**Problem Definition**

**Introduction**

Data is the voice of the customer; Data science is the interpretation of that voice. Business organizations are increasingly turning to analysing data for insights to optimize their operations and unlock greater opportunities to serve their customer better. Brands that dominate today’s business landscape are data-driven companies that effectively utilize valuable insights from their data to guide their decision making. The team decided to focus on multinational company to delivers actionable analytics solution to resolve complex business problems.

**Company Chosen**

The multinational company chosen for this project will be Airbnb. Airbnb is the world leader in accommodations of the “sharing economy” that allows client to find places to stay directly from individuals in thousands of cities around the world.

**Why we choose AirBnb**

In 2018, statistics showed that Airbnb had 150 million users. Data on Airbnb’s website state that about 2 million people stay in an Airbnb every night. With 20TB of data created daily and 1.4 petabytes of archived data, data has become the lifeblood of AirBnb. Data science technology is important and will be a key differentiator for rapid growth of AirBnb. The team aim to develop strategic and creative business solution using the data generated.

The main market that the team will be focusing on will be Singapore market. Airbnb has a long and complex relationship in Singapore. Despite being labelled as an illegal service by the government, Airbnb still have a strong presence in the Singapore market. Tourism being a major industry and contributor to the Singapore economy, Airbnb is able to grow rapidly in Singapore and generate a high comprehensive data within Asia.

**Defined Problems**

**Sentiment Analysis using Natural Language Processing**

Sentiment Analysis is a powerful marketing tool that enables product managers to understand customer emotions in their marketing campaigns. It is important factors when it comes to product and brand recognition. At Airbnb, guest are likely to leave better reviews even if the experience was only just satisfactory due to real-life interaction with the host. These reviews falsely portray a positive image for the host and guest and ratings are always exaggerated. To interpret the true feelings of user, Natural language processing technology will be used to analysis the reviews from the guest through sentiment analysis.

**Predictive Modelling**

Predictive modelling technique can help AirBnb to analyse how various markets will be perform so that resources can be prioritized. Using predictive modelling, Airbnb can create market specific forecast with multiple variables.

**Data Preparation & Cleansing**

The team decided to leverage on data scraping platform to collect the relevant Airbnb Data. Insider Airbnb is a mission driven activist platform that sourced Airbnb data from publicly available information from the Airbnb site. Although the data has been minimally cleansed and processed by the activist platform, the chance of missing value, odd symbol and characters, sparse columns are high.

**For this project, the focus is on the 6 datasets and 1 geojson file.**

|  |  |
| --- | --- |
| **File Name** | **Description** |
| Listings.csv | Detailed listing data for Singapore |
| Calendar.csv | Detailed Calendar data for Singapore |
| Reviews.csv | Detailed Review data for Listing in Singapore |
| Listings.csv | Summary information and metrics for listing in Singapore |
| Reviews.csv | Summary review data and listing ID |
| Neighbourhoods.csv | Neighbourhood list for geo filter. |
| Neighbourhood.geojson | GeoJSON file of neighbourhoods of the city. |

**Sentiment Analysis using Natural Language Processing**

For this component, the required CSV will be Reviews.csv that contains the detailed review data for listing in Singapore. There are 43239 rows and 6 columns.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| listing\_id | Integer | Airbnb’s unique identifiers for each Airbnb apartment |
| id | integer | Unique identifier for the comment |
| date | datetime | Datetime that comment is being written |
| reviewer\_id | integer | Unique identifier for the reviewer |
| reviewer\_name | Text | Name of the reviewer |
| comments | Text | Detailed comment of the listing |

**NLP**

**Data Preparation**

**(refer to Git Hub Customer Review System 1.final data cleansing for review)**

When building a NLP model for sentiment analysis, semantic classification of comments is very important to train and test data. Due to time constraint and huge amount of data, manual labelling of comments is very difficult. The team decided to use SentimentintensityAnalzyer to generate the requires labels that is done manually. Our team choose SentimentintensityAnalzyer as study shows that VADER performs as good as individual human raters at matching ground truth 96%. The role of SentimentintensityAnalzyer is to simulate manual labelling for the dataset only. Different model will be built for this sentiment analysis. The compound index is used to determine the comment into different tiers. 1 to -1 (positive to negative) 7 tiers were created using the compound index. Neutral tier is remove for this project as we are not predicting neutral comment.

|  |  |
| --- | --- |
| Tier | Index |
| #-0.66 to -1 | VN (very negative) |
| #-0.33 to -0.66 | QN (quite negative) |
| #0.00 to -0.33 | N (negative) |
| #0.00 to 0.33 | P (positive) |
| #0.33 to 0.66 | QP (quite positive) |
| #0.66 to 1 | VP (very positive) |
| 0.00 | NE (neutral) (removed) |

**Cleansing**

Several problems were identified when exploring the csv dataset.

1. **Unstructured Data collected due to data scrapping (HTML Tags, Unicode)**

The data was extracted via web scrapping. Web scrapping is an automated method used to extract large amount of data from websites. The data scraping is unstructured. It is necessary to convert the unstructured data into structured form. HTML tags and Unicode must be cleansed before processing to build model for natural language processing.  
Regex (Regular expressions) are useful for defining filters. Regular expression contains a series of character that defines a pattern of text to be matched.

**##function to remove html syntax using functions**

reviewData['comments']=reviewData['comments'].str.replace(r'<[^<>]\*>', '', regex=True)

**##function to remove unicode**

reviewData['comments']=reviewData['comments'].str.replace('[^\x00-\x7F]','',regex=True)

|  |
| --- |
| Positives are  <br/>1. Good value for money stay location for a family of 4  <br/>2. Safe place and decent location |

|  |
| --- |
| æˆ¿é—´å’Œæè¿°çš„ä¸€æ ·çš„ï¼Œç¦»æœºåœºå¾ˆè¿‘ï¼Œå9è·¯ä¸¤ç«™å°±åˆ°åœ°é“ï¼Œè¿˜æ»¡æ–¹ä¾¿çš„ã€‚è¿™ä¸ªæˆ¿é—´ä¸Ž3å·æ˜¯å…±ç”¨å«ç”Ÿé—´çš„ï¼Œæ¸…æ´åº¦ä¸é”™ï¼Œæ—©é¤æ˜¯é¸¡è›‹ç‚’ç±³ç²‰ï¼Œç¨ç®€å•äº†äº›ã€‚ |

**Figure 1 shows the sample comment extracted from the csv**

1. **Removal of Stop-word and punctuation**

Stop words are available in abundance in any human language. By removing these words, low-level information will be removed from comments to give more focus to the important information. Removal of stop words also reduces the dataset size and reduce training time due to the fewer number of tokens involved in the training. Like stop-word, punctuation in comments add up noise that brings ambiguity while training the model.

1. **Remove word and digits containing digits**

It is common that words and digits combine are written in text which will create problem for machines to understand. Words like 24hr and 71bus are difficult to process removing them will reduce the dataset size and reducing training time.

1. **Tokenization**

Tokenization breaks the raw text into words called token. These token helps in understanding the context and developing the model for the NLP. The tokenization helps in interpreting the meaning of text by analysing the sequence of words. Tokenization column is created

1. **Stemming**

Stemming is the process of reducing word to its word stem that affixes to suffixes and prefixes or to the roots of word hence aiding in the pre-processing of text for text normalization. For example, Send, sent and sending. All three words are different tenses of the same root word send. Stem will simplify them into one root word stem.

1. **Lemmatizing**

Lemmatizing is used to reduce words to a normalized form. In lemmatization, the transformation uses dictionary to map different variants of a word back to its root format. Non-trivial inflections such as “is”,”was” are converted to the root “be”.

1. **Translating different Language (German, French etc) into English**

Some guests left their comments in different languages such as German and French. A function langdetect was written to identify comments that is not written in English. After identifying the rows, google deep translator is used to translate those comments into English.

**After Cleansing:**

**We will have 37352 rows and columns highlighted in red is the important column for this project.**

**Review CSV**

|  |  |
| --- | --- |
| **Listing\_id** | Listing ID of the apartment |
| **id** | Id of the comment |
| **Date** | Date of comment |
| **reviewer\_id** | Reviewer ID |
| **reviewer\_name** | Reviewer name |
| **comments** | Cleansed comment without HTML tag, word and digits containing digits |
| **comments\_nostop** | Cleansed comment without HTML tag, word and digits containing digits and without stop word |
| **comments\_tokenized** | Tokenized word |
| **stemmed** | Stemmed word |
| **lemmatized** | Lemmatized word |
| **compound** | Compound index of the rating |
| **Neg** | Negative ranking from the Vader |
| **Neu** | Neutral ranking from the Vader |
| **Pos** | Positive ranking from the Vader |
| **bin** | Temporary bin assign to the different category |
| **Bin\_word** | Bin words of the six different tiers (VP,QN,N,P,QP,VP) |

**Sampling Method**

**(refer to Git Hub Customer Review System 2.Data Preparation,3.undersample(Eda),3.oversample eda)**

**Total row 37352 Rows**

|  |  |
| --- | --- |
| **Tiers** | **Percentage** |
| **VN** | **0.58%** |
| **QN** | **0.70%** |
| **N** | **0.92%** |
| **P** | **2.77%** |
| **QP** | **20.56%** |
| **VP** | **74.45%** |

The table above show the above show the composition of each different tier. Interesting observation discovered was the ratio for each tier is not equal. This could be due to the credibility of the reviews. Customers might not made honest due to real-life interaction with the host thus the ratings are always exaggerated. Another possible reason is companies planting positive reviews of their own product and sully competitors with negative reviews. Thus, there is a need to resample the data. Two methods were used to resample the data.

**Random-Oversampling:** Randomly duplicate rows in the minority classes (825 rows initially) (VP, QN, N) since the positive classes (P, QP, VP) is the majority classes (36527 rows). Minority classes (VP, QN, N) are randomly duplicate to ensure equal sample size as the positive class. Total will be 73054 rows.

**Random under-sampling:** Randomly delete rows in the majority class to ensure same ratio as the minority classes. 219 VP will be selected, 262 QP will be selected, 344 P will be selected to ensure same ratio as N, QN, VN.

**Exploratory Data Analysis on AirBNB using Python (Sentiment Analysis using Natural Language Processing)**

In this phase, we can reveal hidden patterns in the data and generate insights from it. Different techniques are used to understand the text data before building the model. We will explore both random-oversampling and random under-sampling during this stage of EDA.

**Analysing text statistics**

Text statistics visualization are simple but very insightful techniques. Word frequency analysis, sentence length analysis. Those help explore the fundamental characteristics of the text data. We will be using mostly histogram and bar chart.

**Oversampling key-findings:**

**For both (positive tiers and negative tiers)**

Average length is 197.46 char.

Unique word count 1073 word

Important for padding and configuration of the model.

**Positive tiers (P, QP,VP)**

Top 20 word used

Table

Description automatically generated

NGRAM Exploration

Ngrams are simply contiguous sequences of n words. Looking at most frequent n-grams can give a better understanding of the context in which the word was used.

2gram

Chart, bar chart

Description automatically generated

3gram

Chart, funnel chart

Description automatically generated

4gram

Chart, bar chart

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

**Negative tiers (N, QN, VN)**

Table

Description automatically generated

2gram

Chart, funnel chart

Description automatically generated

3gram

Chart, funnel chart

Description automatically generated

4gram

Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated

**Word-cloud**

Word-cloud is a great way to represent text data. The size and colour of each word that appears in the word-cloud indicates its frequency or importance

Positive tiers

A picture containing text, newspaper, screenshot

Description automatically generated

Negative tiers

Text

Description automatically generated

THE word good appear in negative tiers for the following reason. Stop word is already removed. Not Good. Not is removed

**under sampling key-finding**

Average length is 172.24 char.

Unique word counts 140 words

Important for padding and configuration of the model.

**Model-building (LSTM)**

**(refer to Git Hub Customer Review System 3.1 oversampling model and 3.1undersamping model)**

Prior to building the NLP, tokenizing and stemming was done. Vectorizing and padding each comment is also done to ensure each comment have equal length. To create and optimize the model, the team decided to tune using batch size and epochs. The team also decided to evaluate this model using several measurements such as accuracy, precision recall and F1 score. The model that the team use is LSTM (long-short term memory. This model is very helpful in NLP. its use appropriate layers of embedding and encoding. It will also be able to find out the actual meaning in input string and will give the most accurate output class. We also involved validation of dataset in this model.

**The team decided build both under sampling model and oversampling for NLP but decided to only focus on oversampling model because under sampling model only have 80 unique word which is not sufficient and accurate to build NLP model.**

**Result**

Model 1 (vocab\_size = 800, embedding\_size = 32, epochs=25,batch\_size = 64)

Accuracy: 0.9212

Precision: 0.9309

Recall: 0.9158

F1 Score: 0.9233

Model 2 (vocab\_size = 800, embedding\_size = 32, epochs=50,batch\_size = 64)

Accuracy: 0.9390

Precision: 0.9438

Recall: 0.9342

F1 Score: 0.9390

Model 3 (vocab\_size = 800, embedding\_size = 32, epochs=75,batch\_size = 64)

Accuracy: 0.9395

Precision: 0.9434

Recall: 0.9371

F1 Score: 0.9403

Model 4 (vocab\_size = 800, embedding\_size = 32, epochs=75,batch\_size = 32)

Accuracy: 0.9407

Precision: 0.9445

Recall: 0.9368

F1 Score: 0.9406

Model 5 (vocab\_size = 800, embedding\_size = 32, epochs=50,batch\_size = 32)

Accuracy: 0.9389

Precision: 0.9428

Recall: 0.9359

F1 Score: 0.9394

Model 6 (vocab\_size = 800, embedding\_size = 32, epochs=25,batch\_size = 32)

Accuracy: 0.9320

Precision: 0.9382

Recall: 0.9279

F1 Score: 0.9330

The team chose model 3 instead of model 4 due to its overall performance. Although Model 4 has better performance in terms of accuracy, precision and f1 score, the improvement isn’t significant. Model 3 uses less training time compared to Model 4 due to its batch size.