Are Customers Heard Enough?
--Auto Streaming of Customer Feedback Using Text Mining Techniques

Team Hugs for Bugs

Chen Yanzhu	A0206503U
Li Yonglei	A0103467J
Song Zhiying	A0206508J
Xie Zhifang	A0077221M
Yang Yaci	A0102761R

Auto Streaming of Marina Bay Sands Customer Feedback Using Text Mining Techniques

1. Introduction

Many companies live or die on customer feedback, especially in today's digital era when reviews are so easily posted by one and read by another. This is particularly true for hotel business. Before hitting the "Book Now" button, customers will screen through booking sites such as Expedia, Booking.com and TripAdvisor to check out the ratings and comments. Guest review is often a make-or-break factor for potential customers.

Hotel analytics is nothing new. There are many studies available in the research database focusing on the hotel analysis and using analytical approaches such as sentiment analysis and topic modelling. Unfortunately, none of these models alone is practical and insightful enough to make use of customer reviews to improve service and increase business volume as a result.

Our team will propose a fine-grained end-to-end analytics framework using Marina Bay Sands ("MBS") Hotel Singapore as an example, including conducting hotel review analysis to hotel department level as well as providing information visualizations that adds additional insights.

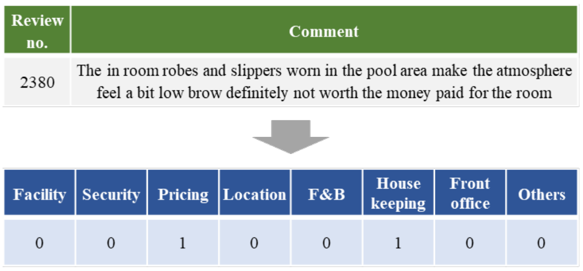
1.1 Problem Statement

In most of the third-party booking portals, although customers are given the options to provide ratings for each individual type of experiences (e.g. facilities, cleanliness, food and beverage etc.), text boxes for input of rating details are not provided after every rating question. Instead, a single text box field is given to customers to describe their overall experience per review. While it saves the customer from getting bored with the survey and phoning it in for the sake of getting through the form, this method leaves a lot of work to the customer service of the hotel to analyse the information, split them into different categories, send to the relevant departments for feedback, and consolidate the answers from the departments (if any) before responding to a review, especially for the negative ones (refer to Figure 1 below for sample review). Therefore, we aim to develop a framework that can help relevant departments in MBS to understand customers' main concerns during their stay based on reviews so that proper measures can be developed.

1.2 Project Methodology

In the context of machine learning, this is a supervised text classification problem. In order to help each department to improve their services from customer feedback, only negative comments (or texts in the "cons" column of booking.com) are required. We scraped this data using Python BeautifulSoup package to create our dataset. As the original dataset doesn't come with a target value, our team decided to label each review manually into one or more relevant departments (i.e. *"Facility", "Security", "Pricing", "Location", "Housekeeping", "F&B", "Front Office" and "Others"*) for model training purposes (refer to Exhibit A below). Since one piece of review can involve more than one department, instead of treating it as a multi-class classification problem - which will only stream every review into one department, we adopted a binary classification approach for each category, i.e. each review is assessed against all departments to determine whether or not it's relevant to each particular department (i.e. 0 corresponds to irrelevant, 1 corresponds to relevant). In this way, our selected models are able to give binary predictions for each department from the input data.

Exhibit A



Review no.	Comment
2380	The in room robes and slippers worn in the pool area make the atmosphere feel a bit low brow definitely not worth the money paid for the room

Facility	Security	Pricing	Location	F&B	House keeping	Front office	Others
0	0	1	0	0	1	0	0

Subsequently, we pre-processed the text data such as removing punctuations and stop words to reduce noise and transfer data from human language to machine-readable format.

This is followed by resampling data using Random Oversampling and Random Undersampling techniques to treat the imbalance of the data. Feature extractions such as taking word counts and TF-IDF counts were then applied, followed by training different models. Seven (7) models are selected as the best model to classify each category.

2. Dataset

2.1 Data Description

Our dataset contains 5919 negative customer reviews scraped from the "cons" column of MBS review site on booking.com.

2.2 Data Preprocessing

The modelling performance highly relies on the quality of text preprocessing. The following data cleaning steps were taken in sequential order:

2.2.1 Data Cleaning

As a first step, we converted the text into lower-case. Then we removed punctuation, non-alphabetic characters from the original text in order. Tokenisation and stopword removal were performed using the NLTK library in order to reduce training time and have the models focused on the words that define the meaning of the text. To further clean the sentences, we used NLTK Snowball to stem each word. This step was exempted for topic modelling which involves lemmatisation and POS filtering.

2.2.2 Feature Engineering

The next step is feature engineering which requires text comments to be converted into different feature vectors. In this project, count vectors and TF-IDF vectors are used to represent the text in comments. Count Vector is a matrix that represents the word frequency in their comments while TF-IDF vector represents the relative frequency of the words in the comments and the whole corpus. In this case, three levels of TF-IDF matrices are used: word level, n-gram level and n-character level, which represent the matrix of TF-IDF scores of every word in different comments, the matrix of TF-IDF scores of the combination of n words in different comments and the matrix of TF-IDF scores of the combination of n characters in different comments respectively.

2.2.3 Data Resampling

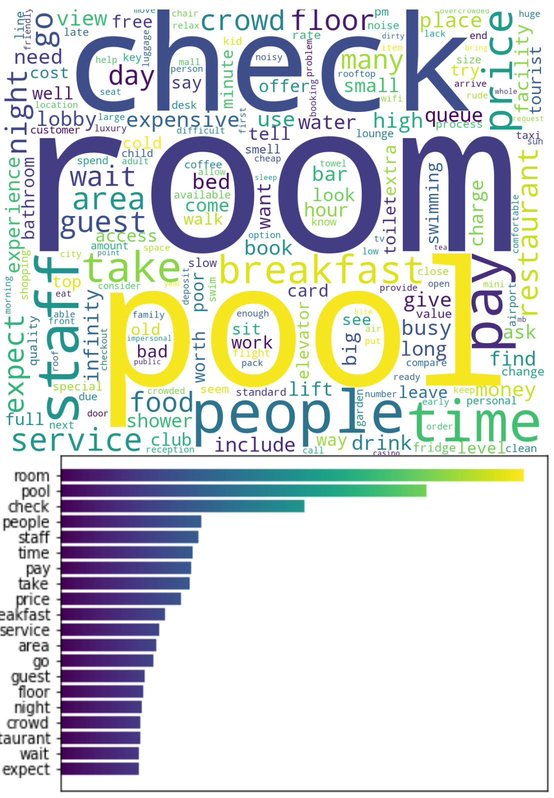
Since each corpus is highly imbalanced, we draw repeated samples using oversampling (i.e. adding more examples from the minority class) and undersampling (i.e. removing samples from the majority class) techniques for better model performance.

2.3 Data Exploration

2.3.1 Word Cloud

As the first step of data exploration, a word cloud (Exhibit B) for the entire corpus of the negative review data was generated using the Wordcloud package in Python.

Exhibit B

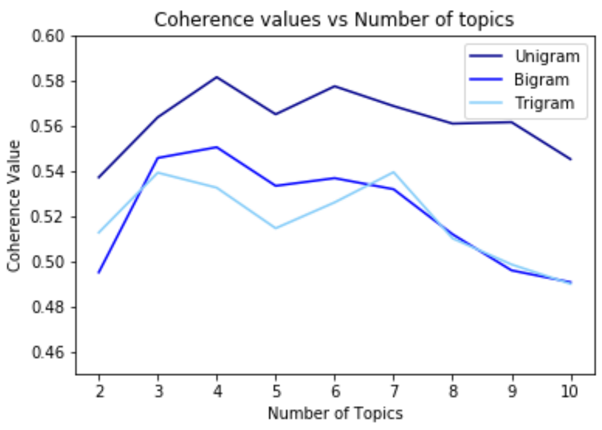


Word Cloud is capable of telling the importance of individual words. In order to have a better understanding of how these important words link to one another and form topics or themes , topic modelling was explored using the Gensim package in Python.

2.3.2 Topic Modelling

Topic modelling is widely adopted to discover the hidden themes of a collection of documents. It is essentially a clustering process which groups the documents into a fixed set of topics. Latent Dirichlet Allocation (LDA) is a popular algorithm for topic modelling with implementations in Python's Gensim Package. Mallet, the Java topic modelling toolkit with an efficient implementation of the LDA, was selected due to its faster execution speed and better performance in topic segregation. The objective is to extract coherent, well segregated and meaningful topics from the negative reviews to understand customers' top concerns, thus the following criteria were used to assess the modelling performance jointly: 1. coherence score, a measure of the degree of semantic similarity between high scoring words in a topic; 2. topic segregation visualised through the pyLDAvis plot (using pyLDAvis package); 3. semantic meaningfulness based on the top 10 keywords of each topic.

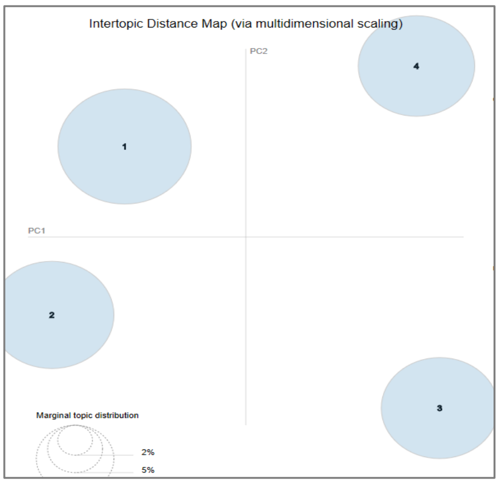
Since topic modelling works on the basis of a pre-defined number of topics, the optimal number of topics need to be identified. After taking the preliminary text preprocessing steps described in Section 2.2.1, bigram and trigram models were created using the Gensim package to complement the original unigram model. Lemmatisation and Part-of-speech (POS) filtering was applied to each model using the spaCy library, only proper nouns, nouns, verbs and adjectives were retained from the lemmatised text. A few additional common terms which add no value to topic modelling (e.g. "hotel", "tower", "think", "feel") were removed as stopwords. The three scenarios of the feature space were then experimented and evaluated in terms of model coherence and topic segregation over a range of number of topics. TF-IDF vectorisation is not necessary in this case, as LDA is a probabilistic model that tries to estimate probability distributions for topics in documents and words in topics (Blei et al, 2010).The below graph demonstrates how the model coherence score varies over the number of topics for each N-gram model.

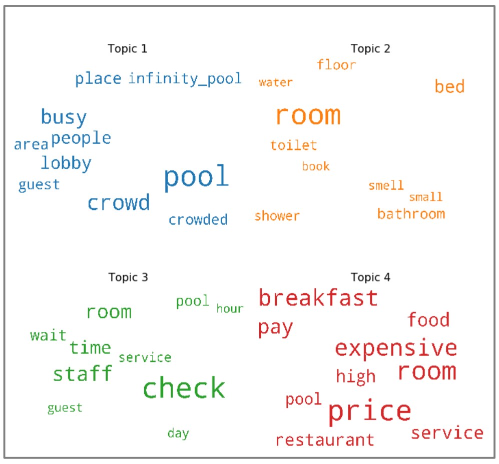


The unigram 4-topic model and the bigram 4-topic model give the highest coherence score (unigram: 0.58; bigram: 0.55) for their respective N-gram scenarios. From the pyLDAvis plot, the bigram model outperforms the unigram model as it generates well segregated topic clusters, whereas the unigram model entails significant overlapping between 2 topics (See Appendix A).

Next, hyperparameter tuning was performed on the alpha parameter of the bigram 4-topic model. Alpha is a representation of document-topic density, a higher alpha results in more topics per document. It was found out with alpha chosen to be 0.5, the LDA model gives the best performance with coherence score of 0.56. The topic segregation and the top 10 keywords per topic are shown as below.

Exhibit C



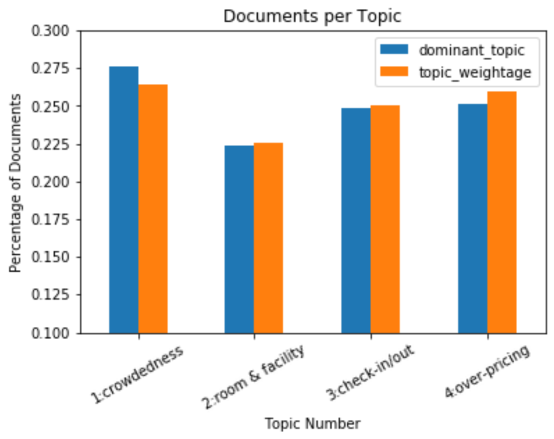


Based on the top 10 keywords listed, the themes of the four topics can be inferred as: 1. crowdedness (at the infinity pool and the lobby); 2. room and facility; 3. check-in/check-out; 4.over-pricing (especially for food & beverages).

Furthermore, alternative topic modelling methods including the baseline LDA model (without Mallet) and Non-negative Matrix Factorization (NMF) with TF-IDF were explored and served as benchmarks. The baseline LDA model gives a slightly lower coherence score of 0.51 and similar results for topic segregation and top 10 keywords per topic (see Appendix B). The NMF model, despite a low coherence score of 0.47, also generates four topics with similar themes (see Appendix C). This validates the topic modelling results for the selected best-performing model.

The below graph (Exhibit D) shows the division of the reviews into the four topics by: a. dominant topic; b. topic weightage. With topic modelling, the customers' concerns, as well as the effectiveness of corresponding solutions can be monitored efficiently over time.

Exhibit D

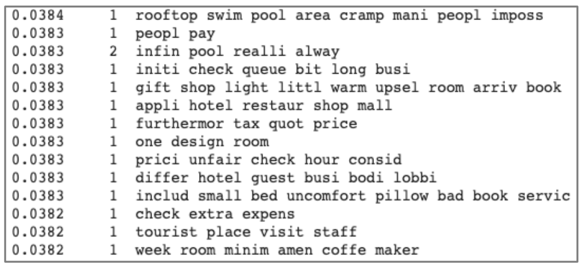


2.2.3 PyTextRank for Phrase Extraction

PyTextRank is a Python implementation of TextRank as a spaCy pipeline extension, which is a popular tool to extract the top-ranked phrases from text documents. Inspired by Google's PageRank, PyTextRank was introduced in 2004 which employs graph algorithms, to be specific, eigenvector centrality to represent links among the candidate phrases. The lemma graph constructed by nouns, verbs and adjective vertices can be enriched and phrase ranking results can be improved using various ways such as semantic relations.

A key benefit of using PyTextRank compared to other keyword extraction models is the number of words to extract each time is not limited, which provides significantly more information content. Outputs of the model are rank of the phrase, count of phrase occurrence and the corresponding phrase. The higher the rank, the more important is the particular phrase to the text document. In our case study, some of the top ranked negative review proposed by PyTextRank include phrases such as "swimming pool area cramp", "check-in queue too long", "pricing unfair", "small bed", "uncomfortable pillow" and "bad booking services" etc (Exhibit E), suggesting improvements to be made on these customer pain points.

Exhibit E


```
0.0384     1  rooftop swim pool area cramp mani peopl imposs
0.0383     1  peopl pay
0.0383     2  infin pool realli alway
0.0383     1  initi check queue bit long busi
0.0383     1  gift shop light littl warm upsel room arriv book
0.0383     1  appli hotel restaur shop mall
0.0383     1  furthermor tax quot price
0.0383     1  one design room
0.0383     1  prici unfair check hour consid
0.0383     1  differ hotel guest busi bodi lobbi
0.0383     1  includ small bed uncomfort pillow bad book servic
0.0382     1  check extra expens
0.0382     1  tourist place visit staff
0.0382     1  week room minim amen coffe maker
```

3. Sentiment Analysis

We intended to use sentiment analysis to distinguish the true negative reviews from the positive reviews that are mistakenly input by users in order to make the auto-comments-classifications process value-added. We first explored a few unsupervised sentiment analysis techniques: 1. VADER (Valence Aware Dictionary and sEntiment Reasoner), a lexicon and rule-based sentiment analysis tool of the NLTK library that returns a compound score between the range of -1 and 1, we deemed reviews with a compound score greater than 0.05 as positive reviews; 2. the Sentiment Analysis tool of the TextBlob library, which returns a polarity score between the range of -1 and 1, similarly a threshold of 0.05 was used to categorise the positive and negative reviews. Since we aim to minimise both the false positive rate and the false negative rate, we decided to use F1 score as the evaluation metric to compare the computed sentiments against the manually labelled sentiments. The two algorithms returned an F1 score of 0.85 and 0.80 respectively, although they are decent scores, they do not add value due to the very low percentage of the positive reviews in the dataset (around 5%).

In light of the impracticability of unsupervised techniques, we explored the use of Logistic Regression model to classify positive and negative reviews. The modelling pipeline consists of the TF-IDF vectorisation, resampling (due to the highly imbalanced dataset) and the classification model. Among Random Oversampling, Random Undersampling and SMOTE (Synthetic Minority Oversampling), SMOTE was chosen as it gave the highest cross validation F1 score on the training set. Then we performed hyperparameter tuning for the modelling pipeline using the test set F1-score as the evaluation metric. The best-performing model gives an F1-score of 0.98 on the test set.

This Logistic Regression model manages to achieve an F1 score close to 1, the good performance on the test set demonstrates that it can be applied for new incoming data and serve as the first stage of the entire auto-comments-classification model pipeline. However, in the context of this project, we could not continue building the model pipeline based on the test set since it is a relatively small dataset for demonstrating the quality of our classification models. Therefore, we decided to develop and implement auto-comments-classification models on the entire dataset (with the original train and test split).

4. Classification Model Selection and Evaluation

To construct an auto-comments-classification model, we fit the training data with different models to find the best model for each department corpus. In this case, we explored 7 different models, including Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, K-nearest Neighbors (KNN), Light Gradient Boosting Model (LightGBM), and Neural Network. We also conducted hyperparameter tuning using GridSearchCV on each model so as to derive the best possible outcome.

To evaluate the model performance, we decided to use F1 score as the metric to compare the performance of selected models and determine the best model for each category. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0. It works well with an imbalanced dataset, stressing the importance of true positive while taking into consideration the false prediction. This is suitable for our current project where distribution of each category is uneven and our priority is to correctly identify the relevant department(s) and label it as "1" and reduce the irrelevant information (labeled as "0") flooded into other departments from each customer review.

The performance of each model measured by F1 score for class 1 is as follows:

Exhibit F

F1 Score (1)	Facility	Security	Pricing	Location	F&B	House-keeping	Front Office	Others
Naive Bayes	0.73	0.28	0.78	0.21	0.50	0.41	0.65	0.60
Logistic Regression	0.74	0	0.83	0.40	0.27	0.13	0.70	0.51
LightGBM	0.87	0.50	0.87	0.43	0.55	0.52	0.76	0.52
SVC	0.75	0.25	0.84	0.40	0.45	0.58	0.74	0.57
KNN	0.65	0.13	0.76	0.15	0.19	0.30	0.49	0.41
Random Forest	0.76	0.17	0.88	0.36	0.55	0.51	0.78	0.60
Neural Network	0.74	0.19	0.84	0.38	0.36	0.52	0.74	0.60

Logistic Regression

Logistic regression, a traditional statistic model, serves as a benchmark for evaluating our models. From the result, it appears that the model has unstable performance in predicting different categories. The F1 score goes as low as 0 for Security and as high as 0.74 for Facility. A possible reason is due to divergent words being learnt to indicate the security problem, thus it is difficult for the model to detect which sentence is truly referring to the security problem and lead to a poor F1 score for this category.

Naive Bayes

Naive Bayes is a supervised learning algorithm that is widely used in the classification problems. In particular, Multinomial Naive Bayes, an algorithm that assumes data follows a multinomial distribution, is one of the classic methods used in text classification. Therefore, we applied Multinomial Naive Bayes to this project. According to the results, Multinomial Naive Bayes performs well in some categories like Facility, Pricing, and Front Office. As for the rest like Security, Location and Others, this model is not able to classify them well. Additionally, the F1 score is much lower than that of other models. Such low F1 scores might result from the small dataset of those 3 corpus.

Random Forest

Random Forest is an ensemble model that is widely used in classification problems. Since it integrates many results from decision trees, it is known to tackle the overfitting problem well. In this project, Random Forest performed very well in some categories, and gave the highest F1 score among all the models we explored. According to the results, it is capable of identifying the comments related to Pricing, Location and Front Office better than other models. However, we also noticed that the F1 score for Security is extremely low. This may be attributed to a small dataset for Security, for which result is not improved even after oversampling effort. The result also implies that Random Forest is not suitable for small dataset.

Support Vector Machine

SVM achieves good performance in many text categorisation tasks as it is able to handle the particular properties of text, including but not limited to high dimensional feature spaces, few irrelevant features (dense concept vector), and sparse instance vectors, given sufficient training data. This is proven in the current project, whereby SVM gives a satisfactory F1 score for the top 2 labels, i.e. 0.68 for Facility & 0.72 for Pricing, however performs poorly even after resampling on the remaining categories with small dataset.

K-Nearest Neighbours

KNN algorithm is used to classify by finding the K nearest matches in training data and then using the label of closest matches by a predefined distance (e.g. Euclidean) to predict. It is recognised as one of the laziest algorithms which does not require the use of training data to perform the classification task. Instead, training data can be used during the testing phase. This has led KNN to becoming one of the most effective methods on the Reuters corpus of newswire stories – a benchmark corpus in text categorization. However, KNN does not seem to be a wise choice in the current project as it gives low F1 scores in predicting all categories, including "top performers" Facility and Pricing. A possible reason is that KNN does not work well on data with high dimensionality, which results in the distance being less representative in identifying similarity.

LightGBM

LightGBM is one of the ensemble algorithms in machine learning. To minimize the errors, the model runs repeatedly until it reaches some validation terms. Due to its fast training speed and better accuracy, LightGBM is explored in this project.

First, to obtain a proper depth of the decision tree, the best value for max_depth parameter was searched over a space from 4 to 8. Then, min_child_samples parameters were mainly tuned to prevent potential overfitting. Tuning of LightGBM model significantly improves the accuracy of the model. It turned out in our study that LightGBM outperformed the rest of the models for 4 out of 8 departments. The highest F1 score derived was 0.87 for predicting Facility. However, for the highly imbalanced Location dataset, even after random oversampling, this model fails to achieve a better F1 score than the current one, which is 0.5.

Deep Learning

Deep learning has gained favour in recent years due to its powerful ability such as capturing complex interactions or nonlinear relationships between the variables. In the hope of leveraging its benefits, a neural network model is explored in this project.

A simple neural network of 2 dense layers is constructed. The hidden layer consists of 5 neurons, using ReLu as the activation function. To prevent any possible overfitting, the dropout rate is a hyperparameter that is optimised using GridSearchCV. From Exhibit F, we can observe that the performance of this neural network is consistently average across all categories, even for highly imbalanced categories - such as reviews for the Security Department, contrasting to 0 score achieved by some simpler models.

The best model for each category is summarised in Appendix D. LightGBM and Random Forest outperform all other models with the following parameters:

LightGBM basic parameters: {'boosting_type': 'gbdt','max_depth': 6, 'reg_alpha': 0.01, 'num_leaves':80, 'objective': 'binary','feature_fraction': 0.8, 'bagging_fraction': 0.8, 'bagging_freq':1, 'learning_rate': 0.1, 'lambda_l2': 0.1,'metric':'binary_error'}

Random Forest basic parameters: {n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None}

5. Conclusion

5.1 Business Application

In general, most of the reviews on booking agencies are read by potential customers of the hotel, but they are rarely well utilised by hotel correspondences. In order for the customers' voice to be better heard and then, in turn, better served, we applied several data exploration techniques, including Word Cloud, Topic Modelling and PyTextRank, to gain a better understanding of the customers' experience and propose an auto-streaming pipeline using different machine learning algorithms which classify relevant feedback to related departments for their future improvement of services.

5.2 Hotel Services Improvement Areas

Overall customer review shows that many customers are unsatisfied with the unfriendly front-office staff, misleading use of fridge, overcrowded infinity pool and room facility malfunctions. These are unacceptable to customers who are charged a high price for the accommodation and services. Some of our suggestions to the hotel management are as follows:

To improve the front office services, hotel management should provide regular training to the staff on ways to deliver a welcoming check-in / check-out service, which drives improved outcomes in guest interaction and higher guest ratings. In addition, to address the "long queue" concern, it's advisable to the hotel to adopt an operational research approach to optimise resource allocation during peak hours.

With regards to customers' complaints on the usage of fridge and mini-bar, we would suggest that the hotel put clear instructions in the fridge area to advise guests on the use and the corresponding charges. Front office staff should also provide this information to the guests during the check-in process.

To make the infinity pool more appealing and deal with the disappointment around the overcrowded pool and towels occupying the chairs, hotel management can arrange more chairs to be set up during peak hours, put up signage to suggest other interesting activities nearby for guests to explore, or limit entry to guests during peak hours.

Regarding facility malfunctions in hotel rooms, we would suggest that the housekeeping department should check the condition of the room facilities in conjunction with the cleaning activities so that issues can be reported to the relevant department to conduct repair work before the next customer checking in.

5.3 Limitation of the Current Study

In this project, we manually assigned each sentence to the corresponding department. To avoid discrepancy, we discussed some ambiguous sentences. The labels, however, may still contain mis-classification due to slightly divergent standards. More rigid business rules are needed to better define each category.

In addition, while we are confident that we have sufficient dataset to support our conclusion, it is conceivable that reviews on a single site (i.e. booking.com) have some sort of invisible bias. To account for that possibility, future studies should also analyse reviews collected on other social media sites.

Given the same dataset, sentiment analysis and the model training could not be integrated into the same pipeline in our current project. However, for any new future data, we propose to filter any noise using sentiment analysis before feeding the real negative comments into the final model for prediction.

5.4 Possible Future Work

In this project, we only analysed comments written in English and discarded other languages for simplicity. In future, it is possible to cover more languages using the same framework proposed in the current study to obtain more insights on the customer experiences and pain points for hotel departments to improve their performance on.

To further improve customer services' productivity in responding to the customers' feedback, it is also possible to apply NLG (Natural Language Generation) techniques to generate automatic reply to the corresponding review.

6. Resources

The code implementation of this report can be found on this link: http://tiny.cc/hugsforbugs.

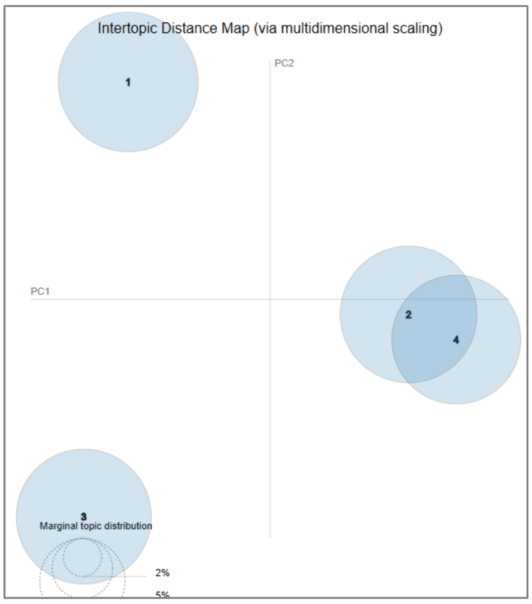
7. References

1. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Paper presented at the *, 1398* 137-142. doi:10.1007/s13928716
2. Guo, G., Guo, G., Wang, H., Wang, H., Bell, D., Bell, D., . . . Greer, K. (2006). Using KNN model for automatic text categorization. *Soft Computing, 10*(5), 423-430. doi:10.1007/s00500-005-0503-y
3. Trstenjak, B., Mikac, S., & Donko, D. (2014). KNN with TF-IDF based framework for text categorization. *Procedia Engineering, 69*, 1356-1364. doi:10.1016/j.proeng.2014.03.129
4. Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into texts. In Lin, D., & Wu, D. (Eds.), Proceedings of EMNLP 2004, pp. 404–411 Barcelona, Spain. Association for Computational Linguistics.
5. Blei, D.M., Ng, A.Y., & Jordan M.L.(2003). Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022
6. Keras Hyperparameter Tuning using Sklearn Pipelines & Grid Search with Cross Validation https://medium.com/@am.benatmane/keras-hyperparameter-tuning-using-sklearn-pipelines-grid-search-with-cross-validation-ccfc74b0ce9f

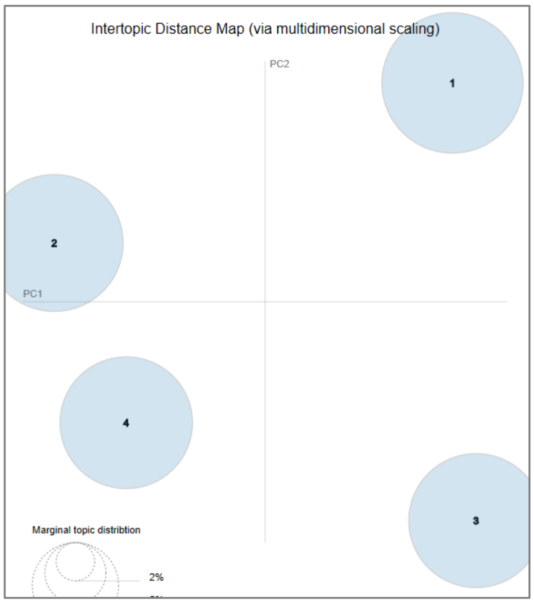
Appendix A

Topic segregation: unigram 4-topic model v.s. Bigram 4-topic model

Unigram 4-topic model

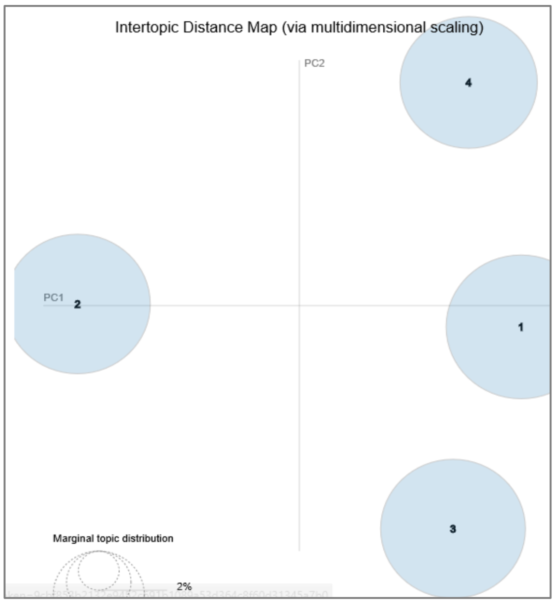


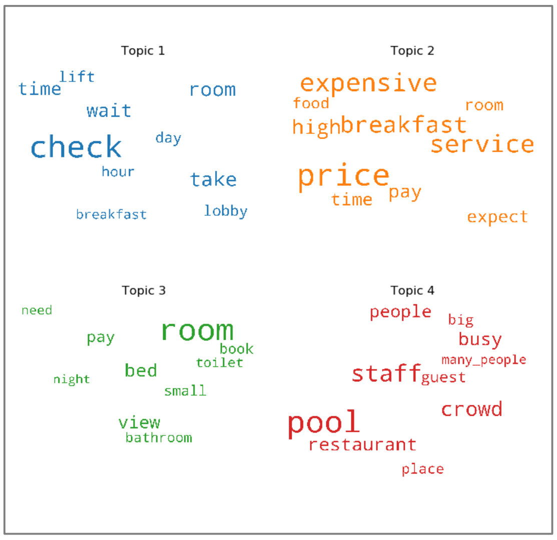
Bigram 4-topic model



Appendix B

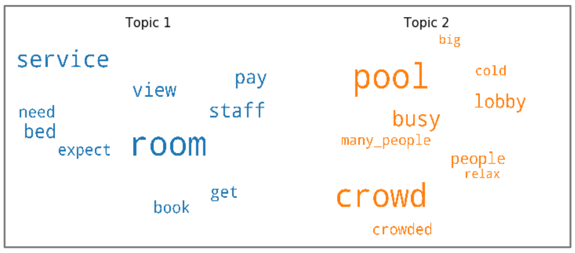
Topic segregation and top 10 keywords per topic for baseline LDA model

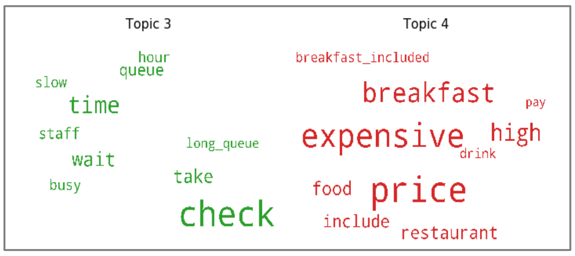




Appendix C

Top 10 keywords per topic for NMF model (pyLDAvis plot is only applicable to LDA models)





Appendix D

Best Model	Best Parameters	Facility	Security	Pricing	Location	F&B	House-keeping	Front Office	others
Light GBM	min_child_samples	14	14		16	17	17		
	lambda_l1	0.15	0.15		20	10	10		
Random Forest	n_estimators			300				100	100