustrated in Figure 13:

We split the dataset into 2 sets of train and test set in the ratio 9:1 and 7:3 respectively. The training set was passed through ALS and SVD algorithms with various number of maximum iterations and the test set was used to measure the mean average error (MAE).

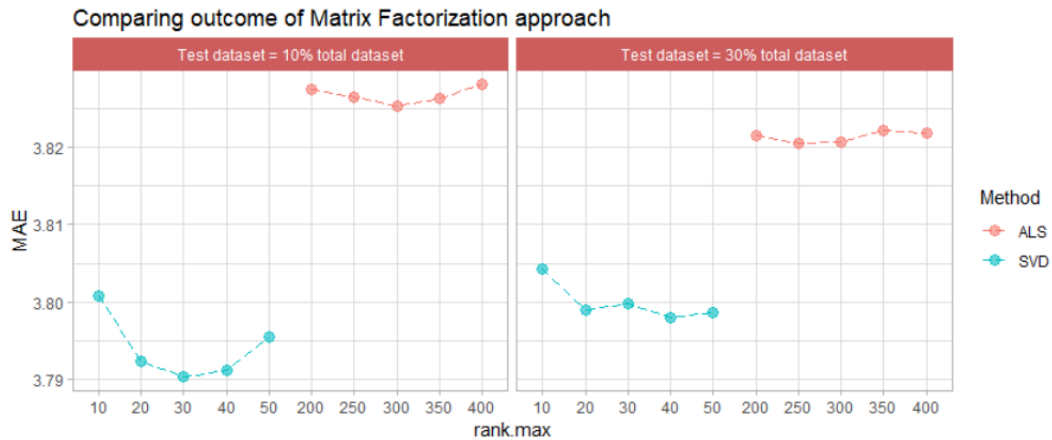
Figure 13 Matrix factorization iteration process



Findings for the iteration process were as below with reference to Figure 14:

- 1) It took **less iterations for SVD** than ALS to converge at the least MAE.
- 2) **SVD produce better prediction result than ALS**, however the outcome seemed to converge as the test set gets larger. It was possible that **ALS would perform better for larger data sets**.

Figure 14 Iteration outcome for Matrix factorization



#### 5.1.4 Evaluating Methodologies

Comparing the KNN and Matrix factorization methods, we arrived at the following conclusions with reference to Table 12:

- 1) KNN – User-based approach produced better outcome (smaller MAE) but it had **scalability issue** with **low computation efficiency**.
- 2) Matrix factorization – SVD approach produced better outcome (smaller MAE) but it was **not definitive to conclude** that SVD was better.
- 3) Matrix factorization method had **low bias and high variance**, in contrast to KNN method which had **high bias and low variance**.
- 4) Nonetheless, Matrix factorization was a better approach among the two as the model was closer to real life situation whereby sparsity problem was common.

Table 12 Criterion to evaluate collaborative filtering approaches

| Assessing Criterion  |            | Hardware                 |                       | Business implications    |               |   |
|----------------------|------------|--------------------------|-----------------------|--------------------------|---------------|---|
|                      |            | Computational efficiency |                       | Avg Min MAE <sup>9</sup> | Valid dataset |   |
|                      |            |                          | Avg time per rank.max |                          |               | Comments  |
| KNN                  | User-based | Poor                     | 2 hours               | 0.002                    | Small         | Bias model → favours listings with many reviews, not able to resolve long-tail problem              |
|                      | Item-Based | Moderate                 | 20 min                | 0.045                    | Small         | Cold-start problem → generation of weak recommendations   |
| Matrix Factorization | ALS        | Good                     | 8 min                 | 3.825                    | Full          | Better prediction → Model learns to factorize rating matrix into user and listings representations  |
|                      | SVD        | Very good                | 5 min                 | 3.79                     | Full          | Address long tail problem → represent latent features equally for popular and less popular listings |

<sup>9</sup> We could not compare the MAE between KNN and Matrix factorization because KNN was ran using a subset of the total data while full data (more sparse) was used to run the Matrix factorization algorithms.

## 5.2 Evaluating Recommendations

Based on the conclusion in Section 5.1.4, we could not decide between SVD and ALS by looking at the MAE alone. We would select the best algorithm between ALS and SVD by testing the recommender outcome with the set up in Table 13 based on the following criteria (which is aligned with our business objectives mentioned in Executive Summary):

- 1) **Accuracy** – How well the recommendation fit user’s preference based on past history
- 2) **Variety** – Are the recommendations spread over a good range of products including those that are popular and not so popular (long-tail)

Table 13 ALS and SVD model test

| Assessing Criterion                                    | Accuracy                   |       | Variety                  |
|--|----------------------------|-------|--------------------------|
| ALS<br>(no. of iterations equals to 250)               | 20% test set (13765 users) | 23%   | 58.1%<br>(4596 listings) |
|  | 40% test set (27530 users) | 22.7% |                          |
| SVD<br>(dimension of the singular values equals to 30) | 20% test set (13765 users) | 23.1% | 55.6%<br>(4397 listings) |
|  | 40% test set (27530 users) | 22.7% |                          |

### 5.2.1 Assumptions

- 1) Referring to Figure 14, we used the **best parameter setting** to test for ALS and SVD (i.e. rank.max that gives the minimum MAE for each model)
- 2) Dataset is split into 2 sets (20% and 40% test set) to **test for model consistency** (i.e. variance)
- 3) **Threshold chosen for testing is 4** → when the predicted rating score >4, assume that the user likes the listing (3 = neutral, see Table 10 for rating description)
- 4) True positive value (TP) → prediction >= 4 and actual rating for this accommodation is >= 4, indicating that we correctly recommend a listing that the user likes.
- 5) Variety is calculated by the number of listings that are recommended (top three) to any of the users divided by the total number of the listing (7,909).

### 5.2.2 Outcome Interpretation

- 1) **Average accuracy** for both models is **a little past 22%**. This is largely due to the **sparsity of data**. Referring to Figure 6, nearly two thirds of listings have only one review and the same goes for users. Since the models were tested with the whole dataset, a low accuracy is expected.
- 2) Given almost the same accuracy, **ALS is a better algorithm** in terms of **variety** as it recommends 201 more listings than that recommended by SVD.

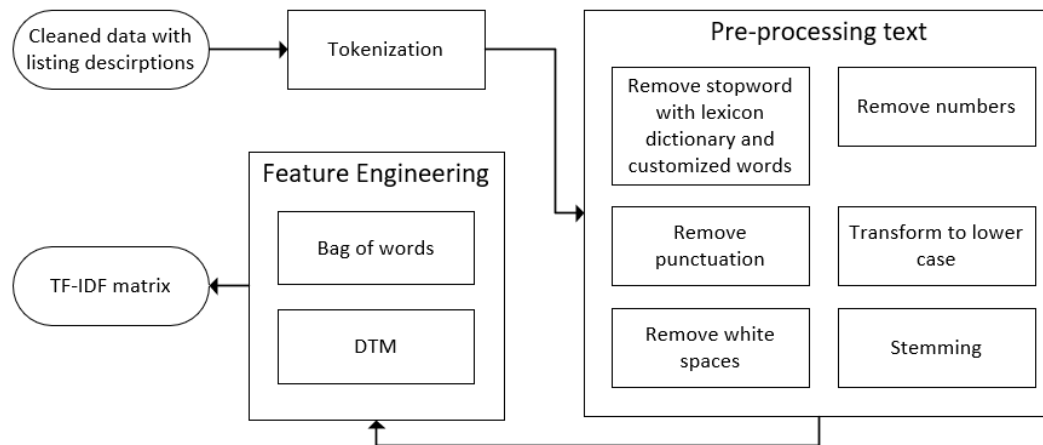
## 6. Content-based Filtering

Content-based filtering approach **utilized discrete characteristics** of a product to recommend items with similar properties. In Airbnb's case, we defined the discrete characteristic as the **description of each listing**. Description contained information about the listing such as location, accessibility, and facilities, among others. Listing with similar descriptions would be recommended to the guest browsing Airbnb's website.

### 6.1 Making up Corpus Object

The tokenizing and cleaning steps to make a document term matrix (DTM) was shown as below:

Figure 15 DTM process flow



### 6.2 Document Similarity Algorithms

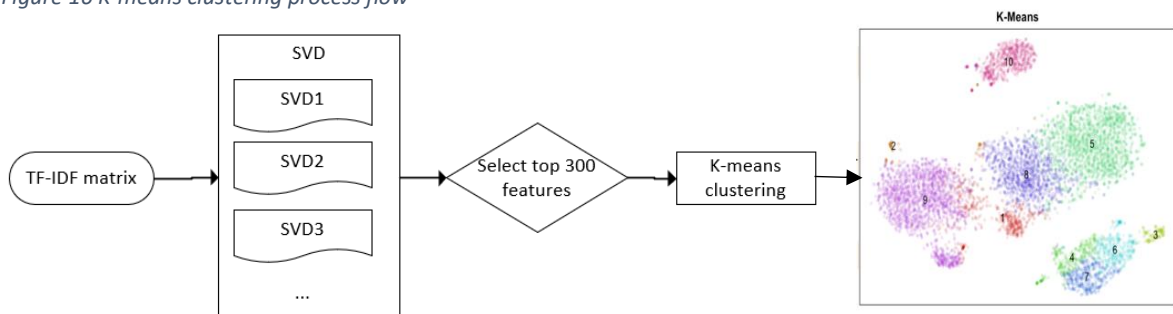
#### 6.2.1 Cosine Similarity

Cosine similarity **measured the cosine of the angle** between word counts in two documents, which were in our case the listing descriptions. The smaller the angle, the higher the similarity, irrespective of the size of the document.

#### 6.2.2 K-means Clustering

We used K-means clustering to **group similar documents into clusters** by measuring the **Euclidean distance** between the word counts. Singular value decomposition (SVD) was performed as a prior step to reduce the DTM dimension to overcome data sparsity problem. The process flow was shown in Figure 16.

Figure 16 K-means clustering process flow



### 6.2.3 LDA Topic Modelling

The LDA process described the model by the **topic matrix and the hyperparameter** for topic-distribution of documents and could therefore be ignored to compute the likelihood of unseen documents.

To derive the number of topics, we used:

- 1) Log-likelihood  $L(W)$
- 2) The perplexity of held-out documents

$$\mathcal{L}(w) = \log p(w|\Phi, \alpha) = \sum_d \log p(w_d|\Phi, \alpha).$$

equation 1. Calculate log-likelihood <sup>v</sup>

$$\text{perplexity}(\text{test set } w) = \exp\left\{-\frac{\mathcal{L}(w)}{\text{count of tokens}}\right\}$$

equation 2. Calculate perplexity <sup>v</sup>

How the recommendation engine would work on topic modelling was as depicted in Figure 17.

Figure 17 Topic modelling process flow

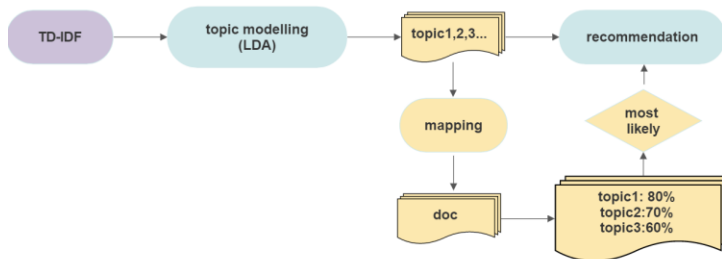
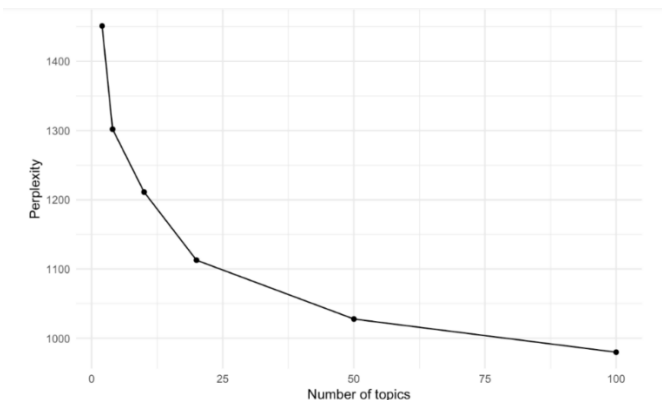


Figure 18 suggested that topic = 20 was the optimal point as the gradient of the curve reduced after this point. However, considering that in actual business situation, too many topics could split the dataset in an unbalanced way, causing bias in each topic. Therefore, we **chose 8 as the final number of topics**. Classification of the topics included content discussed in Section 4.1.3, such as access to transport, presence of shopping mall and availability of facilities etc. (see Appendix 2: Topics Description for full list of topics).

Figure 18 Perplexity value per number of topics

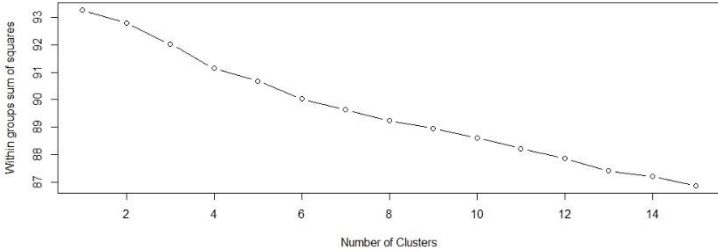
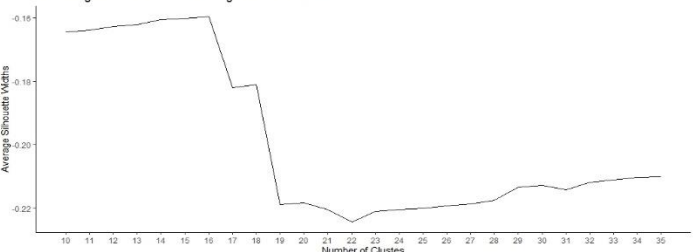


### 6.2.4 Evaluating Methodologies

Analyzing the above algorithms, we concluded that there were at present no appropriate measures to test the accuracy of Cosine similarity and Topic modelling since our dataset was not tagged.

We could only evaluate k-means clustering using the strategies below in Table 14 to assess how well an observation was classified.

Table 14 Evaluation of k-means clustering

| Strategy              | Within Sum of Squares  | Average Silhouette widths   |
|-----------------------|--|---|
| <b>Interpretation</b> | Measures of the variability of the observations within each cluster                    | Measures dissimilarity between clusters   |
| <b>Measurement</b>    | Small value -> more compact cluster  | -1: point is closer to neighboring cluster<br>0: point belongs to neither clusters<br>1: point is well matched to its own cluster |
| <b>Visualization</b>  |     |   |
| <b>Observation</b>    | There is no clear elbow in the scree plot indicating a change in gradient of the curve | The best score of silhouette width is -0.16   |
| <b>Conclusion</b>     | <b>Clustering is not an appropriate algorithm</b> for the Airbnb listing dataset       |   |

## 7. Limitations and Conclusion

### 7.1 Limitations

Limitations for our recommender system were summarized below:

- 1) **Choosing the best sentiment analysis package** – Due to time and capacity restrictions, we have chosen to use a readily available package to capture the polarity in the reviews. As such, context *specific words would be missed out*, such as internet slangs and custom abbreviations or colloquial language expressions.
- 2) **Hardware restrictions** – Our machine could not run user-based and item-based similarity matrix without dimension reduction due to data sparsity. As a result, though they may be better algorithms, we must drop them in favor of matrix factorization due to dataset bias. Therefore, *we were not able to assess all the algorithms* for a collaborative filtering system, and *hence our conclusion of ALS may not have been the best* algorithm actually.
- 3) **Testing capabilities** – We were not able to test the outcome of our recommender system adequately particularly for content-based algorithms due to the *lack of tagging* for unsupervised machine learning.

### 7.2 Further Improvements

The system can be further improved by:

- 1) Developing a **customized dictionary** to classify the review sentiments.
- 2) **Encourage users to write reviews** for every booking to reduce the data sparsity problem.
- 3) **Perform A/B testing** to collect user feedbacks on the recommender machine and fine tune the algorithms.

### 7.3 Implementation Considerations

The success of our recommendation engine would be measured by:

- 1) **Not pushing listings** in which the user **previously booked**
  - repeated listings are filtered out after running the models
- 2) **Include less popular listings** (on the long tail)
  - tested the recommendations to ensure that it picks up less popular listings (with only 1 or 2 reviews)
- 3) Allows **filtering for locations**
  - Listings are concentrated in the central area. By allowing users to filter by location, it is easier for them to find listings in the less popular locations should they have such needs
- 4) Allows **filtering for price**
  - mimicking real life situations

## Appendix 1: Algorithms Description

| Algorithms  | Description  |
|---|--|
| <b>Euclidean distance</b>   | The Euclidean distance between two points is the length of the line segment connecting them.   |
| <b>Cosine similarity</b>  | Cosine similarity measures the cosine of the angle between two non-zero vectors of an inner product space.   |
| <b>Pearson correlation</b> ( <i>remove NA during the calculation of each pairwise correlation</i> ) | Only remove NA during the calculation of each pairwise Pearson correlation. (the Pearson user means may be different)  |
| <b>Pearson correlation</b>  | Have all cases with missing values removed in the first place, then calculate the Pearson correlation.   |
| <b>Jaccard index</b>  | The Jaccard similarity index compares members for two sets to see which members are shared and which are distinct.   |
| <b>Normalized Cosine</b>  | Adjusted cosine similarity measure is a modified form of vector-based similarity where we take into the fact that different users have different ratings schemes by subtracting users mean rating from their individual ratings. |
| <b>Normalized Euclidean</b>   | Normalized Euclidean measure is a modified form of Euclidean distance where we eliminate bias by subtracting users mean rating from their individual ratings.  |



## Appendix 2: Topics Description

| Topics  | Descriptions   |
|---------|--|
| Topic 1 | The taste of the niche, but the furniture and transportation are convenient              |
| Topic 2 | Convenient neighborhood near Chinatown, with spacious street and full shower             |
| Topic 3 | Accessible transportation near the food court in the east                                |
| Topic 4 | Locations near Marina Bay, easy to drive from airport. There are many parks near blocks. |
| Topic 5 | Many shopping malls, located around Orchard  |
| Topic 6 | Great community infrastructure, such as spacious swimming pool and free gym              |
| Topic 7 | The facilities in the house are relatively complete, washing machine, dryer, kitchenware |
| Topic 8 | Convenient public transportation in CBD area   |