CMPT 732: Big Data Lab 1 (Fall 2018) – Project Report

Date: 7th December 2018

The Battle for Bookings:
A Comparative Assessment of Hotels and Airbnb's on a Global Scale

Team CodeNinja: Anurag Bejju – abejju@sfu.ca, Andrew Wesson-awesson@sfu.ca

1. Problem Statement:

Based on a recent study, Travel and tourism is one of the world's fastest-growing sectors, with bookings hitting close to $1.6 trillion in 2017[1]. Each year, the global traveler pool is flooded with millions of new consumers from both emerging and developed markets, many with rising disposable incomes and a newfound ability to experience the world. With most travelers trying to get the best bargains on basics such as accommodations and transportation, we wanted to analyze the cost of accommodations around the globe. Since there are many options available, tourists either opt for hotel rooms or turn to Airbnb: an online community marketplace to plan their stay. As part of our Big Data project, we decided to compare the costs of hotel and Airbnb rates in 63 cities around the world by leveling the location and ratings of the measuring parameters to see how the deals stack up.

2. Methodology:

Our project's Big Data Pipeline consists of 5 stages *[Appendix 1]*. Each stage has been briefly explained in the below mentioned sections.

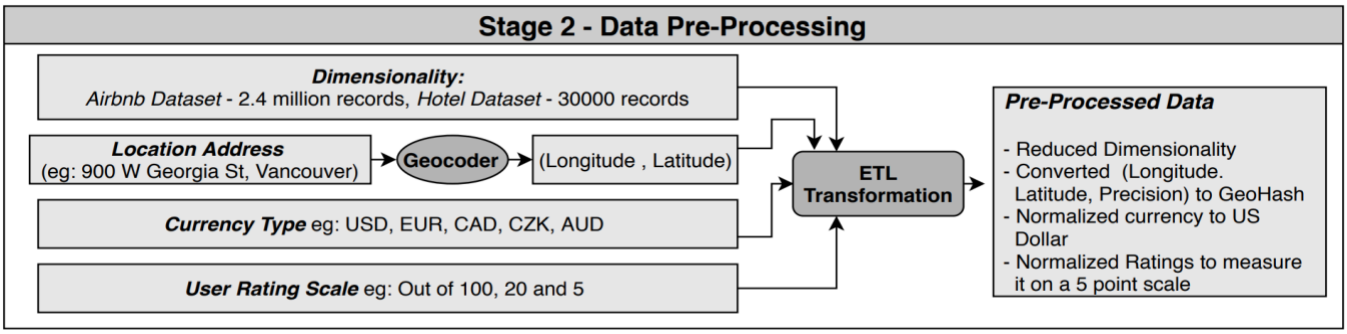
2.1. Data Collection

Airbnb Data: InsideAirbnb[2] has made Airbnb data accessible by using publicly available information from the Airbnb site and provides a snapshot of searchable listings with various parameters like dates, prices of future available dates, reviews and locations with listing metadata. In order to improve scalability, reduce contention, and optimize performance, this data was partitioned based on the city.
*(Data Size – 63 Cities * 27000 listings each (approx.) * 100 parameters => 1.75 million * 100 records)*

Hotels.com Data: This data was compiled using web scraping feature provided by import.io. We have extracted publicly available data about the hotel name, street, locality, guest reviews, rating and room pricing for all the above 63 cities. We have latter partitioned this dataset based on the city.

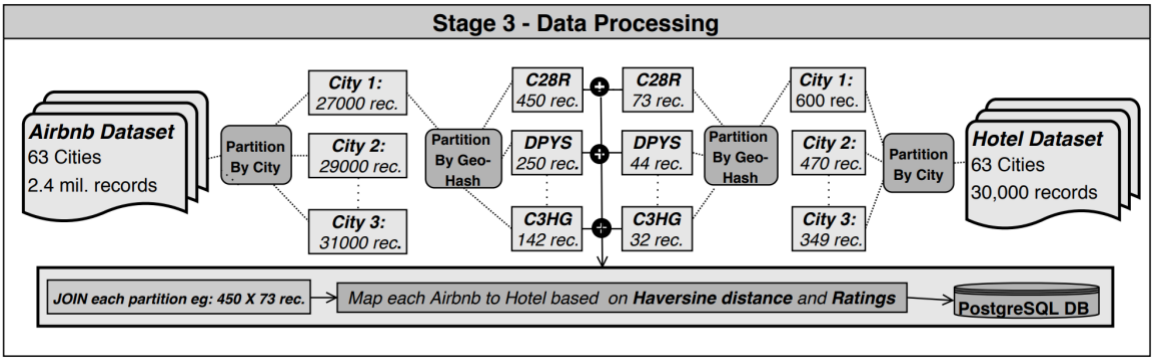
2.2. Data – Preprocessing

Real-world data at its earliest stages can often be very un-structured and dirty in form. In order to make our data more concise and be able to perform a more accurate analysis, this particular stage was one of the most crucial, significant and complex parts of our project. Since the data collected was from completely two different sources, it brought in significant challenges with it. Here are some the problems we have tackled in our project.



Data – Preprocessing Tasks
Varying dimensionality of our dataset
The hotel dataset had a different parameter dimensionality for each city we have scrapped. Since each one follows different location addressing system, they had a combination of different parameters like postal codes, zip codes, provinces, localities, counties, etc. We have fixed it by writing an ETL script that maps values to any one of these parameters and then transforming it into a single unified address. We have also reduced dimensionality by removing columns that went out of the scope of this project.
Converting location address to geo-coordinates using GeoPy library
As our analysis uses geohashing and haversine formula to find hotels and Airbnb's that are in close proximity, we had to geocode our addresses for hotel locations. This was achieved by using a python geocoder that inputs the location address and provides it's GPS coordinates. We had to process it individually by the city as the location address can be same in different cities around the globe.
Converting GPS coordinates to Geo-Hash Values
As explained in the latter part of this report, to perform joins more efficiently, we have used geo-hashing techniques to convert GPS coordinates to their Geo-Hash Values. This inputs longitude, latitude and precision and generates a hash value having similar prefixes if they are nearby to each other.
Normalizing Hotel and Airbnb prices to a common currency (i.e US Dollars)
Since the hotel dataset spanned over multiple cities in different countries, the prices scrapped were in various currencies. We had to convert it to US Dollars in order to perform comparisons more accurately.
Normalizing Ratings to a 5 point scale system
The Airbnb Ratings were rated on a scale of 0-100 and the hotel ratings were a combination of hotel.com ratings (0-5) and trip advisor rating (0-10). All the ratings were normalized on a 5 point scale and the hotel ratings were generated considering both the values.

2.3. Processing Dataset

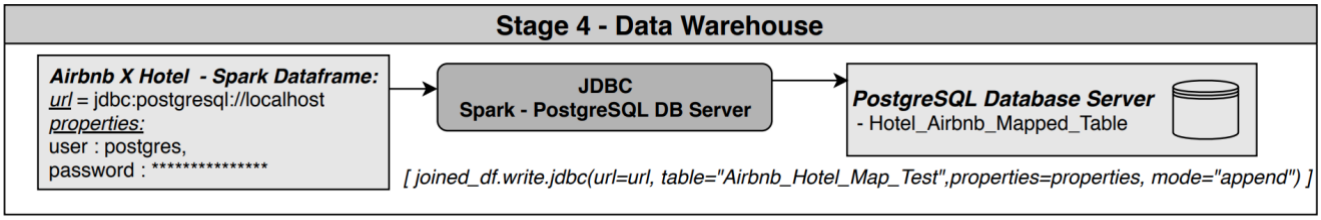


In order to facilitate the matching of Airbnbs to nearby and similarly-rated hotels, we perform an ETL step on each dataset to load them into HDFS in Parquet format. The Airbnb dataset was partitioned by the city, with each city having about approx. 26,000 records. The data was loaded using pandas and then converted to a spark dataframe with correct schema and then write it to HDFS in Parquet format. During this ETL step, the two datasets were co-partitioned so that the join happens for relevant portions of each dataset efficiently, especially if the scale of the hotel dataset was increased. To increase the number of partitions while keeping locations that were close to each other within the same partition as much as possible, we used Geohashing [3] technique to convert latitude-longitude coordinate pairs into strings which group together nearby points, with variable precision [4]

We used a precision of four for this project, which seemed to reduce the partition size substantially but not excessively. Then the two datasets are joined based on the city and geohash values and then each Airbnb-hotel pair is computed based on the similarity in location and review scores. We compute the distance using the Haversine formula, which estimates the distance between two points on a sphere [5]. Later for each Airbnb, we filter out all hotels that are within 50 km distance and have similarity in review scores by measuring a similarity score out of 10. Finally, this score helps us map the hotel that resulted in the highest score for each Airbnb.

2.4. Data Warehouse

Once we finished processing Airbnb and Hotel data, we decided to store it in a PostgreSQL database. We chose this as it is one of the best-performing relational databases which has been a solid choice for most business applications where data mining or reporting is significant. Companies like Amazon which use Redshift as their data warehousing platform, use a modified version of PostgreSQL in it. We have also chosen it as it is very compatible with Tableau.

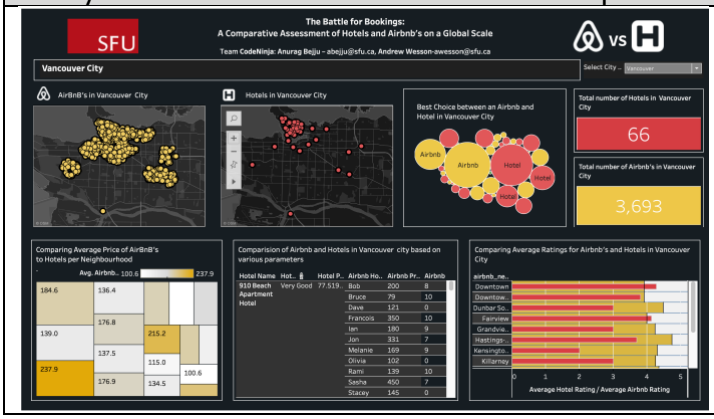
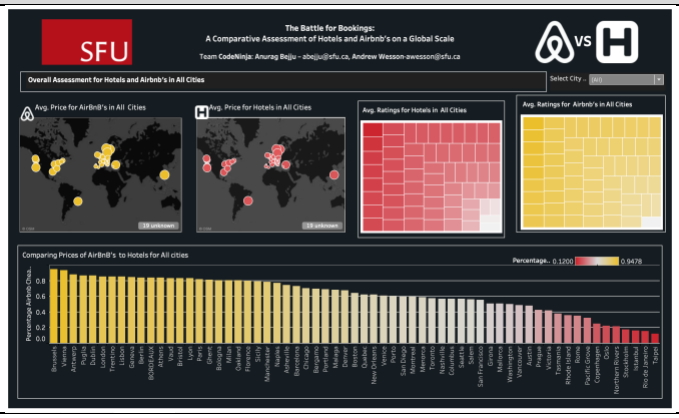


Therefore, we transferred data present in a spark dataframe to a PostgreSQL database, by setting up a JDBC Driver that connects spark to locally hosted PostgreSQL database server.

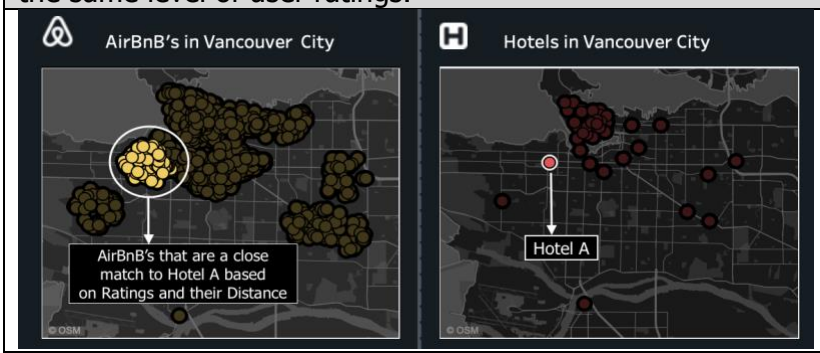
2.5. Visualization

As human brain processes information presented in form of charts or graphs better over data compressed in a spreadsheet, we tried to make our visualization as intuitive as possible. Since our goal was to compare prices for Airbnb's and Hotel's which were in close proximity and had the same level of user ratings on a global scale, we made sure our dashboard provided a clean and concise analysis for the said problem statement. Tableau was used to make these dashboards.

The Battle for Bookings Dashboards

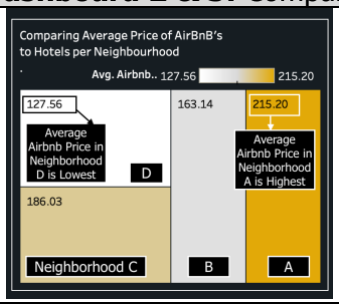
City-Wise Assessment of Hotel and Airbnb prices	Overall Assessment of Hotel and Airbnb Prices
	

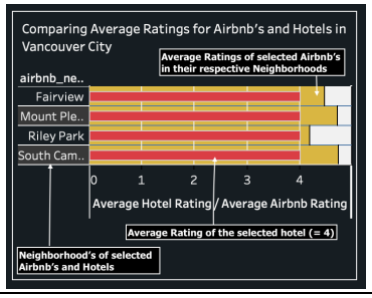
Dashboard 1: Visualizing the mapping between a Hotel and Airbnb's that are in close proximity and have the same level of user ratings.



In this dashboard, *Map 1* shows all the Airbnb's and *Map 2* shows Hotels in the chosen city. When you select a hotel on Map 2, it shows all the Airbnb's that are in close proximity to each other and have the same level of user ratings.

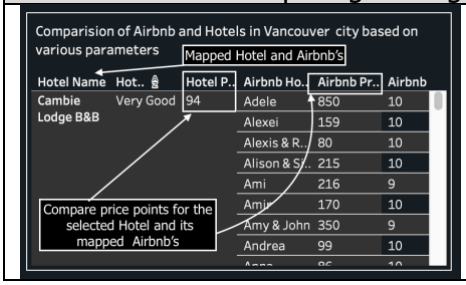
Dashboard 2 & 3: Comparing Average Ratings & Price of Airbnb's and Hotels per Neighborhood





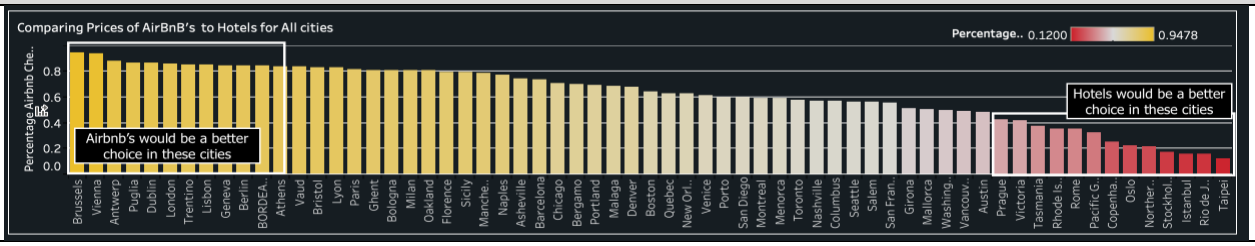
Dashboard 2 and *3* provide the Average Ratings and Prices for the Neighborhoods where the selected Airbnb's and Hotels are located. This lets you compare the price and ratings of the neighborhood you are interested to live more conveniently.

Dashboard 4: Comparing prices for the current selection Airbnb's and Hotels per Neighborhood
Dashboard 5: Comparing Average Price of Airbnb's and Hotels per Neighborhood



Dashboard 4 lets you compare prices for the current selection of Airbnb's and Hotels and *Dashboard 5* provides the best choice you can make between them. The biggest circle represent that the best choice for the selected Airbnb's and Hotels.

Dashboard 6: Comparing prices for Airbnb's and Hotels for all 63 Cities



This dashboard gives an overall price assessment of all 63 cities. We found that Airbnb's in most European cities and hotels in Asian and Scandinavian countries would be an optimum choice for travelers who would like to pay less for the same location & quality of a particular accommodation type.

Dashboard 7: Comparing Average Ratings and Hotels for all 63 cities

This dashboard compare's rating for hotels and Airbnb's across 63 cities. Asheville had the highest average hotel rating of *4.598* and Trentino had the lowest rating with *2.518* score. Whereas Mallorca had the best average Airbnb rating of *4.347* and Oakland had the lowest rating with *3.283* score.

3. Scalability, Limitations and Future Work

We have devised our big data pipeline keeping scalability in mind. By partitioning data based on city and geohash values and using tableau for visualization, we could effectively scale it up to handle a much larger dataset. Also, the computation could be vectorized[6] in order to improve overall performance If *Apache Arrow* is installed on the cluster. One of our main limitations was the lack of a large, robust, and detailed hotel dataset which we could have devised a more complex metric of similarity, taking into account things such as amenities, number of allowed guests, and maximum stay length. We also could make the price comparison more accurate by taking into account the cleaning fee and allowing the user to enter in the desired stay length. This would account for cases where an Airbnb is a more expensive option for shorter stays due to one-time fee, but is cheaper for stays of a certain length.

4. Project Summary

S.No	Category	Points
1	*ETL*: Extract-Transform-Load work and cleaning the data set.	5
2	*Visualization*: Visualization of analysis results.	4
3	*Bigness/parallelization*	3.5
4	*Getting the data: Acquiring/gathering/downloading.*	2.5
5	*UI:* User interface to the results, possibly including web or data exploration frontends.	2
6	*Problem:* Work on defining problem itself and motivation for the analysis.	2
7	*Technologies: New technologies learned as part of doing the project.*	1
	Total:	**20**

5. References:

[1] Douglas Quinby, Phocuswright Conference, Florida, November 9, 2017.
[2] http://insideairbnb.com/get-the-data.html
[3] https://github.com/wdm0006/pygeohash
[4] http://www.bigfastblog.com/geohash-intro
[5] https://www.movable-type.co.uk/scripts/gis-faq-5.1.html
[6] https://databricks.com/blog/2017/10/30/introducing-vectorized-udfs-for-pyspark.html

[Note 1: The dashboards can be accessed on https://www.anuragbejju.com/big_data_project/]

[Note 2: The code base can be found on https://csil-git1.cs.surrey.sfu.ca/CodeNinja-big-data-1-project/BIG_Data_1_Project]

Appendix 1 :The Battle for Bookings: A Comparative Assessment of Hotels and Airbnb's on a Global Scale

Big Data Processing Pipeline

Data Collection

Airbnb Dataset
63 Cities - **27000** listings/City
Over 100 parameters
Total: **2.4 million* 100 records**
Partition based on City

import io

Hotels Dataset
Web Scrapped **30000** Hotels
spanning over **63** Cities
Over **30** parameter per city
Partition based on City

Data Pre - Processing

Pre - Processing Datasets

- Reduced Dimensionality by removing unnecessary columns.
- Extracted (longitude, latitude) information by geocoding address information
- Converting (longitude, latitude, precision) to a geoHash value
- Normalizing price values to US dollar
- Normalizing Ratings to measure it on 5 point scale
- Restructuring Data to fit desired schema

Spark

Data Processing

Processing Datasets

- Loaded datasets into HDFS, partitioned by city and geohash
- Joined Airbnb and hotel datasets on city and geohash
- Scored Airbnb-hotel pairs based on location and review score similarity
- Selected the highest scoring pair for each distinct airbnb

Spark

Data Warehouse

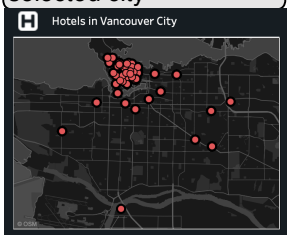
Hotel_Airbnb_Mapped_Data

PostgreSQL

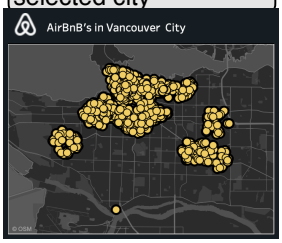
Visualization

Battle of the Bookings Dashboard - City

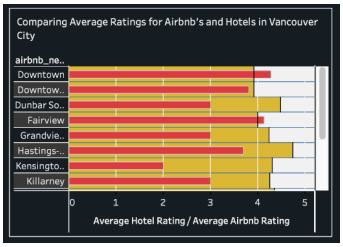
Map of Hotel's in the selected city



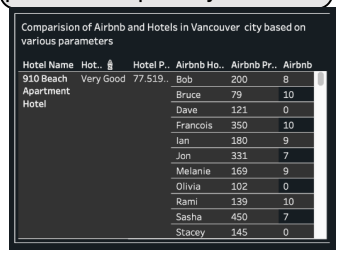
Map of Airbnb's in the selected city



Average Ratings for Airbnb's and Hotels in a City

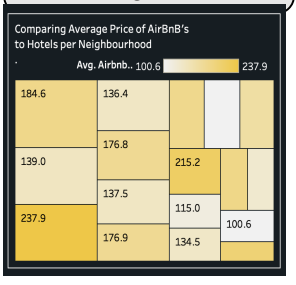


Comparison of Airbnb and Hotels based on various parameters per city

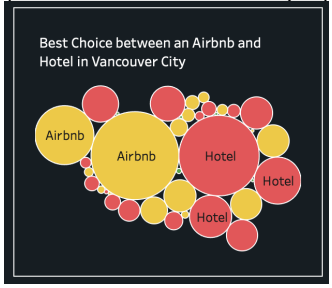


Hotel Name	Hot..	Hotel P..	Airbnb Ho..	Airbnb Pr..	Airbnb
910 Beach Apartment Hotel	Very Good	77.519..	Bob	200	8
			Bruce	79	10
			Dave	121	0
			Francois	350	10
			Ian	180	9
			Jon	331	7
			Melanie	169	9
			Olivia	102	0
			Rami	139	10
			Sasha	450	7
			Stacey	145	0

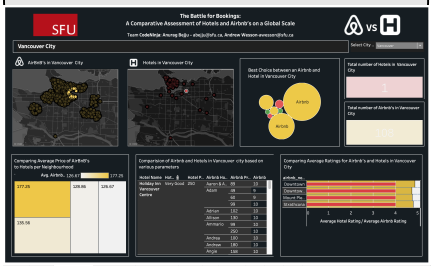
Comparing Average Price of AirBnB's to Hotels/ Neighborhood.



Best Choice between an Airbnb and Hotel in a City



On Select - Dashboard Changes



tableau