Agenda :

1) **Select a topic**
2) **Deliverable per Person**
3) **Any concerns**

1. SELECT FINAL TOPIC

Dataset (Incl. Link)	Problem/Opportunity?	Notes
Retail Data	Predict sales, Model effects of markdowns, recommend actions based on highest impact	2010-2012, multiple files (merge)
Rent Index	Predict rent by area (state/city), show history and future pricing	Zillow, can try to find a different set
http://archive.ics.uci.edu/ml/datasets/Online+Retail	Retail Walmart Sales data	This is another set of Retail Data
Airbnb (Toronto)	Predicting AirBnb Value https://www.kaggle.com/kerneler/starter-toronto-airbnb-dataset-0e18d1bb-8 We can edit the code below by forking into a Group Kaggle account, I imported starter code from Kaggle into my account: **https://www.kaggle.com/fahadahmad100/capstone-airbnb/edit**	Related Lits : https://airbnbdataanalysis.wordpress.com) https://siddharthsabat.medium.com/a-data-analysis-of-airbnb-boston-listings-324de993ca9e#:~:text=It%20has%203585%20OBSERVATIONS%20%28property%20listings%29%20and%2095,The%20data%20sets%20are%20obtained%20from%20Inside%20Airbnb. https://medium.com/@fuweijia0528/analysis-of-seattle-airbnb-data-34ec1ceed5f7
COVID world vaccination rates Country regional world GDP COVID cases deaths recoveries	Covid vaccination rates and impacts on global economy	May need to pull a secondary dataset to join showing the GDP or inflation rates globally over time?

Fahad said: I like the datasets. I was wondering if we could implement clustering (a form of customer segmentation) followed by regression (i.e., linear or logistic) model on the retail data?

Not sure what the Business problem could be for retail Walmart Sales data. Perhaps, the company is trying to increase sales which in turn increase revenue and in order to achieve this goal, the company must attract new customers and reduce customer churn by identifying the main characteristics of customers such as type of transactions of the prime buyers of the company's products. I guess it relates to customer segmentation.

Fahad said: I think the Airbnb dataset provided by Roma might be a good option. Has a lot of features and could apply clustering and regression.

Is there a target variable for airbnb data? Looks like it's reviews_scores_rating.

Perhaps Business objective is looking at ways to improve reviews of airbnb to have customers rent. Not sure how airbnb works. Factors or features that drive airbnb value in terms score on review, could be neighborhood, room type, number of nights available. There seem to be many features. I am not sure about the business objective, but I think with the machine learning code to run, it would be clustering which do a segmentation resulting in clusters (i.e., 1 cluster where the room type is large and so forth, another cluster where the room type is small and so forth). So each cluster would contain quantity of the features of dataset which would distinguish it from features of other data set. Then do a regression on each cluster, with target variable y being review score. Haven't found any tutorial that showed clustering followed by multiple or logistic regression. Perhaps, we could reach out to Sasha as she mentioned previous group she worked with also did cluster analysis followed with some type of regression.

Jeni's thoughts:

Instead of picking random datasets and trying to decide what to do with them, what would people think about aligning on a problem of interest and then sourcing the dataset? Some things that interest me:

1. People analytics (from an organizational perspective - we could hone in on a topic like gender parity in the workplace, gender paygap or even the workstyle/benefits an organization offers in comparison to annual revenue?)
2. Covid Vaccination rates and impacts on the economy This could be really neat since it's a global issues, we could do some cool visuals with tableau using maps, and there is datasets related to world vaccine progress: https://www.kaggle.com/fedesoriano/coronavirus-covid19-vaccinations-data

What else is everyone interested in? Perhaps once we can align there we can work together to find a dataset to study that problem?

Mariana: leaning towards airbnb (because of the data) or covid vaccines (because it is a global issue)

2. DELIVERABLES

At the onset of the project, what would the group think about gaining consensus on our plan for dividing work? We could start by entering our names below and then information about our background and strengths so we can utilize all of the skills on our team in the best way?

Idea for dividing up the following deliverables:

Capstone Deliverable 1: Capstone Proposal 15% - September 30 - Dongy, Roma, Fahad
- One person?

Capstone Deliverable 2: Cleaning and Visualization Report 20% - October 14 - Fahad, Roma, Mariana
- Two people? One does cleaning, the other visualization report

Capstone Deliverable 3: Predictive Modeling 15% - November 11 - Jordan, Fahad, Ryan
- One person?

Capstone Deliverable 4: Visualization with Dashboard 15% - November 25 - Mariana, Huy, Jordan
- Two people? One works on one sheet, the other works on another sheet and both go into one dashboard

For all deliverables - I believe the one or two people should have it done a couple days before the due date so that the rest of the team can review and add to it.

Capstone Final Deliverable 5: Presentation 15% - December 2 - Jennifer, Hugh, Fahad
- One or two people? So that each can divide what to say
    - The others will be available to answer the questions at the end of the presentation

Capstone Final Deliverable 6: Report 20% - December 16 - Jennifer, Roma, Ryan
- All - divide this up equally or all add to the report
    - One person in charge of cleaning it up / submitting

Name	Background / Expertise	Project Responsibilities
Jeni Miller-O'Connor	Analysis Presentations Project Management Research ROI / Financial Analysis / Accounting Tool preference: Jupyter Notebook, Tableau	**Deliverable 5** **Deliverable 6**
Roma Casubha	Financial Analysis Tool Preference : Python or R (coursework) Tableau : Used a bit at work	**Deliverable 1** **Deliverable 2** **Deliverable 6**
Huy Huynh	Business Analyst: Experienced in Marketing Analytics Relevant Tools: Python Tableau/Power BI (novice level) Relevant Skills: Data Prep/EDA Predictive Modelling (from course material) Data Visualization/Presentation	**Deliverable 4** **Deliverable 5**
Mariana Mariles	Sports Analyst	**Deliverable 2**

	Background in Mathematics and CS Python, R, SQL Tableau (Use at work) Web Scraping Cloud Computing (Azure)	**Deliverable 4**
Fahad Ahmad	Analysis Presentations Research Tool preference:Jupyter Notebook Python, R Tableau, PowerBI (course work) Data visualization (course work) Data cleaning (course work) Machine Learning (course work)	**Deliverable 1** **Deliverable 2** **Deliverable 3** **Deliverable 4-Assist** **Deliverable 5-Assist**
Dongyan Hong	Accounting professional Financial Analysis and Research Tool Preference : Python or R; data visualization - Tableau	**Deliverable 1**
Jordan	Researcher at a Labour Union Data cleaning - Python, R, Power Query (Decent in each) Data Analysis - Python, R Data Viz - Power BI (advanced), Tableau (intermediate) Econometrics Not great at writing reports	I would love to do the dashboard for Assignment 4 **Deliverable 3** **Deliverable 4**
Ryan	Python, R, JS, C# Problem solving Organization/Proofing	**Deliverable 3** **Deliverable 6**

TEAM 4 DELIVERABLE 1 : PROJECT PROPOSAL (Deadline : September 30)

"Success in Hosting"

I. Problem Statement & Stakeholders

AirBnb, established in 2008, is one of the most popular online housing platforms offering cheaper vacation rental alternatives for travellers compared to the more expensive hotel accommodations. Despite being hardly hit because of the pandemic travel restrictions, AirBnb continues to be a valued company, and even went public last December 2020. As businesses reopen and economies in various sectors such as hospitality and tourism regain foothold due to post-recovery from COVID-19, we believe that the demand for travel will flourish and will serve as a great opportunity to generate revenue in rental hosting. To maximize potential revenue, AirBnB hosts can make use of attributes that drive booking price and improve these features for the benefit of travellers.

Hosts pay commissions to Airbnb when customers book accommodations via Airbnb's online platform. Customers rely mainly on AirBnb's past customer ratings. The ratings therefore reflect overall customer satisfaction. To leverage this, knowing rental properties features that significantly affect customers' satisfaction rating during their visits could help the host improve their service. Hosts will have the opportunity to explore potential areas for improvement, shape their reputation and increase satisfaction of the customers. This can therefore lead to high commissions for Airbnb with the goal of increasing revenue by 2%.

II. Data-set review and Methodology

The Airbnb Toronto data set for which a description of features is provided in Table 1 and provides AirBnb house/room ratings between August 2009 and September 2020 in the Greater Toronto Area. The Airbnb data is sourced from Kaggle[1]. As our business objective is to identify the major factors that affect customer rating that would influence revenue generated for Airbnb, we plan to implement multiple regression as our main machine learning algorithm. A clustering analysis will be implemented if the multiple regression provides results that require further explanation.

We will make use of 73 explanatory variables including but not limited to room type, number of accommodation and host response rate. However, as there are 73 explanatory variables, some of these variables may not be relevant for our predictions. We provide a data dictionary with a list of fields and descriptions in Table 1. For example, host_response_time refers to the time to respond to customer booking inquiry, and ranges from 1 hr to a few days; availability_30 refers to availability of property in 30 days; room_type refers to the specific type of room. The target

variables are price (i.e., price per stay per night) and review_scores_rating (i.e., customer-provided review score attributed to a listing based on overall experience and satisfaction).

In terms of data cleaning, we will first select the features out of the 73 features in the Airbnb dataset that are relevant for our machine learning algorithm represented in linear regression. After which the standard data exploration for selection of relevant features such as with use of visualization will be implemented. The following step would be the data cleaning process such as removal of NaN values, feature engineering and data normalization would be undertaken. Finally, we will implement two multiple regression models to identify major factors that predict price and predict customer satisfaction. One target variable being the price for one multiple regression model and the other target variable which is of more interest to us, being review ratings score.

The successful implementation of the two regression models will enable us to determine what features in the dataset are predictive of positive impact on revenue generation for Airbnb. We believe that features in the dataset that are predictive of price of stay per night and review ratings score will enable Airbnb company president and relevant stakeholders such as Airbnb Hosts to develop business strategies such as marketing strategies to emphasize these features when providing Airbnb service to existing and potential customers and as a result dramatically increase Airbnb revenue based on number of listings rented.

Appendix

Table 1. Fields or features in the AirBnb Data Set with description and type of feature stated

Reference

1 https://www.kaggle.com/robinkongninglo/toronto-airbnb-dataset

Number of Variables	Column Name	Description	Feature type(e.g., Numeric, String)
	id	Unique listing id (Primary Key)	Numeric
1	listing_url	Link to the rental property listing on Airbnb	String
2	scrape_id	Identifier for scraper	Numeric
3	last_scraped	Last date listing was scraped	Numeric

4	name	Title of Posting	String
5	description	Description of Posting	String
6	neighborhood_overview	Overview of neighborhood	String
7	picture_url	Link to the main vacation rental listing image on Airbnb	Image
8	host_id	Unique id for each host	Numeric
9	host_url	Link to the host profile on Airbnb	String
10	host_name	Host Name	String
11	host_since	Date an individual became a host	Numeric
12	host_location	Location of Listing	String
13	host_about	Description of host - relationship status, interests and hobbies	String
14	*host_response_time*	Time to respond to customer booking inquiry; ranges from 1hour to few days	Numeric
15	*host_response_rate*	Ranges from 0% to 100% for reply to booking inquiries	Numeric
16	*host_acceptance_rate*	Ranges from 0% to 100% response for acceptance of booking	Numeric
17	host_is_superhost	Is either t(true) or f(false)	Boolean
18	host_thumbnail_url	Host thumbnail URL	Image
19	host_picture_url	Host Picture URL	Image
20	host_neighbourhood	Description of neighborhood listing	String
21	host_listings_count	Current number of host listings	Numeric
22	host_total_listings_count	Total number of listings made by host	Numeric
23	host_verifications	Identifies how the host has completed the identity verification process	String
24	host_has_profile_pic	Host Profile Pic	Image
25	host_identity_verified	Identifies if the host has completed the verification process by indicating true or false	Boolean
26	neighbourhood	Specific location of Toronto area	String

27	neighbourhood_cleansed	Represents one of boroughs in Toronto in which a listing resides	String
28	neighbourhood_group_cleansed	No values (N/A)	N/A
29	latitude	The angular distance of a location or object north or south of the Earth's celestial equator	String
30	longitude	The angular distance of a location or object east or west of the meridian	String
31	property_type	Type of property (e.g., Entire house)	String
32	room_type	Specific type of room (e.g., Entire home/apt)	String
33	accommodates	How many people can stay (e.g., 6)	Numeric
34	bathrooms	No values, NA	N/A
35	bathrooms_text	Number of bathrooms in property	Numeric
36	bedrooms	Number of bedrooms in property	Numeric
37	beds	Number of beds in property	Numeric
38	amenities	List of amenities such as shampoo available in property	String
39	*price*	Price of stay per night	Numeric
40	minimum_nights	Minimum nights can be booked by same individual	Numeric
41	maximum_nights	Maximum nights can be booked by same individual	Numeric
42	minimum_minimum_nights	Same values as Minimum nights	Numeric
43	maximum_minimum_nights	Same values as Maximum_nights	Numeric
44	minimum_maximum_nights	Same values as Maximum_nights	Numeric
45	maximum_maximum_nights	Same values as Maximum_nights	Numeric
46	minimum_nights_avg_ntm	Average minimum nights can be booked by same individual	Numeric
47	maximum_nights_avg_ntm	Average maximum nights can be booked by same individual	Numeric
48	calendar_updated	No values (N/A)	N/A
49	has_availability	True or False if listing is available	Boolean
50	availability_30	Availability of Property in 30 days	Numeric

51	availability_60	Availability of Property in 60 days	Numeric
52	availability_90	Availability of Property in 90 days	Numeric
53	availability_365	Availability of Property in 365 days	Numeric
54	calendar_last_scraped	Date last scraped	Numeric
55	number_of_reviews	Total number of reviews that a listing has received from customers	Numeric
56	number_of_reviews_ltm	The number of reviews that a listing has received last twelve month	Numeric
57	number_of_reviews_l30d	The number of reviews that a listing has received per 130 days	Numeric
58	first_review	Date of first review by customer	Numeric
59	last_review	Date of last review by customer	Numeric
60	***review_scores_rating***	***Customer-provided score rating (0% to 100%); A customer-provided review score attributed to a listing based on overall experience and satisfaction***	Numeric
61	review_scores_accuracy	Accuracy of review scores (0 to 10)	Numeric
62	review_scores_cleanliness	Cleanliness score (0 to 10)	Numeric
63	review_scores_checkin	Over-all check in score (0 to 10)	Numeric
64	review_scores_communication	Score on communication with host (0 to 10)	Numeric
65	***review_scores_location***	***Score on location based on factors such as nearby transportation, noise level (0 to 10)***	Numeric
66	review_scores_value	Over- all value/quality/experience (0 to 10)	Numeric
67	license	No values (N/A)	N/A
68	instant_bookable	True or False if customer can instantly book	BOolean
69	calculated_host_listings_count	Calculated number of listings by host	Numeric
70	calculated_host_listings_count_entire_homes	Number of host listings which are entire homes	Numeric
71	calculated_host_listings_count_private_rooms	Number of host listings which are private rooms	Numeric
72	calculated_host_listings_count_shared_rooms	Number of host listings which are shared homes	Numeric

73	reviews_per_month	Number of Customer reviews of the host accommodation or accommodations per month	Numeric

In Slate> Content> 0:Rubrics

Criteria	Description	Deductions	Overall Marks
Problem statement	- A statement which summarizes what your business problem/opportunity is - Must include the financial measure of success	- Problem statement is too vague - No tie to the business - No financial measure of success stated	/10
Stakeholders	- A description of who this solution will impact	- Section is missing completely	/5
Data-set review	- A description of where the data-source originated from	- No description of where the data originated from - No description of each field	/20
	- A description of each data field and its description (for example: Start_time: A date/time flag of the transaction) - Include a link to the data source if online		
Data-science approach	- A description on how you plan to solve the problem, such as a regression or predictive model - This section is not meant to be very detailed as you did not start the project yet	- Section is missing completely	/5
Deductions	Overall marks will be deducted for: - Report lacks professionalism and quality - Report lacks a business perspective		

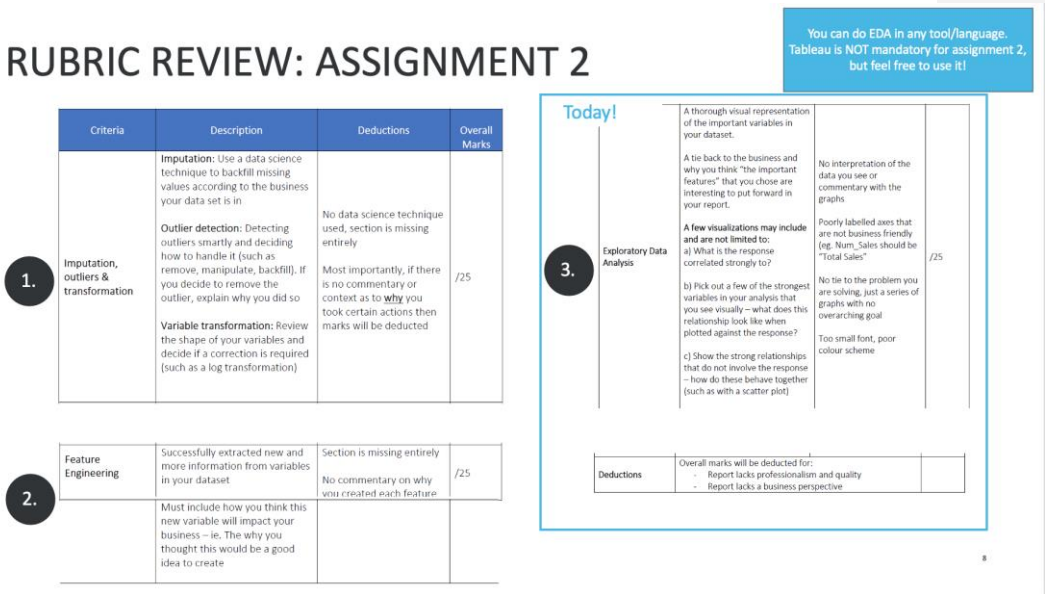
DELIVERABLE 2 : CLEANING AND VISUALIZATION REQUIREMENTS :October 14

-Imputation, Outliers and Transformation

-Exploratory Analysis with visualization

-Selection of relevant features

-Feature engineering



Airbnb Toronto Datascience Project

Team 4

Deliverable 2: Data Exploration and Data cleaning

Sheridan College

Date: Oct. 28th/2021

Table of Contents

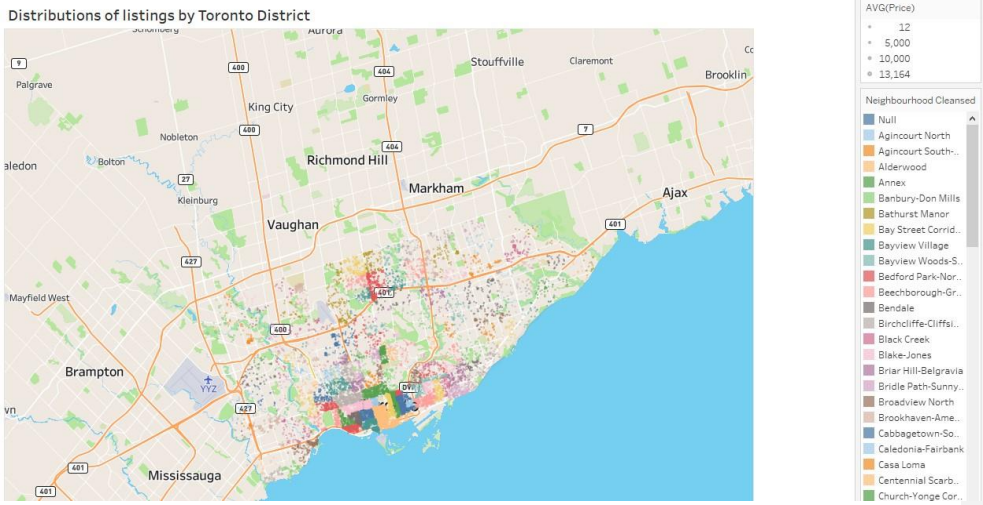
TEAM 4-DELIVERABLE 2 : Data cleaning and Visualization Report (Submitting: Oct 28, 7 pm)

Purpose of Deliverable 2

The goal of Deliverable 2 of our project was to prepare a clean Toronto Airbnb dataset and visualizations that would enable stakeholders to understand why essential steps characteristic of data cleaning such as the removal of features and imputation took place. As noted in our earlier report, we believed that features in the dataset that are predictive of price of stay per night and review ratings score would enable the Airbnb company president and relevant stakeholders such as Airbnb Hosts to develop business strategies such as marketing strategies to emphasize these features when providing Airbnb service to existing and potential customers. As a result there would be a dramatic increase in Airbnb revenue based on the number of listings rented. We utilized the programming language, Python for the steps required in Data cleansing and Tableau visualization software (version version 2021.3) was used for the Exploratory data analysis (EDA).

The Toronto Airbnb dataset presented a challenge in the data cleaning process for four main reasons. There were several features comprising 74 features (including listing ID) or variables in the dataset. On closer examination, we identified features such as listing_url, that were not relevant to our business objective and we noted other features that contained data that was redundant across features. Furthermore, features were removed because they contained more than 40% missing values. Finally, the complexity of the dataset can be noted from the total number of 19343 host listings in the dataset, and shown below visually is a map of Toronto displaying host listings in each neighborhood.

Figure 1. Listings by Neighbourhood in Toronto as displayed in Tableau



Overview of the data cleaning process and relevance of EDA

Exploratory analysis as in EDA of the original dataset (uncleaned) was conducted by use of Tableau to generate graphs and plots to identify relevant features and the relationships between the features (refer to Part III of this report). Secondly, as is common with real world data, several features contained missing values and as a result imputation methods were implemented in Python to replace the missing values. Thirdly, outliers were identified in the dataset using both Tableau and Python. We detected both negative and positive skewness in several features of the dataset. The interquartile range(IQR) method was selected to remove the outliers.

Finally, we feature engineered three features that would be relevant to our business objective. One particular variable that was introduced in the dataset was 'former city' referred to as Toronto district. There were more than 40 neighbourhoods as a result we web scraped using python the wikipedia page (https://en.wikipedia.org/wiki/Toronto), under the section titled "Neighborhoods" to collect all the districts and associated neighbourhoods in Toronto. Using this method, we were able to group the neighbourhoods by district. *Figure 1* below displays a sample list of neighbourhoods grouped by Toronto districts (i.e., 'former_city').

```
[5]: df2[['price','review_scores_rating','neighbourhood_cleansed','former_city']].sample(10)
```

[5]:

	price	review_scores_rating	neighbourhood_cleansed	former_city
13980	$250.00	NaN	Cabbagetown-South St.James Town	Old City of Toronto
13795	$75.00	60.0	Leaside-Bennington	East York
16291	$80.00	NaN	Lawrence Park South	Old City of Toronto
17805	$45.00	NaN	Dorset Park	Scarborough

To select relevant features and verify the data cleaning we used Tableau for exploratory analysis (EDA) for both uncleaned and cleaned datasets. Python was used to generate the cleaned dataset that would be used for EDA in Tableau for generation of the dashboard and for implementation of machine learning model. Therefore, in this report we provide in detail the steps taken to clean our Airbnb dataset using Python (Part 1 & Part II) and the necessary EDA in Tableau for the uncleaned dataset (Part III).

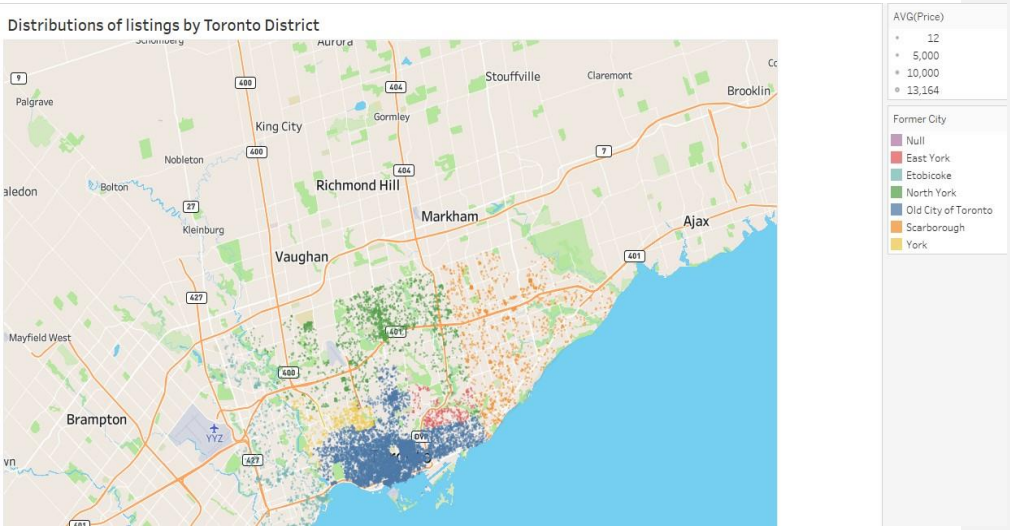
Part I

Steps in the removal of features

To begin, our EDA using Tableau to characterize price variation by neighborhood or review ratings score by neighborhood was difficult to interpret as there are more than 70 neighbourhoods in Toronto. We web scraped using python, the wikipedia page for the city of Toronto to collect all the districts and relevant neighbourhoods and as a result we were able to group the neighbourhoods by district. A merge function was implemented in Python to combine the web scraped table with our uncleaned dataset. As a result, a new feature called 'Former City' was introduced into the Toronto Airbnb dataset. For the feature 'Former City', the Toronto districts were Etobicoke, North York, East York, Old City of Toronto, York and Scarborough. Each of these Toronto districts have more than five neighbourhoods, with Old City of Toronto containing the most neighbourhoods in Toronto. For example, East York encompasses Broadview North, Danforth East York, Leaside-Bennington, O'Connor-Parkview, Old East-

York, Thorncliffe Park and Woodbine-Lumsden. In *Figure 2*, below you can see how listings can be viewed better by district rather than by neighborhood as shown with the uncleaned dataset in Tableau.

Figure 2. Listings by District in Toronto



In *Table 1* in the Appendix, we have highlighted features in the original or uncleaned dataset in terms of the basis of removal (i.e., irrelevant, too many missing values and redundancy). There were 73 features (Id being a primary key) in the uncleaned dataset. Features were removed due to redundancy and irrelevance to our business objective and more than 40% missing values. We have also highlighted features that had NaN values.

We used Python to reduce the 73 features to a set of features that were relevant to our machine learning model. We first did a correlation plot with the target variable being price. *Figure 2 in Appendix* displays the output of such correlations. In general, there were few features correlated to price, highlighting the importance of data cleaning. Our second step in data cleaning involved removing features not relevant to our business objective. We removed id, listing_url, scrape_id, last_scraped, name, description,neighbourhood_overview, picture_url, host_id, host_url, host_name, host_thumbnail_url. Textual information such as description and url web links were features which were removed. Features such as 'minimum_nights_avg_ntm' and 'maximum_nights_avg_ntm' were removed because of redundancy in the data.That is, both features had duplicate values. Thirdly, we removed features for which there were no values. Fourthly, we removed features that had more than 40% missing values. These features were 'host_about', 'neighbourhood_group cleansed','bathrooms', 'calendar_updated', 'license'. The 40% missing value threshold is what most businesses use for removal of missing values in datasets. Refer to the figure below for the output from Python, in terms of percentage of missing values.

```
Columns with more than 40% missing values:
 ['host_about' 'neighbourhood_group_cleansed' 'bathrooms'
 'calendar_updated' 'license']
```

Imputation of missing values in Categorical features

As a result of the above steps of data cleaning, we had 36 features in our dataset. There were ? categorical, ? numeric features. Refer to *Figure 3* in Appendix for profile of features with data type. To replace NaN values in the categorical features with data type of object, we imported the SimpleInputer from Sklearn Python library to impute missing values with most frequent. For example, host response time, host response rate and host acceptance rate were features for which NaN values were imputed.

```
df.head(3)
```

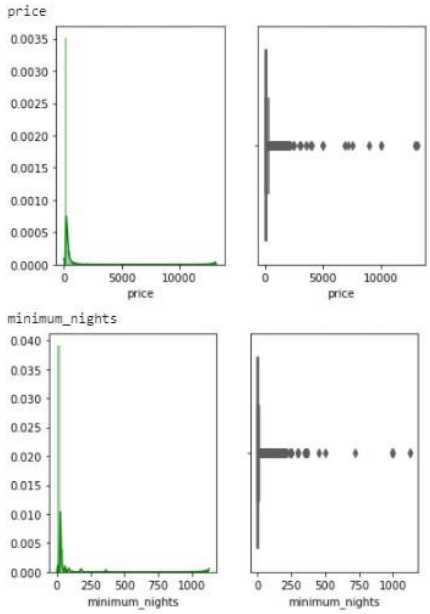
	room_type	host_since	host_response_time	host_is_superhost	host_acceptance_rate	host_response_rate
0	Entire home/apt	2008-08-08	NaN	f	NaN	NaN
1	Entire home/apt	2011-06-07	NaN	f	NaN	NaN
2	Entire home/apt	2012-06-01	NaN	t	100%	NaN

```
df.head(3)
```

	room_type	host_since	host_response_time	host_is_superhost	host_acceptance_rate	host_response_rate
0	Entire home/apt	2008-08-08	within an hour	f	100%	100%
1	Entire home/apt	2011-06-07	within an hour	f	100%	100%
2	Entire home/apt	2012-06-01	within an hour	t	100%	100%

Outlier detection and transformation of data

As can be seen from the box plot and histogram distributions generated in Python and for the selected features in Tableau, displayed in *Figure 4* in appendix*;* most of the data contained within features was positively or negatively skewed. Several outliers were present within each feature. The outliers were noted from use of describe function and confirmed from the box plots and visualization of the distributions. For example, in the unclean or original dataset; for price, the maximum value is $13164, but the average price is $141.28 and 75% are below $150. Similarly for minimum_nights, the maximum value is 1125, but the average is 10 and 75% below 5. For both bedrooms and beds, mean was 1.39 and 1.63 respectively; maximum being 16 and 17. Outliers were detected in the features: price, accommodates, host_total_listings_count, reviews per month, bedrooms, beds, minimum_nights, maximum_nights, number of reviews, number_of_review_ltm, review_scores_accuracy, review_scores_cleanliness, review_scores_checkin, review_scores_communication, review_scores_value, reviews_per_month.



Three options for removing outliers were available to us, the Interquartile range (IQR) method, Winsorize and Log transform. Log transformation reduces the skewness of data and tries to make it normal. However we had zero or negative values contained within some of our features. With winsorizing, any value of a variable above or below a percentile k on each side of the variables' distribution is replaced with the value of the k-th percentile itself. For example, 90% winsorization means the replacement of the top 5% and bottom 5% of the data. The top 5% of the data is replaced by the value of the data at the 95th percentile and the value of the bottom 5% of the data is replaced by the value of the data at the 5th percentile. However, winsorization would not be feasible as it could not be applied properly to particular features in our data and would replace values in the dataset.

For the above reasons, we selected the IQR method. For the IQR method, the third quartile (75th percentile) and first quartile (25th percentile) is subtracted to get the IQR. Any numbers less than the First quartile subtracted from 1.5 x IQR is considered an outlier and removed; whereas any number greater than the third quartile subtracted from 1.5 x IQR is considered an outlier and removed. We applied the IQR method using Python, to all the numerical features in our dataset to remove outliers. The Figure below shows outliers in Price and Minimum nights and after removal of outliers, using the describe function. The dataframe named 'df4' contained features such as price with outliers, whereas the data frame named 'df' contains features such as price without outliers.

```
df4[['price', 'review_scores_rating', 'minimum_nights', 'host_total_listings_count']].describe()
```

	price	review_scores_rating	minimum_nights	host_total_listings_count
count	19343.000000	15010.000000	19343.000000	19339.00000
mean	141.278116	94.304930	9.988730	5.52283
std	290.664182	8.899373	37.022953	15.39491
min	12.000000	20.000000	1.000000	0.00000
25%	63.000000	93.000000	1.000000	1.00000
50%	100.000000	97.000000	2.000000	1.00000
75%	150.000000	100.000000	5.000000	4.00000
max	13164.000000	100.000000	1125.000000	272.00000

```
df[['price', 'review_scores_rating', 'minimum_nights', 'host_total_listings_count']].describe()
```

	price	review_scores_rating	minimum_nights	host_total_listings_count
count	19343.000000	19343.000000	19343.000000	19343.000000
mean	117.516891	95.257847	3.834876	2.742685
std	70.959498	4.043789	3.598508	2.770920
min	12.000000	86.500000	1.000000	0.000000
25%	63.000000	94.000000	1.000000	1.000000
50%	100.000000	95.000000	2.000000	1.000000
75%	150.000000	99.000000	5.000000	4.000000
max	280.500000	100.000000	11.000000	8.500000

Part II

Feature Engineering

Table 2 in the appendix represents the features in the cleaned dataset that were feature engineered. To extract some useful features from our dataset we converted two relevant features (host response rate and host acceptance rate) to categorical features, by the use of binning and labels. We felt the response rate of the host would have a strong influence on both price and review ratings score. Host response rate refers to the host responding to the request of the client or customer before renting airbnb, during the stay in airbnb. The requests from clients could be on inquiries on the host listing location and characteristics of the rental space. Since, price fluctuated with neighbourhood or Toronto district, we felt there would be a relationship with the response rate of the host. If there was a higher price for rent, it was likely that the host was responding on time to the inquiries of the customer. A more direct relationship would be between review ratings score and response rate of host. We felt customers would give a higher score on review ratings if the host responded often to their inquiries before the transaction and during the rental period. Host response rate was formatted with % sign removed and was binned into 5 bins and labels were assigned as in very strict, strict, accepting and very accepting. For host_response_rate, 5 bins were also created with labels ‘very slow’, ‘slow’, ‘fast’, ‘very fast’.

Another categorical feature that was feature engineered, was host acceptance rate. It was defined as an ‘object’ variable in python, with numbers ranging from 100% to 50%, 30% and

0%. We felt in locations of Toronto and for listings where hosts showed a low acceptance rate there would be a strong relationship with both price of rent and review ratings score. Perhaps with a high acceptance rate, the price of rent was low which may reflect the quality of the rental space. In terms of review ratings score, perhaps higher review ratings score was provided to hosts who had a high acceptance rate.

Finally, we thought that a factor that a customer would take into account when choosing to rent an airbnb in Toronto would be the length of time the host has been a host of an airbnb. Perhaps the customer considered the experience of the host as measured by the number of years the individual has been an Airbnb host as an important factor in renting the host's airbnb. We split the date into year under the column "Host_since_year" and month using Python and then subtracted the current year being 2021 from "Host_since_year" to get the new feature "Host_length".

Figure. Display of sample values in new features

```
df[['host_acceptance_rate','labels_host_acceptance_rate','host_response_rate','labels_h
```

	host_acceptance_rate	labels_host_acceptance_rate	host_response_rate	labels_host_response_rate	host_since_year
0	100	Very Accepting	100	Very high	2008
1	100	Very Accepting	100	Very high	2011
2	100	Very Accepting	100	Very high	2012
3	100	Very Accepting	100	Very high	2012
4	100	Very Accepting	100	Very high	2012
...	...	...	...	...	...
19338	100	Very Accepting	100	Very high	2019
19339	100	Very Accepting	100	Very high	2020
19340	94	Very Accepting	100	Very high	2017
19341	75	Accepting	100	Very high	2019
19342	100	Very Accepting	100	Very high	2020

19343 rows × 5 columns

Imputation of missing values in Numerical features

There were NaN values present in some of our numerical features. We have highlighted these features in *Table 1*. Using Python code, we imported the library called IterativeImputer to enable an iterative imputer from skitlearn to impute NaN values in the numerical features in the Airbnb dataset, with the prediction method. The Iterative Imputer models the missing values based on a prediction compared to the other features in the data set. i.e., known variables are used as a train set and used to predict the test (missing) rows. The profile of the cleaned dataset with features is shown in *Figure ?*.

Dummy coding categorical variables

In anticipation of the implementation of Machine learning algorithms as in Clustering and Regression of the data, categorical features were dummy coded using the get_dummies function

in Python Pandas. The complete set of features including features which are dummy coded are presented in Figure ? in Appendix.

Part III

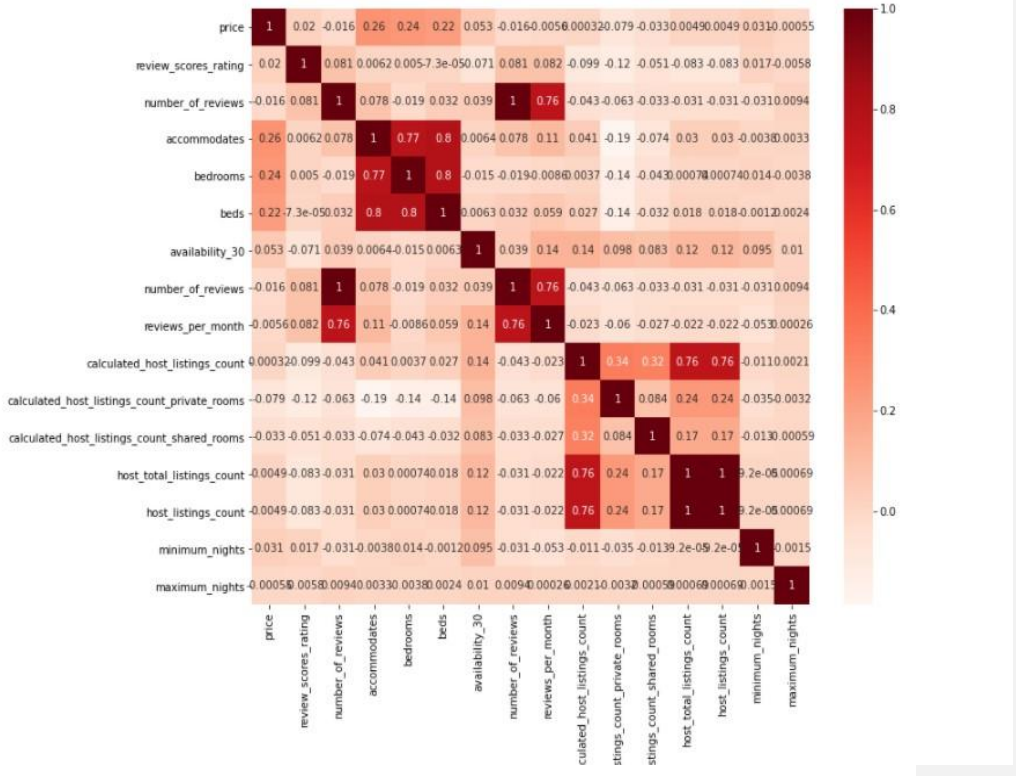
Exploratory Data Analysis

Using Tableau we conducted an exploratory data analysis of both the original and cleansed Airbnb datasets.

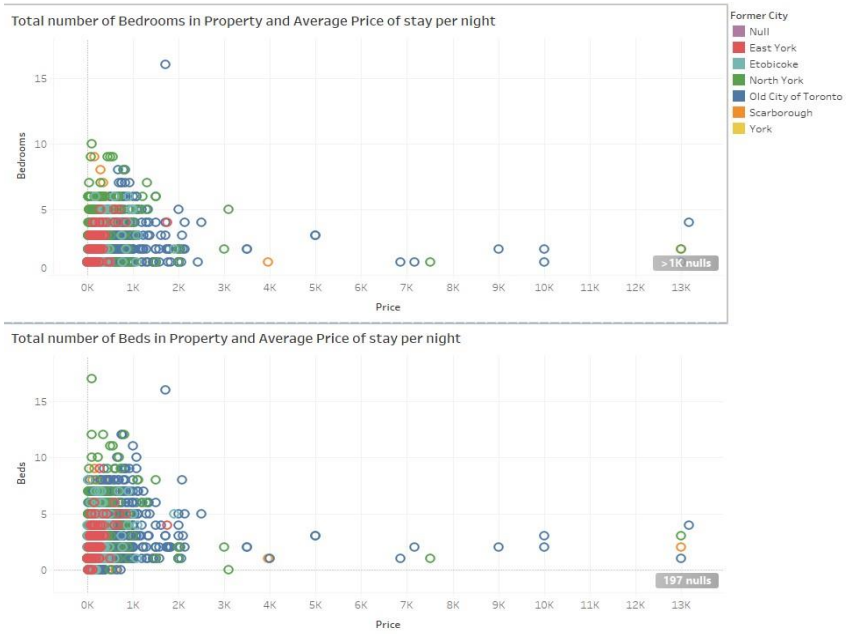
EDA of original dataset

Before conducting the explanatory analysis of the original dataset, we needed to assess which variables or features were most correlated with our target variable being price and the other target variable being review scores rating. Our correlation plot shown in the figure below, generated in Python showed 'accommodates' (r = .26), 'bed' (r = .22) and 'bedrooms' (r = .24) to be the three features that were more correlated to 'price' than other features. What was also interesting was that 'review ratings score' had almost no relationship with 'price' (r = .02).

Figure: Correlation plot

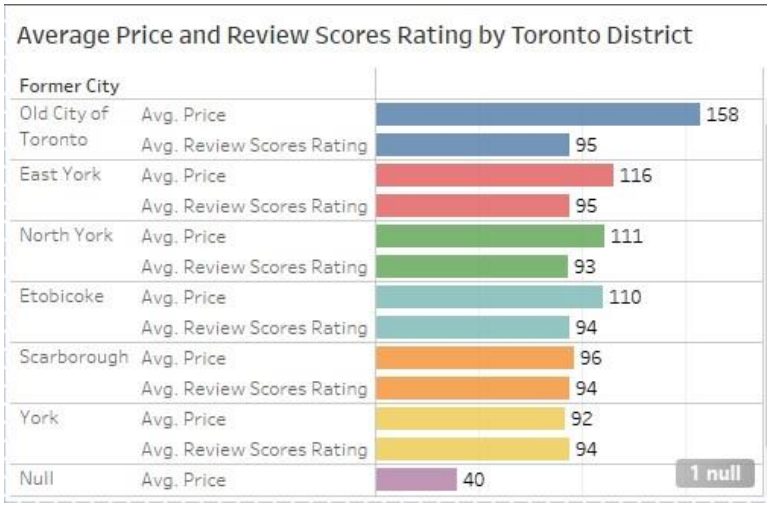


Our EDA of the original data also showed there were missing values or NaN values for most of the features and there were outliers for most of the features, even in the target variables as in Price or Review Ratings Score. Our EDA in Tableau showed why Price was most strongly correlated to ‘accommodates’, ‘bed’ and ‘bedrooms’.

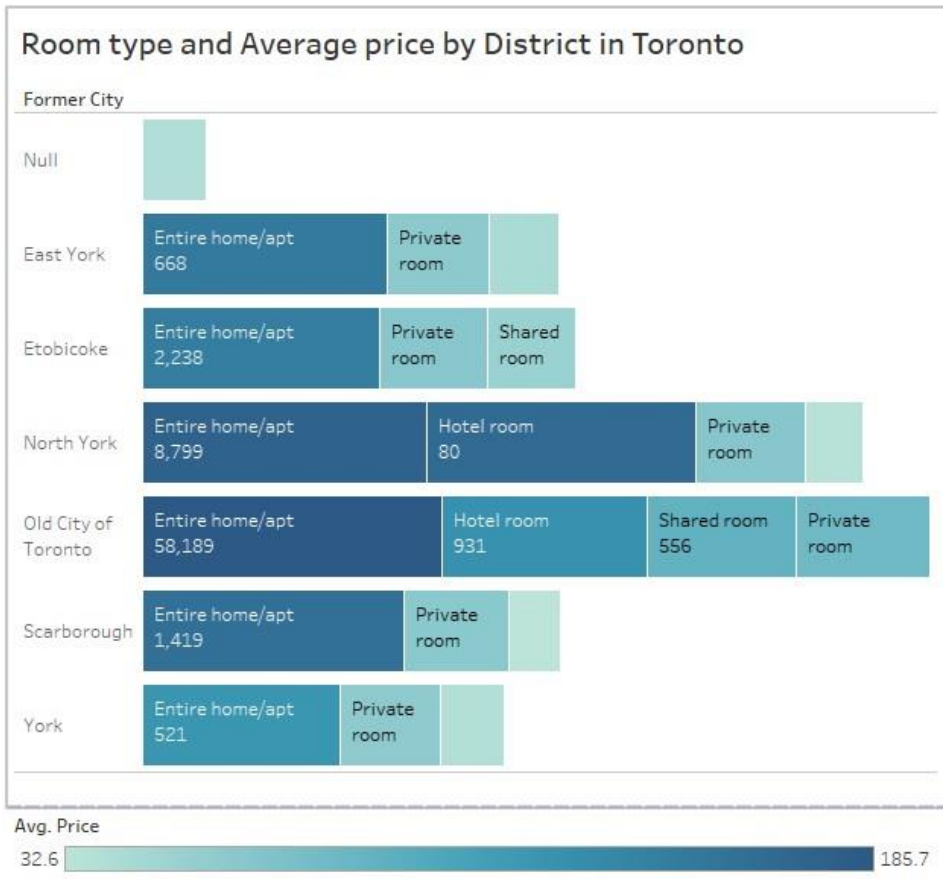


However, correlation does not measure the relationship between numerical and categorical variables. As a result several categorical features could relate to average price or review ratings score. One particular feature that could also relate to price or review ratings score was the location of the Airbnb listing. When we plotted in Tableau the listings by neighbourhood, a bird's eye view gave us the impression that in the district called "Old City Toronto" there were more listings compared to other districts such as Scarborough and York as can be seen from *Figure 2*, combined with the fact that price seemed higher for listings in location close to waterfront neighborhood in Old City of Toronto than areas further away from the waterfront neighborhood, such as North York and Etobicoke.

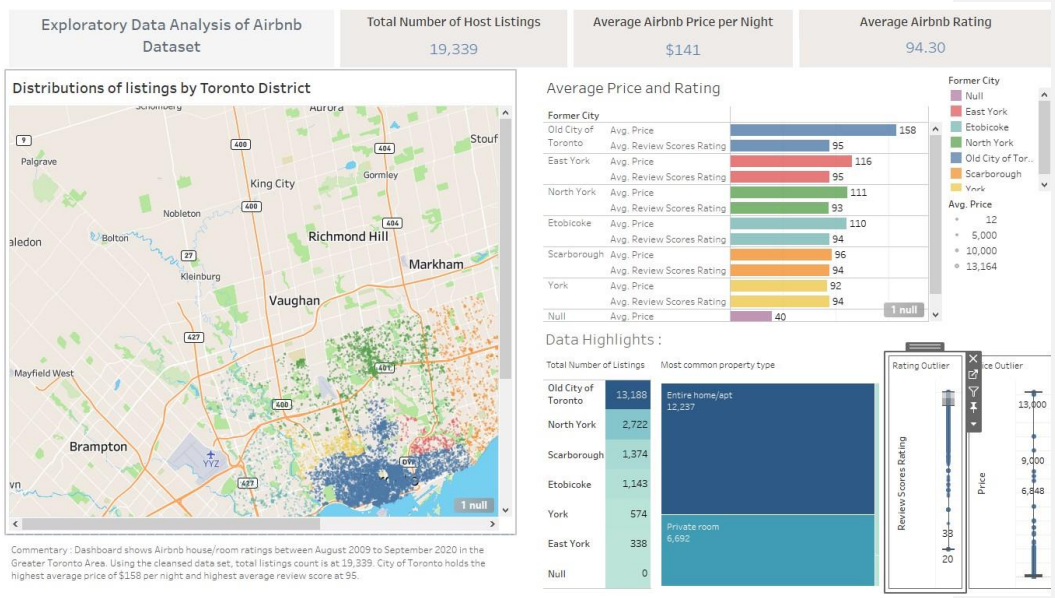
As a result, our first business query was does Average price and rating vary by the location of the Airbnb listing? As can be seen from the Figure below, price per day was much higher for listings in the Old City of Toronto ($158) compared to other districts. Lowest average price per listing was for listings in York ($92) and Scarborough ($96) which were located furthest from Old City of Toronto. Interestingly, review scores rating did not vary by location of listing.



Our second business query was does average price and total host listings also vary by the room type of the property and by the district in Toronto? From our EDA of the unclean data we found that Average price of listing in a district could also depend on the type of room (i.e., Entire home/apartment, Hotel room, Private room and Share room) and in the district in Toronto. As seen from the figure below, there were more listings of Entire home/apartment in the Old City of Toronto, North York and Scarborough compared to other districts. Moreover, the Old City of Toronto seemed to have listings which offered a more diverse distribution of room type. Finally, the average price of Entire home/apartment was the highest in the Old City of Toronto and lower in York, Etobicoke, and East York.

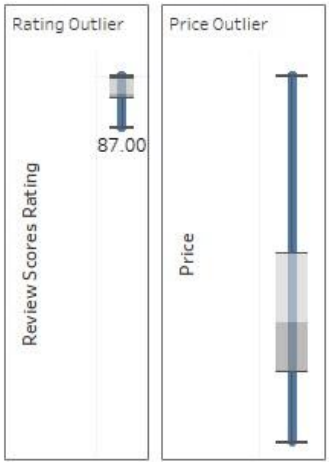


The figure below displays the dashboard with the uncleansed data being the data source and shows Airbnb house/room ratings between August 2009 to September 2020 in the Greater Toronto Area. Using the cleansed data set, total listings count is at 19,339. Old City of Toronto holds the highest average price of $158 per night and highest average review score at 95. Outliers are shown in the rating and price outlier boxplots.



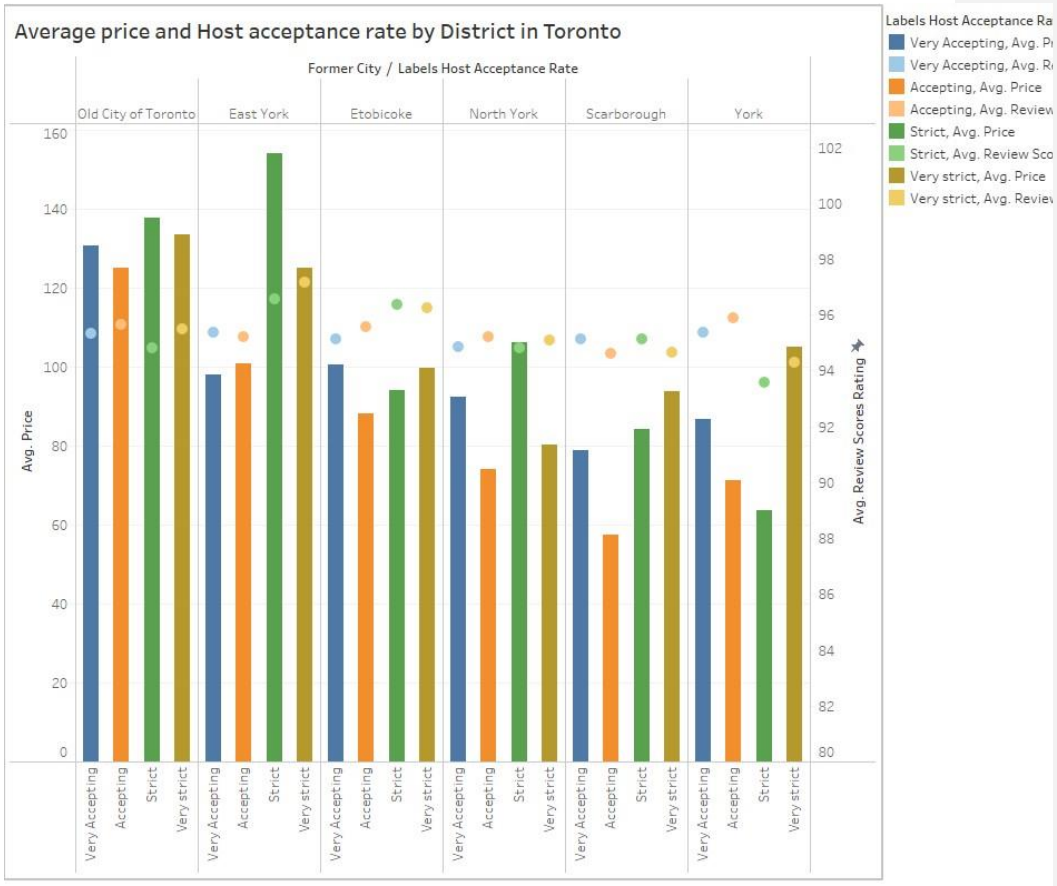
Visualization from Tableau of Clean data

We also did an EDA of the cleaned Airbnb dataset, however the details of the EDA will be provided in deliverable 4 for which dashboards of the cleaned dataset will be presented. The main purpose of showing a brief EDA of the clean data in this report was to provide evidence that missing values and outliers in most of the features were removed using data cleaning processes such as imputation in Python. For example, there were no outliers for both review ratings score and price after the data cleaning process as evidenced by the figure below from visualization in Tableau.



Interestingly, the cleaned data did show that the general patterns present in the original dataset remained the same even after data cleaning. For example, the average price for a host Airbnb listing was still highest in Old City of Toronto compared to other districts of Toronto, as can be seen from the figure below.

We also did a visualization of the relationship between our two features: host response rate and host acceptance rate engineered using Python by the use of a binning process. As can be seen from the figures below, host acceptance rate does vary by district in Toronto. In addition, average price does seem to relate to host acceptance rate in certain districts of Toronto, but not other districts such as the Old City of Toronto.



Conclusion

Reference

1 https://www.kaggle.com/robinkongninglo/toronto-airbnb-dataset

Appendix

Table 1. Fields or features in the AirBnb Data Set with description and type of feature stated

Original Dataset features; In total there were 73 features in the original dataset that were not cleaned. Legend
Orange-Irrelevant features
Blue-Features with more than 40% NaN values
Red-Target features
Dark Magenta-Redundant features
Purple-Relevant features

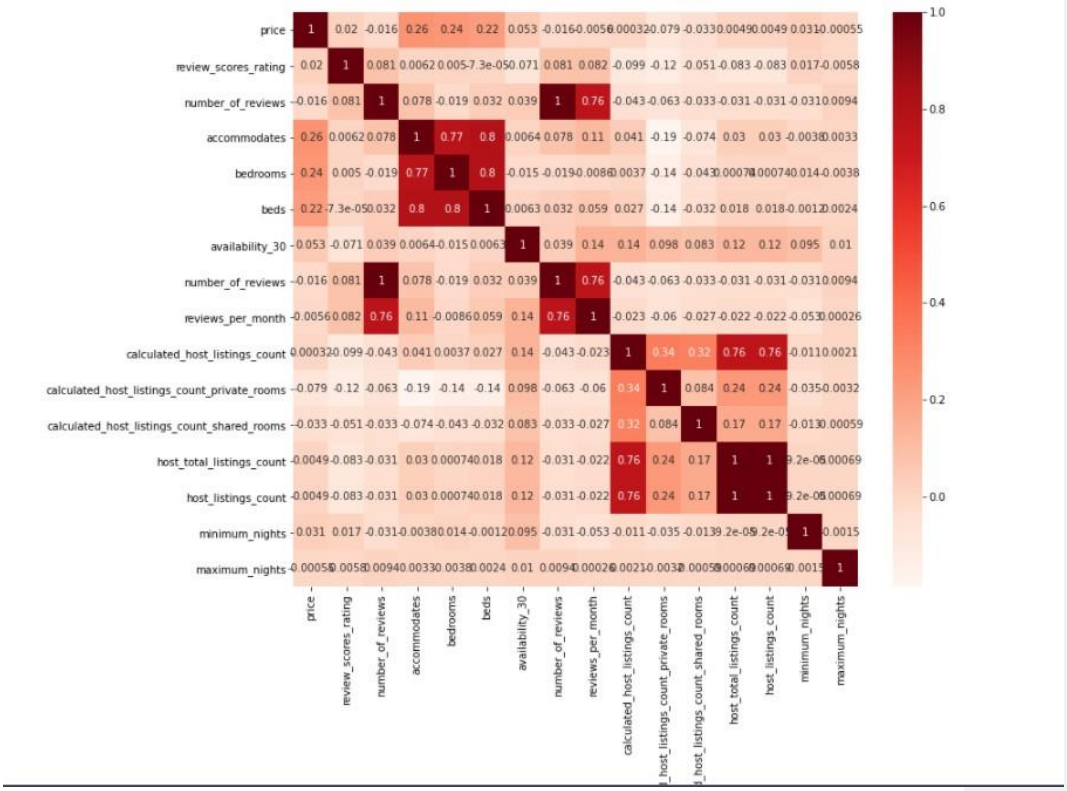
Number of Variables	Column Name	Description	Feature type(e.g., Numeric, String)
	id	Unique listing id (Primary Key)	Numeric
1	listing_url	Link to the rental property listing on Airbnb	String
2	scrape_id	Identifier for scraper	Numeric
3	last_scraped	Last date listing was scraped	Numeric
4	name	Title of Posting	String
5	description	Description of Posting	String
6	neighborhood_overview	Overview of neighborhood	String
7	picture_url	Link to the main vacation rental listing image on Airbnb	Image
8	host_id	Unique id for each host	Numeric
9	host_url	Link to the host profile on Airbnb	String
10	host_name	Host Name	String
11	host_since	Date an individual became a host	Numeric
12	host_location	Location of Listing	String
13	host_about	Description of host - relationship status, interests and hobbies	String

14	*host_response_time*	Time to respond to customer booking inquiry; ranges from 1hour to few days	Numeric
15	*host_response_rate*	Ranges from 0% to 100% for reply to booking inquiries	Numeric
16	*host_acceptance_rate*	Ranges from 0% to 100% response for acceptance of booking	Numeric
17	host_is_superhost	Is either t(true) or f(false)	Boolean
18	host_thumbnail_url	Host thumbnail URL	Image
19	host_picture_url	Host Picture URL	Image
20	host_neighbourhood	Description of neighborhood listing	String
21	host_listings_count	Current number of host listings	Numeric
22	host_total_listings_count	Total number of listings made by host	Numeric
23	host_verifications	Identifies how the host has completed the identity verification process	String
24	host_has_profile_pic	Host Profile Pic	Image
25	host_identity_verified	Identifies if the host has completed the verification process by indicating true or false	Boolean
26	neighbourhood	Specific location of Toronto area	String
27	neighbourhood_cleansed	Represents one of boroughs in Toronto in which a listing resides	String
28	neighbourhood_group_cleansed	No values (N/A)	N/A
29	latitude	The angular distance of a location or object north or south of the Earth's celestial equator	String
30	longitude	The angular distance of a location or object east or west of the meridian	String
31	property_type	Type of property (e.g., Entire house)	String
32	room_type	Specific type of room (e.g., Entire home/apt)	String
33	accommodates	How many people can stay (e.g., 6)	Numeric
34	bathrooms	No values, NA	N/A
35	bathrooms_text	Number of bathrooms in property	Numeric

36	bedrooms	Number of bedrooms in property	Numeric
37	beds	Number of beds in property	Numeric
38	amenities	List of amenities such as shampoo available in property	String
39	*price*	Price of stay per night	Numeric
40	minimum_nights	Minimum nights can be booked by same individual	Numeric
41	maximum_nights	Maximum nights can be booked by same individual	Numeric
42	minimum_minimum_nights	Same values as Minimum nights	Numeric
43	maximum_minimum_nights	Same values as Maximum_nights	Numeric
44	minimum_maximum_nights	Same values as Maximum_nights	Numeric
45	maximum_maximum_nights	Same values as Maximum_nights	Numeric
46	minimum_nights_avg_ntm	Average minimum nights can be booked by same individual	Numeric
47	maximum_nights_avg_ntm	Average maximum nights can be booked by same individual	Numeric
48	calendar_updated	No values (N/A)	N/A
49	has_availability	True or False if listing is available	Boolean
50	availability_30	Availability of Property in 30 days	Numeric
51	availability_60	Availability of Property in 60 days	Numeric
52	availability_90	Availability of Property in 90 days	Numeric
53	availability_365	Availability of Property in 365 days	Numeric
54	calendar_last_scraped	Date last scraped	Numeric
55	number_of_reviews	Total number of reviews that a listing has received from customers	Numeric
56	number_of_reviews_ltm	The number of reviews that a listing has received last twelve month	Numeric
57	number_of_reviews_l30d	The number of reviews that a listing has received per 130 days	Numeric
58	first_review	Date of first review by customer	Numeric
59	last_review	Date of last review by customer	Numeric

60	***review_scores_rating***	***Customer-provided score rating (0% to 100%); A customer-provided review score attributed to a listing based on overall experience and satisfaction***	Numeric
61	review_scores_accuracy	Accuracy of review scores (0 to 10)	Numeric
62	review_scores_cleanliness	Cleanliness score (0 to 10)	Numeric
63	review_scores_checkin	Over-all check in score (0 to 10)	Numeric
64	review_scores_communication	Score on communication with host (0 to 10)	Numeric
65	***review_scores_location***	***Score on location based on factors such as nearby transportation, noise level (0 to 10)***	Numeric
66	review_scores_value	Over- all value/quality/experience (0 to 10)	Numeric
67	license	No values (N/A)	N/A
68	instant_bookable	True or False if customer can instantly book	BOolean
69	calculated_host_listings_count	Calculated number of listings by host	Numeric
70	calculated_host_listings_count_entire_homes	Number of host listings which are entire homes	Numeric
71	calculated_host_listings_count_private_rooms	Number of host listings which are private rooms	Numeric
72	calculated_host_listings_count_shared_rooms	Number of host listings which are shared homes	Numeric
73	reviews_per_month	Number of Customer reviews of the host accommodation or accommodations per month	Numeric

---



<class 'pandas.core.frame.DataFrame'>

Int64Index: 19343 entries, 0 to 19342

Data columns (total 75 columns):

#	Column	Non-Null Count	Dtype
0	id	19343 non-null	int64
1	listing_url	19343 non-null	object
2	scrape_id	19343 non-null	int64
3	last_scraped	19343 non-null	object

4	name	19342 non-null	object
5	description	18623 non-null	object
6	neighborhood_overview	12364 non-null	object
7	picture_url	19343 non-null	object
8	host_id	19343 non-null	int64
9	host_url	19343 non-null	object
10	host_name	19339 non-null	object
11	host_since	19339 non-null	object
12	host_location	19329 non-null	object
13	host_about	10958 non-null	object
14	host_response_time	11814 non-null	object
15	host_response_rate	11814 non-null	object
16	host_acceptance_rate	13672 non-null	object
17	host_is_superhost	19339 non-null	object
18	host_thumbnail_url	19339 non-null	object
19	host_picture_url	19339 non-null	object
20	host_neighbourhood	15552 non-null	object
21	host_listings_count	19339 non-null	float64
22	host_total_listings_count	19339 non-null	float64
23	host_verifications	19343 non-null	object
24	host_has_profile_pic	19339 non-null	object
25	host_identity_verified	19339 non-null	object
26	neighbourhood	12364 non-null	object

27	neighbourhood_cleansed	19343 non-null	object
28	neighbourhood_group_cleansed	0 non-null	float64
29	latitude	19343 non-null	float64
30	longitude	19343 non-null	float64
31	property_type	19343 non-null	object
32	room_type	19343 non-null	object
33	accommodates	19343 non-null	int64
34	bathrooms	0 non-null	float64
35	bathrooms_text	19330 non-null	object
36	bedrooms	17914 non-null	float64
37	beds	19147 non-null	float64
38	amenities	19343 non-null	object
39	price	19343 non-null	object
40	minimum_nights	19343 non-null	int64
41	maximum_nights	19343 non-null	int64
42	minimum_minimum_nights	19343 non-null	int64
43	maximum_minimum_nights	19343 non-null	int64
44	minimum_maximum_nights	19343 non-null	int64
45	maximum_maximum_nights	19343 non-null	int64
46	minimum_nights_avg_ntm	19343 non-null	float64
47	maximum_nights_avg_ntm	19343 non-null	float64
48	calendar_updated	0 non-null	float64
49	has_availability	19343 non-null	object

50	availability_30	19343 non-null	int64
51	availability_60	19343 non-null	int64
52	availability_90	19343 non-null	int64
53	availability_365	19343 non-null	int64
54	calendar_last_scraped	19343 non-null	object
55	number_of_reviews	19343 non-null	int64
56	number_of_reviews_ltm	19343 non-null	int64
57	number_of_reviews_l30d	19343 non-null	int64
58	first_review	15278 non-null	object
59	last_review	15278 non-null	object
60	review_scores_rating	15010 non-null	float64
61	review_scores_accuracy	14976 non-null	float64
62	review_scores_cleanliness	14976 non-null	float64
63	review_scores_checkin	14974 non-null	float64
64	review_scores_communication	14978 non-null	float64
65	review_scores_location	14971 non-null	float64
66	review_scores_value	14972 non-null	float64
67	license	0 non-null	float64
68	instant_bookable	19343 non-null	object
69	calculated_host_listings_count	19343 non-null	int64
70	calculated_host_listings_count_entire_homes	19343 non-null	int64
71	calculated_host_listings_count_private_rooms	19343 non-null	int64
72	calculated_host_listings_count_shared_rooms	19343 non-null	int64

73 reviews_per_month 15278 non-null float64

74 former_city 19343 non-null object

dtypes: float64(20), int64(21), object(34)

memory usage: 11.2+ MB

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19343 entries, 0 to 19342
Data columns (total 37 columns):
 #   Column                                       Non-Null Count  Dtype
---  ------                                       --------------  -----
 0   host_total_listings_count                    19339 non-null  float64
 1   neighbourhood_cleansed                       19343 non-null  object
 2   latitude                                     19343 non-null  float64
 3   longitude                                    19343 non-null  float64
 4   accommodates                                 19343 non-null  int64
 5   bedrooms                                     17914 non-null  float64
 6   beds                                         19147 non-null  float64
 7   price                                        19343 non-null  float64
 8   minimum_nights                               19343 non-null  int64
 9   maximum_nights                               19343 non-null  int64
 10  availability_30                              19343 non-null  int64
 11  availability_60                              19343 non-null  int64
 12  availability_90                              19343 non-null  int64
 13  availability_365                             19343 non-null  int64
 14  number_of_reviews                            19343 non-null  int64
 15  number_of_reviews_ltm                        19343 non-null  int64
 16  number_of_reviews_l30d                       19343 non-null  int64
 17  review_scores_rating                         15010 non-null  float64
 18  review_scores_accuracy                       14976 non-null  float64
 19  review_scores_cleanliness                    14976 non-null  float64
 20  review_scores_checkin                        14974 non-null  float64
 21  review_scores_communication                  14978 non-null  float64
 22  review_scores_location                       14971 non-null  float64
 23  review_scores_value                          14972 non-null  float64
 24  instant_bookable                             19343 non-null  object
 25  calculated_host_listings_count               19343 non-null  int64
 26  calculated_host_listings_count_entire_homes  19343 non-null  int64
 27  calculated_host_listings_count_private_rooms 19343 non-null  int64
 28  calculated_host_listings_count_shared_rooms  19343 non-null  int64
 29  reviews_per_month                            15278 non-null  float64
 30  former_city                                  19343 non-null  object
 31  room_type                                    19343 non-null  object
 32  host_since                                   19343 non-null  object
 33  host_response_time                           19343 non-null  object
 34  host_is_superhost                            19343 non-null  object
 35  host_acceptance_rate                         19343 non-null  object
 36  host_response_rate                           19343 non-null  object
dtypes: float64(14), int64(14), object(9)
memory usage: 5.6+ MB
```

Cleaned Dataset features

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19343 entries, 0 to 19342
Data columns (total 41 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   bedrooms                                      19343 non-null  float64
 1   beds                                          19343 non-null  float64
 2   host_total_listings_count                     19343 non-null  float64
 3   review_scores_rating                          19343 non-null  float64
 4   review_scores_accuracy                        19343 non-null  float64
 5   review_scores_cleanliness                     19343 non-null  float64
 6   review_scores_checkin                         19343 non-null  float64
 7   review_scores_communication                   19343 non-null  float64
 8   review_scores_location                        19343 non-null  float64
 9   review_scores_value                           19343 non-null  float64
 10  reviews_per_month                             19343 non-null  float64
 11  neighbourhood_cleansed                        19343 non-null  object
 12  latitude                                      19343 non-null  float64
 13  longitude                                     19343 non-null  float64
 14  accommodates                                  19343 non-null  int64
 15  price                                         19343 non-null  float64
 16  minimum_nights                                19343 non-null  int64
 17  maximum_nights                                19343 non-null  int64
 18  availability_30                               19343 non-null  int64
 19  availability_60                               19343 non-null  int64
 20  availability_90                               19343 non-null  int64
 21  availability_365                              19343 non-null  int64
 22  number_of_reviews                             19343 non-null  int64
 23  number_of_reviews_ltm                         19343 non-null  int64
 24  number_of_reviews_l30d                        19343 non-null  int64
 25  instant_bookable                              19343 non-null  object
 26  calculated_host_listings_count                19343 non-null  int64
 27  calculated_host_listings_count_entire_homes   19343 non-null  int64
 28  calculated_host_listings_count_private_rooms  19343 non-null  int64
 29  calculated_host_listings_count_shared_rooms   19343 non-null  int64
 30  former_city                                   19343 non-null  object
 31  room_type                                     19343 non-null  object
 32  host_since                                    19343 non-null  object
 33  host_response_time                            19343 non-null  object
 34  host_is_superhost                             19343 non-null  object
 35  host_acceptance_rate                          19343 non-null  int64
 36  host_response_rate                            19343 non-null  int64
 37  labels_host_acceptance_rate                   19343 non-null  category
 38  labels_host_response_rate                     19343 non-null  category
 39  host_since_year                               19343 non-null  int64
 40  host_length                                   19343 non-null  int64
dtypes: category(2), float64(14), int64(18), object(7)
memory usage: 5.8+ MB
```

Airbnb Toronto Datascience Project

Team 4

Deliverable 3: Modeling

Sheridan College

Date: Nov.22nd/2021

Table of Contents

Objective of Deliverable 3

The goal for Deliverable 3 was to create linear regression models to address the business case developed in Deliverable 1 with the clean dataset created in Deliverable 2. The first model would predict the price of an AirBnb rental listing or unit per day while the second would predict the review score. In addition, we needed to find the best model based on evaluation metrics that would predict price and review score.

The two models would serve two primary groups - Stakeholders at Airbnb and Airbnb Hosts. Stakeholders at Airbnb would use the models to develop business strategies that maximize features in the model that are predictive of higher prices and better reviews. For example, a finding that the location of listing positively affects price could be used to promote listings in areas where prices are higher. Additionally, a new Airbnb host could use these models to help set the price of their unit while giving them insights into how to maximize their review score.

Overview of Steps in Development of Linear Regression Models of Price and Review Scores rating

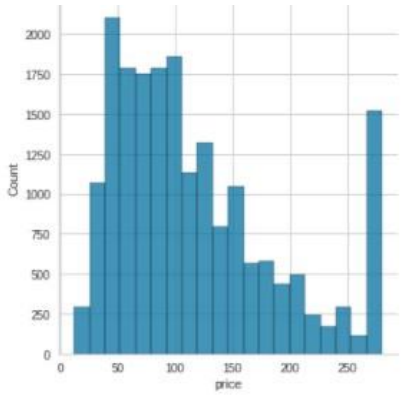
The cleaned data generated from Deliverable 2 was utilized by the team to produce eight regression models by four types of Regression supervised method, which predicted the price and review score for rental units in Toronto. Also in Deliverable 3 are insights regarding what factors have the greatest impact and suggestions for improving the model over time.

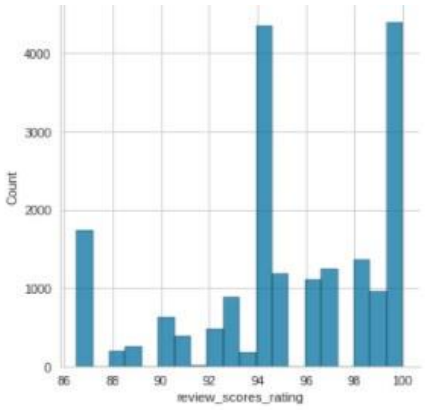
The modeling process followed a systematic set of steps. We first conducted a brief EDA of the cleaned dataset generated from Deliverable 2 to identify correlations between the features with target variables being price and review ratings score. Our intent was to determine if there were certain features predictive of price or review ratings score. After dummy coding categorical variables in our clean dataset, we implemented a standard regression to identify the contribution of features as predictors to price or review ratings score. We then implemented a machine learning based linear regression with either price or review ratings score as being the target variables. Next, a random forest regression method was implemented, as we felt the complexity of our dataset would necessitate a more sophisticated machine learning method where decision trees were present. Finally, we implemented the pycaret package to identify which machine learning regression model would be best suited for prediction of our target variables. The Linear gradient boost regression (LGBM) and Gradient boost regression (GBM) were identified as the best model for price and review ratings score respectively, based on metrics of R-squared score and Root Mean Squared Error (RMSE). Thus, we implemented LGBM on price and GBM on review ratings score as targets. The results showed that there was a significant difference in model prediction for price compared to review ratings score.

Importing Packages and brief EDA

Relevant packages were imported for Linear regression from Statsmodel and Sklearn library to conduct standard linear regression and machine learning of the data. As presenting the description of the data types and correlations would take up space, we present the figures in the appendix. In the cleaned dataset as noted from *Figure 1* in appendix, there were 39 features, 14 of float data type, 16 of integer and 8 of the object type. We conducted a correlation analysis of price with other features in the dataset. Our correlation table shown in the *Figure 2* in Appendix below, generated in Python showed 'accommodates' (r = .56), 'bedrooms' (r = .54), beds (r = '0.49') to be the three features that were more correlated to 'price' than other features. What was also interesting was that 'review ratings score' had almost no relationship with 'price' (r = .08). We subsequently conducted a correlation analysis of review ratings score with other features in the dataset. Our correlation table shown in the *Figure 3* in Appendix, generated in Python showed 'review_scores_cleanliness' (r = .65), 'review_scores_accuracy' (r = .64), 'review_scores_value' (r = 0.64) to be the three features that were more correlated to 'price' than other features. However, 'review ratings score' had almost no relationship with 'price' (r = .08).

Another aspect of the data which was relevant to the output of the regression models was the difference in distributions in data by 'price' and 'review_ratings_score'. As seen from the figures below, Price has more of a normal distribution, while review score is skewed to the right.





Standard Linear regression

The purpose of conducting standard linear regression, which is a form of EDA leading into Machine Learning was to determine the contribution of predictors to target variable.Two metrics are relevant to standard linear regression, p-value < .05 as significance of contribution of predictor to variance in dependent variable and R-squared which is the proportion of variance in the dependant variable which is predicted from the independent variable. The relevant output for

the analysis is shown below.The relevant figures in appendix are *Figure 4* and *Figure 5*. As shown from an excerpt of output from Standard Linear regression with target being price, the R squared value was 0.53, meaning that the model explained 53% of the variance in price and with a F statistic of 509.3 indicating some features contributed significantly to variance in price. A description of statistical significant features in the model are provided. Longitude and bedrooms had high coefficients showing that when longitude increased by 1, price would increase by $82 and when bedrooms increased by 1, price would increase by $29. Interestingly, for the dummy coded categorical variable 'former city', the districts North York and City of Toronto had large coefficients showing that with increase by one listing in North York and Old City of Toronto, price would increase by $23 and $20 respectively. However, certain features led to prediction of a decrease in price. For example, when listings were private room and shared room, price would be predicted to decrease by $35 and $51 respectively.

OLS Regression Results

Dep. Variable:	price	R-squared:	0.532
Model:	OLS	Adj. R-squared:	0.531
Method:	Least Squares	F-statistic:	509.3
Date:	Wed, 17 Nov 2021	Prob (F-statistic):	0.00
Time:	20:48:00	Log-Likelihood:	-1.0255e+05
No. Observations:	19343	AIC:	2.052e+05
Df Residuals:	19299	BIC:	2.055e+05
Df Model:	43		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.663e+04	1059.930	15.688	0.000	1.46e+04	1.87e+04
longitude	82.3151	9.308	8.844	0.000	64.071	100.559
latitude	-233.2391	15.835	-14.729	0.000	-264.277	-202.201
bedrooms	29.1728	0.931	31.343	0.000	27.348	30.997
beds	-3.4195	0.779	-4.387	0.000	-4.947	-1.892
host_total_listings_count	1.8417	0.278	6.631	0.000	1.297	2.386
host_acceptance_rate	0.3453	0.076	4.542	0.000	0.196	0.494
host_length	0.4061	0.167	2.437	0.015	0.079	0.733
instant_bookable_t	-2.0346	0.797	-2.552	0.011	-3.598	-0.472
former_city_Etobicoke	8.5782	3.570	2.403	0.016	1.581	15.576
former_city_North York	23.4454	3.097	7.572	0.000	17.376	29.515
former_city_Old City of Toronto	20.4164	2.803	7.283	0.000	14.921	25.911
former_city_Scarborough	3.1213	3.197	0.976	0.329	-3.144	9.387
former_city_York	6.1394	3.529	1.740	0.082	-0.778	13.057
room_type_Hotel room	5.4936	6.379	0.861	0.389	-7.009	17.997
room_type_Private room	-35.5520	1.495	-23.776	0.000	-38.483	-32.621
room_type_Shared room	-51.9939	2.971	-17.500	0.000	-57.817	-46.170

Interestingly, as shown in excerpt from output in the figure below, for standard linear regression with target as 'review ratings score', the F-statistic was more significant (F = 811.1) with R-squared value being 0.64 which was higher than the price model, yet different predictors contributed to the variance in the model. Refer to *Figure 4* in the Appendix for the complete

output. Statistical significant features in the model can be seen in the figure below. For example, latitude, total host listings count, and all measures relevant to review ratings score such as review scores location were predictive of review ratings score, and there was contribution of district (i.e., former city). Interestingly, when review scores of cleanliness and review scores location increased by 1, review scores rating would increase by 51 and 14 respectively. Yet with a decrease in host total listings count by 1, there was a decrease in review scores rating.

Why the large difference from the price model? As noted earlier, price has more of a normal distribution as most of the values are around the median, while review score is skewed to the right which is likely a factor.

OLS Regression Results

Dep. Variable:	review_scores_rating	R-squared:	0.644
Model:	OLS	Adj. R-squared:	0.643
Method:	Least Squares	F-statistic:	811.1
Date:	Sun, 21 Nov 2021	Prob (F-statistic):	0.00
Time:	21:39:57	Log-Likelihood:	-44489.
No. Observations:	19343	AIC:	8.907e+04
Df Residuals:	19299	BIC:	8.941e+04
Df Model:	43		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	106.4247	53.004	2.008	0.045	2.531	210.318
longitude	0.3017	0.463	0.651	0.515	-0.607	1.210
latitude	-2.1074	0.791	-2.664	0.008	-3.658	-0.557
bedrooms	-0.0135	0.047	-0.284	0.777	-0.106	0.079
beds	0.0036	0.039	0.094	0.925	-0.072	0.080
host_total_listings_count	-0.0386	0.014	-2.797	0.005	-0.066	-0.012
review_scores_accuracy	2.5273	0.067	37.797	0.000	2.396	2.658
review_scores_cleanliness	1.7785	0.035	51.237	0.000	1.710	1.847
review_scores_checkin	1.2208	0.099	12.381	0.000	1.028	1.414
review_scores_communication	2.6206	0.112	23.355	0.000	2.401	2.841
review_scores_location	1.0985	0.075	14.573	0.000	0.951	1.246
calculated_host_listings_count_entire_homes	0.0044	0.025	0.175	0.861	-0.045	0.054
calculated_host_listings_count_private_rooms	-0.0336	0.048	-0.701	0.483	-0.128	0.060
calculated_host_listings_count_shared_rooms	-2.407e-16	8.76e-17	-2.747	0.006	-4.12e-16	-6.89e-17
host_acceptance_rate	-0.0019	0.004	-0.495	0.620	-0.009	0.006
host_length	0.0086	0.008	1.041	0.298	-0.008	0.025
instant_bookable_t	0.0129	0.040	0.325	0.745	-0.065	0.091
former_city_Etobicoke	0.0568	0.177	0.320	0.749	-0.291	0.405
former_city_North York	0.2630	0.154	1.707	0.088	-0.039	0.565
former_city_Old City of Toronto	0.0384	0.140	0.275	0.783	-0.235	0.312
former_city_Scarborough	0.1517	0.159	0.955	0.340	-0.160	0.463
former_city_York	0.2510	0.175	1.431	0.152	-0.093	0.595

Metrics for our evaluation of Machine Learning Regression Models

Standard Linear regression enables a general understanding of whether there are features predictive of the target variable. To evaluate machine learning models which would enable us to determine the best machine learning model applied to our clean dataset, six metrics were collected. Mean Absolute error (MAE)-how far away predicted values are from observed values; Mean Squared Error (MSE)-The quality of a predictor based on the average square difference

between the observed and predicted values; Root Mean Squared Error (RMSE)-Value difference between the true and predicted values;R-squared (R)-Proportion of variance in the dependant variable which is predicted from the independent variable, Root Mean Squared Log Error and Mean Absolute Percentage Error. A detailed definition of Metrics is provided in *Figure 6*.

Modeling-Linear Regression

To test out a linear regression model on our cleaned dataset, we split the dataset into a training and test dataset. We decided on a larger test/train split (70/30) due to the skew of data points in order to get a more accurate result. We assigned 70% of our dataset to training and 30% to testing. We imported the LinearRegression from sklearn.linear_model and created an object of the Linear Regression class, after which we fitted our x and y to the machine learning regression model to predict price or review score.

Price as target

The excerpt of code below shown along with output indicates the R-squared, MAPE, MAE and RMSE values for Linear regression model with price being the target.As shown from the output, our R-squared value was 0.53 and the linear equation was y = 16770.51 +81.53x; MAPE being 0.39, MAE = 36 and RMSE = 48. The graph below shows prediction error with trend line for Linear Regression. *Figure 7* displays output with sample actual and predicted values.

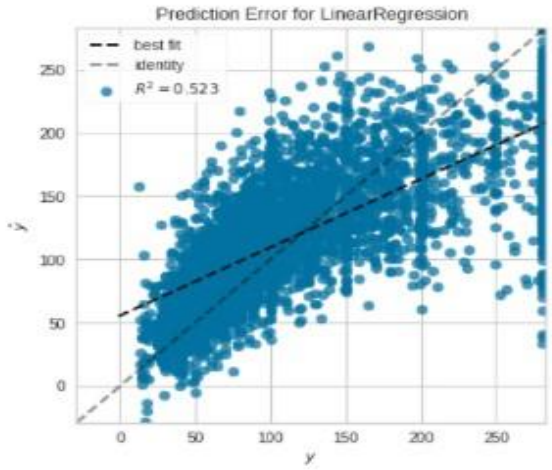
```
#In this case, 70% of data allocated to training set
xtrain, xtest, ytrain, ytest = train_test_split(X,y,train_size=0.70,random_state=42)

xtrain.shape, xtest.shape, ytrain.shape, ytest.shape

#Create the regressor
lm2 = LinearRegression()
lm2.fit(xtrain, ytrain)

predicted = lm2.predict(xtest)
```

```
R2 (explained variance): 0.53
Mean Absolute Prediction Error (Σ(|y-pred|/y)/n): 0.39
Mean Absolute Error (Σ|y-pred|/n): 36
Root Mean Squared Error (sqrt(Σ(y-pred)^2/n)): 48
 y = 16770.51990337229 + x * 81.53415644397592
```



A split of training set into 80% of the dataset yielded a similar output, as shown below.

```
R2 (explained variance): 0.52
Mean Absolute Perc Error (Σ(|y-pred|/y)/n): 0.37
Mean Absolute Error (Σ|y-pred|/n): 36
Root Mean Squared Error (sqrt(Σ(y-pred)^2/n)): 48
 y = 16770.51990337229 + x * 81.53415644397592
```

Review ratings score as target

Interestingly, the linear regression model for review ratings score was more significant in comparison.As shown from the output below, our R-squared was 0.64 and the equation was y = 183.67-3.59x; MAPE being 0.02, MAE = 2 and RMSE = 2. *Figure 8* displays output with sample actual and predicted values. When comparing the RMSE(value difference average in values) from linear models with Price and Review ratings score as target; we can note for price min is 12 and max is ~280, with RMSE being ~40. For score min is ~5.5 and max is ~10, with RMSE being 2. A spread of 40 in a range of 270 is better than a spread of 2 in a range of ~5. Thus, the RMSE for linear model on price is better than that of review ratings score.

```
R2 (explained variance): 0.64
Mean Absolute Prediction Error (Σ(|y-pred|/y)/n): 0.02
Mean Absolute Error (Σ|y-pred|/n): 2
Root Mean Squared Error (sqrt(Σ(y-pred)^2/n)): 2
 y = 183.66782964660536 + x * -3.588079172700013e-14
```

Modeling-Random Forest regression

We felt that a different regression model such as the case with Random Forest regression would yield better prediction of price or review ratings score based on training dataset. The rationale being that random forest regression is very useful for complex datasets by splitting the

data as a tree-like structure, into smaller and smaller subsets and then make predictions based on what subset a new example would fall into. An advantage of implementation of decision trees is that the sum of squared residuals is minimized by splitting the training examples as a result of learning by decision trees. The output value of a decision tree is predicted by taking the average of all of the examples that fall into a certain leaf on the decision tree and the leaf is used as an output prediction [1].

Price as target

Indeed, with the implementation of a random forest decision tree on our dataset, R-squared increased from 0.53 generated from Linear Regression model to 0.6 and MSE decreased from 41 from Linear Regression Model to 31. All indicators that the RF model was working better than Linear regression model.

```
R2 (explained variance): 0.6
RandomForest Regressor MSE is: 31.661670549715666.
RandomForest Regressor RMSE is: 5.626870404560218.
```

Review ratings score as target

Similar to what was found with linear regression, there was higher R-squared for random forest regression model with review ratings score being the target variable. MSE and RMSE were significantly reduced compared to the Random Forest regression model with review ratings score being the target variable.

```
R2 (explained variance): 0.65
RandomForest Regressor MSE is: 1.4889771186538576.
RandomForest Regressor RMSE is: 1.2202365011151968.
```

Modeling-Light Gradient Boosting Model selected as best Linear Regression model

As it would not be efficient to test out every model one at a time on our cleaned dataset to determine the best model for prediction of our target variables, we opted to use a well designed package that allowed a quick comparison of several regression models applied to our dataset. The supervised modeling package called PyCaret was implemented to select the best model for each feature. The documentation on PyCaret can be found here [2].

Price LightGBM Model

Setting up the Price Model

After validating our cleaned data, it was sent through a preprocessing pipeline included in the package. This will do imputations, code dummy variables among other things. As the data was already cleaned, dummy columns were created automatically for the categorical data in preparation for model testing. We decided on a larger test/train split (70/30) due to the skew of data points in order to get a more accurate result.

```
In [7]: #Using pycaret module to determine which model would be best.
        #Hit enter when prompted to continue set up, asking about column variable types.
        from pycaret.regression import *
        s = setup(data, target = 'price',
                  session_id = 123, train_size = 0.7)
```

	Description	Value
0	session_id	123
1	Target	price
2	Original Data	(19343, 40)
3	Missing Values	False

Determining metrics of importance and PyCaret model evaluation

Using the compare_models() function, the data went through each of the 17 models and returned a table listing metrics for comparison and an output report was generated showing the standard six metrics mentioned earlier for each as well as the time taken to generate the model. In terms of why we selected certain metrics over other metrics for evaluation of regression models, we provide our reasoning as the following. Firstly, when reporting the predicted price, we were less focused on reporting precise values and more on a range of values. There were several factors that are not included in our preprocessed and cleaned dataset such as the aesthetic value, relative location to popular attractions and venues and crime levels. As such we decided to use RMSE rather than the R-squared value to determine the viability of our tested models. This would allow us to tell potential renters a range of potential earnings they can expect for their rental unit. For review ratings scores, we were most concerned with how accurate our predictions were. As most of the review ratings are received during or after listing transactions, we wanted to find which features had the greatest impact on a rental listing's review rating. We decided to go with the R-squared score for our determinable metric as it signifies the variation of the review and we can find which features have the most impact on it.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lightgbm	Light Gradient Boosting Machine	31.4060	1929.9485	43.9074	0.6199	0.3567	0.3129	0.0690
rf	Random Forest Regressor	32.2055	2027.4467	45.0038	0.6006	0.3645	0.3223	1.2620
gbr	Gradient Boosting Regressor	33.3540	2108.1634	45.8943	0.5848	0.3776	0.3377	0.4250
et	Extra Trees Regressor	32.5954	2165.4108	46.5085	0.5735	0.3744	0.3250	1.4540
lr	Linear Regression	36.0115	2366.6580	48.6258	0.5340	0.4371	0.3769	0.2890
ridge	Ridge Regression	35.9928	2367.1384	48.6306	0.5339	0.4337	0.3761	0.0210
br	Bayesian Ridge	36.0759	2382.4473	48.7876	0.5309	0.4300	0.3757	0.1350
omp	Orthogonal Matching Pursuit	36.8443	2497.4200	49.9512	0.5083	0.4290	0.3869	0.2380
lasso	Lasso Regression	37.5344	2563.1525	50.6071	0.4954	0.4301	0.3998	0.2840
en	Elastic Net	41.0248	2878.7650	53.6444	0.4332	0.4714	0.4715	0.0910
huber	Huber Regressor	39.2292	2924.5352	54.0645	0.4242	0.4641	0.3890	0.6670
ada	AdaBoost Regressor	51.5616	3523.7673	59.3484	0.3061	0.5769	0.6938	0.3750
dt	Decision Tree Regressor	42.7730	4046.5708	63.5813	0.2023	0.4991	0.4187	0.0520
knn	K Neighbors Regressor	51.7494	4584.2339	67.6944	0.0975	0.5855	0.5938	0.0800
llar	Lasso Least Angle Regression	57.0030	5082.7976	71.2876	-0.0004	0.6439	0.7275	0.2960
dummy	Dummy Regressor	57.0030	5082.7975	71.2876	-0.0004	0.6439	0.7275	0.0220
par	Passive Aggressive Regressor	61.4767	6126.6620	76.5601	-0.1999	0.7654	0.7576	0.0400

The comparison above lists Light Gradient Boosting Machine as the model of choice for RMSE, as well as each of the other metrics followed closely by random forest. LightGBM is an open source framework based on random forest modeling but paths vertically by leaves rather than horizontally by branches. Trees are added one at a time and fit to correct predictions errors made by previous generations, thereby 'boosting' the results. The model itself has an RMSE of around 40, which is acceptable when predicting a range of values for renting.

Creating the Price Model

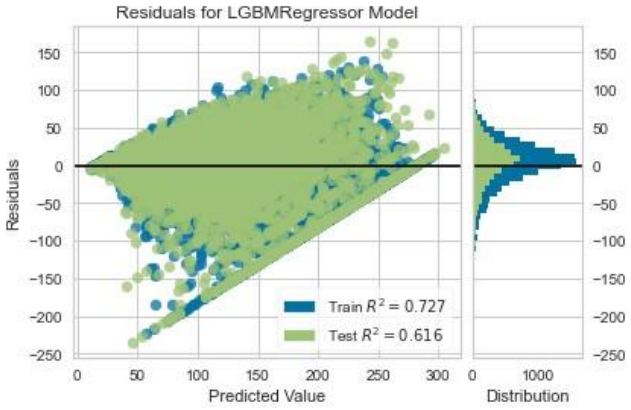
The next step was to create the price model by passing the abbreviation for our chosen model 'lightgbm' into the create_model function. We chose to do 10 folds for validation during creation. This means that the data was split into 10 random subsets and 10 iterations were completed with each subset being chosen as the test set once, with the unchosen remainder being the train set. The scores were then viewed, and if there were significant differences between iterations there is likely an issue with the data (outliers, improper scaling, etc).

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	31.8590	1985.0409	44.5538	0.6140	0.3517	0.3073
1	30.6679	1822.9177	42.6956	0.6439	0.3446	0.3026
2	29.3134	1688.4623	41.0909	0.6750	0.3398	0.2954
3	31.9212	2003.7859	44.7637	0.6146	0.3567	0.3101
4	31.3446	1950.6029	44.1656	0.6235	0.3548	0.3058
5	32.6895	2121.5910	46.0607	0.5937	0.3751	0.3287
6	32.5593	2088.8907	45.7044	0.5822	0.3631	0.3232
7	31.7084	1948.3915	44.1406	0.6022	0.3690	0.3219
8	31.0981	1889.2625	43.4656	0.6071	0.3625	0.3229
9	30.8985	1800.5397	42.4328	0.6429	0.3502	0.3108
Mean	31.4060	1929.9485	43.9074	0.6199	0.3567	0.3129
SD	0.9391	126.2602	1.4460	0.0261	0.0103	0.0102

As seen from the above output, most of the scored values did not vary by a large degree. As a result, we accepted the LGBM as an accurate model for predicting rental price per day.

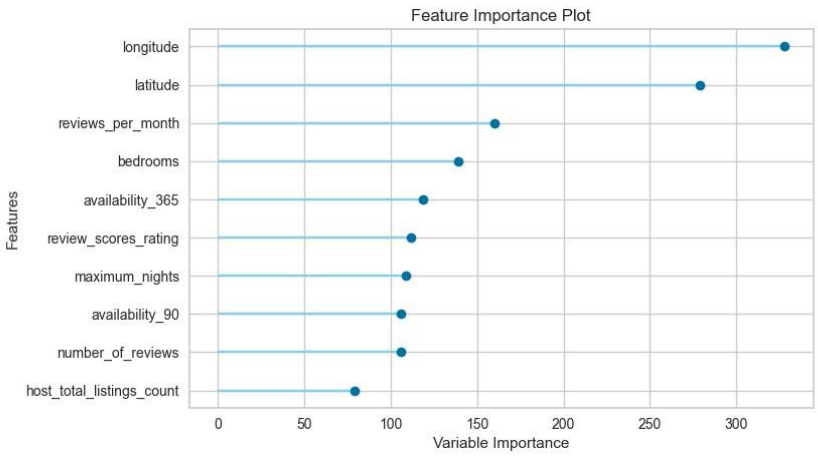
Plotting the Price Model

The plot_model function generated a plot which shows a comparison for the residuals and relative distribution for the test and training sets. As can be seen from the figure below, for this comparison the overlap on residuals was fairly consistent and the value distributions were similar (i.e., lining up) showing that there is no significant overfitting issue in the training set. The test set had a lower R-squared but not to an extreme. This was likely due to differences between the data contained in the test and train sets.



Evaluating the Price Model

We implemented the evaluate_model function to generate explanatory plots to assist in explaining the model. From the generated figure below, we could see which features are most important in predicting price.The feature importance plot indicated that the physical location (longitude and latitude) are the biggest contributing factor to determining price. This is most likely due to how close they are to major attractions. Amount of reviews and review rating are also important, indicating that the length and score of reviews for a rental directly impacts how much can be charged. And finally the total amount of listings per client and the maximum nights stay are likely due to corporate entities such as hotels.



Finalizing and Saving the Model

The model was exported for deployment, finalize_model fit the estimator onto the dataset. The output below shows the arguments for the model and allows the user to tune the model (more branches, give weights, max depth, etc.) . The model was then exported (*fig*) for deployment, finalize_model would fit the estimator onto the dataset.

```
#Finalizing the model for deployment
finalize_model(price_model)
```

```
LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
              importance_type='split', learning_rate=0.1, max_depth=-1,
              min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
              n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
              random_state=123, reg_alpha=0.0, reg_lambda=0.0, silent=True,
              subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

Review Score GBR Model

Setting up the Review Score Model

Similar to the setup of the price model, we are instead chose review ratings score as the target variable while keeping the test train split at 70/30.

```
In [17]: #Continuing, we will do the same for review scores.
         s = setup(data, target = 'review_scores_rating',
                   session_id = 123, train_size = 0.7)
```

	Description	Value
0	session_id	123
1	Target	review_scores_rating
2	Original Data	(19343, 40)
3	Missing Values	False

For review score, we have already decided to select R-squared as our metric of determination. We implemented a similar pipeline as before, replacing ‘R-squared’ as the sort. As seen from the figure below we saw that Gradient Boosting Regressor (GBR) had the highest R-squared as well as most of the other metrics. LightGBM was extremely close and if processing time was a factor would be chosen. However as we were focused on maximizing accuracy, we selected GBR. GBR is another decision tree model with a stronger emphasis on reducing bias over simplification. While it isn’t a factor in this case, it also helps improve models by reducing the effect that overfitting would have.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	1.4718	5.5544	2.3551	0.6627	0.0248	0.0156	0.4440
lightgbm	Light Gradient Boosting Machine	1.4543	5.6000	2.3648	0.6599	0.0249	0.0154	0.0530
br	Bayesian Ridge	1.6357	5.8661	2.4207	0.6437	0.0255	0.0174	0.1230
ridge	Ridge Regression	1.6354	5.8700	2.4215	0.6435	0.0256	0.0174	0.0170
lr	Linear Regression	1.6356	5.8706	2.4216	0.6434	0.0256	0.0174	0.0500
omp	Orthogonal Matching Pursuit	1.6158	5.8721	2.4220	0.6433	0.0256	0.0172	0.2360
rf	Random Forest Regressor	1.4686	5.8923	2.4258	0.6420	0.0256	0.0156	1.2090
et	Extra Trees Regressor	1.5153	6.6061	2.5680	0.5986	0.0271	0.0161	1.3480
ada	AdaBoost Regressor	2.0287	7.7024	2.7744	0.5319	0.0291	0.0214	0.2250
en	Elastic Net	2.5643	10.4956	3.2389	0.3625	0.0342	0.0272	0.0240
lasso	Lasso Regression	2.6037	10.7556	3.2789	0.3467	0.0346	0.0277	0.0220
dt	Decision Tree Regressor	1.8296	11.3538	3.3676	0.3100	0.0355	0.0194	0.0490
huber	Huber Regressor	2.6800	11.7639	3.4292	0.2853	0.0362	0.0285	0.7370
llar	Lasso Least Angle Regression	3.2357	16.4627	4.0569	-0.0002	0.0428	0.0345	0.2700
dummy	Dummy Regressor	3.2357	16.4627	4.0569	-0.0002	0.0428	0.0345	0.0180
knn	K Neighbors Regressor	3.2473	17.5998	4.1945	-0.0693	0.0442	0.0346	0.0730
par	Passive Aggressive Regressor	3.5355	20.9040	4.3938	-0.2633	0.0459	0.0372	0.0580

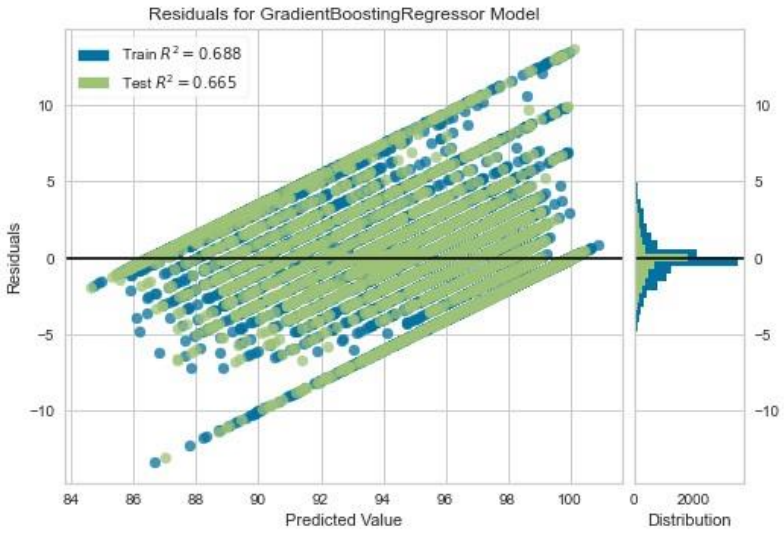
Creating the Review Score Model

As before, we created the model using the gbr argument with 10 folds for validation. The output is displayed in the figure below.

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	1.4320	5.0629	2.2501	0.6914	0.0237	0.0152
1	1.4260	5.1476	2.2688	0.6817	0.0239	0.0151
2	1.4616	5.6728	2.3818	0.6729	0.0251	0.0155
3	1.5416	6.0749	2.4647	0.6238	0.0260	0.0164
4	1.5368	6.1902	2.4880	0.6323	0.0263	0.0164
5	1.4053	5.0455	2.2462	0.6838	0.0236	0.0149
6	1.4861	5.6047	2.3674	0.6567	0.0250	0.0158
7	1.5466	5.8707	2.4229	0.6534	0.0256	0.0165
8	1.4489	5.7951	2.4073	0.6569	0.0254	0.0154
9	1.4334	5.0800	2.2539	0.6739	0.0238	0.0152
Mean	1.4718	5.5544	2.3551	0.6627	0.0248	0.0156
SD	0.0501	0.4172	0.0886	0.0211	0.0010	0.0006

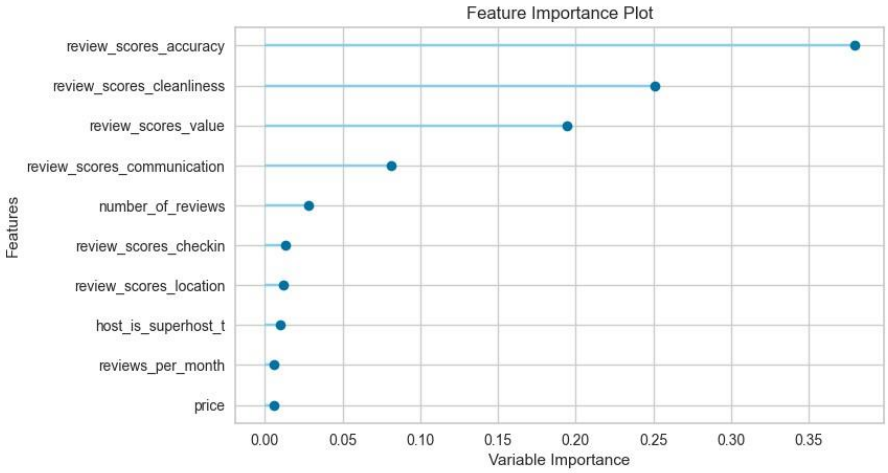
Plotting the Review Score Model

The plot for score showed that the test and train sets are quite close in their r2 and value distribution. The model seemed acceptable for use.



Evaluating the Review Score Model

Reviewing the important features it seemed that the accuracy of the rental listing was the most important feature for determining score in this model. Other important features were perceived value, cleanliness and communication. It may be prudent to recommend to new renters that these factors play an important role in maximizing their review scores.



Finalizing and Saving the Model

The model was exported (*fig*) for deployment, finalize_model fit the estimator onto the dataset.

```
#Finalizing the model for deployment
finalize_model(score_model)
```

```
GradientBoostingRegressor(alpha=0.9, ccp_alpha=0.0, criterion='friedman_mse',
                          init=None, learning_rate=0.1, loss='ls', max_depth=3,
                          max_features=None, max_leaf_nodes=None,
                          min_impurity_decrease=0.0, min_impurity_split=None,
                          min_samples_leaf=1, min_samples_split=2,
                          min_weight_fraction_leaf=0.0, n_estimators=100,
                          n_iter_no_change=None, presort='deprecated',
                          random_state=123, subsample=1.0, tol=0.0001,
                          validation_fraction=0.1, verbose=0, warm_start=False)
```

Conclusion

After trying standard modeling techniques (Linear Regression and Random Forest), the team turned to supervised ML to find which model would perform best with our data. Based on evaluation of metrics, we selected models based on LightGBM and GBR to be used to support our business objectives. Both models performed reasonably well and should be suitable for use. In the next section we will use Tableau for further analysis and generating graphical representations of our findings.

References

[1] https://mlcorner.com/linear-regression-vs-decision-trees/

[2] *PyCaret Regression Library Documentation*. PyCaret. (2020, July 31). Retrieved November 9, 2021, from https://pycaret.org/regression1/

[3]https://machinelearningmastery.com/light-gradient-boosted-machine-lightgbm-ensemble/

Appendix

Code:

https://colab.research.google.com/drive/1_d0RTZmyVJg9DRQu7db5VRmNsBYUyPSG#scrollTo=qCTdbZrTP4og

Figures

Table 1. Fields or features in the AirBnb Data Set with description and type of feature stated

Original Dataset features; In total there were 73 features in the original dataset that were not cleaned.

Number of Variables	Column Name	Description	Feature type(e.g., Numeric, String)
11	host_since	Date an individual became a host	Numeric
14	*host_response_time*	Time to respond to customer booking inquiry; ranges from 1hour to few days	Numeric
15	*host_response_rate*	Ranges from 0% to 100% for reply to booking inquiries	Numeric
16	*host_acceptance_rate*	Ranges from 0% to 100% response for acceptance of booking	Numeric
17	host_is_superhost	Is either t(true) or f(false)	Boolean
20	host_neighbourhood	Description of neighborhood listing	String
21	host_listings_count	Current number of host listings	Numeric
22	host_total_listings_count	Total number of listings made by host	Numeric
29	latitude	The angular distance of a location or object north or south of the Earth's celestial equator	String
30	longitude	The angular distance of a location or object east or west of the meridian	String
31	property_type	Type of property (e.g., Entire house)	String
32	room_type	Specific type of room (e.g., Entire home/apt)	String
33	accommodates	How many people can stay (e.g., 6)	Numeric
36	bedrooms	Number of bedrooms in property	Numeric
37	beds	Number of beds in property	Numeric

39	*price*	Price of stay per night	Numeric
40	minimum_nights	Minimum nights can be booked by same individual	Numeric
41	maximum_nights	Maximum nights can be booked by same individual	Numeric
50	availability_30	Availability of Property in 30 days	Numeric
51	availability_60	Availability of Property in 60 days	Numeric
52	availability_90	Availability of Property in 90 days	Numeric
53	availability_365	Availability of Property in 365 days	Numeric
55	number_of_reviews	Total number of reviews that a listing has received from customers	Numeric
56	number_of_reviews_ltm	The number of reviews that a listing has received last twelve month	Numeric
57	number_of_reviews_l30d	The number of reviews that a listing has received per 130 days	Numeric
60	***review_scores_rating***	***Customer-provided score rating (0% to 100%); A customer-provided review score attributed to a listing based on overall experience and satisfaction***	Numeric
61	review_scores_accuracy	Accuracy of review scores (0 to 10)	Numeric
62	review_scores_cleanliness	Cleanliness score (0 to 10)	Numeric
63	review_scores_checkin	Over-all check in score (0 to 10)	Numeric
64	review_scores_communication	Score on communication with host (0 to 10)	Numeric
65	***review_scores_location***	***Score on location based on factors such as nearby transportation, noise level (0 to 10)***	Numeric
66	review_scores_value	Over- all value/quality/experience (0 to 10)	Numeric
68	instant_bookable	True or False if customer can instantly book	BOolean
69	calculated_host_listings_count	Calculated number of listings by host	Numeric
70	calculated_host_listings_count_entire_homes	Number of host listings which are entire homes	Numeric
71	calculated_host_listings_count_private_rooms	Number of host listings which are private rooms	Numeric

72	calculated_host_listings_count_shared_rooms	Number of host listings which are shared homes	Numeric
73	reviews_per_month	Number of Customer reviews of the host accommodation or accommodations per month	Numeric

EDA

Figure 1.

```
#Check data is intact
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19343 entries, 0 to 19342
Data columns (total 40 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   longitude                                     19343 non-null  float64
 1   latitude                                      19343 non-null  float64
 2   bedrooms                                      19343 non-null  float64
 3   beds                                          19343 non-null  float64
 4   host_total_listings_count                     19343 non-null  float64
 5   review_scores_rating                          19343 non-null  float64
 6   review_scores_accuracy                        19343 non-null  float64
 7   review_scores_cleanliness                     19343 non-null  float64
 8   review_scores_checkin                         19343 non-null  float64
 9   review_scores_communication                   19343 non-null  float64
 10  review_scores_location                        19343 non-null  float64
 11  review_scores_value                           19343 non-null  float64
 12  reviews_per_month                             19343 non-null  float64
 13  accommodates                                  19343 non-null  int64
 14  price                                         19343 non-null  float64
 15  minimum_nights                                19343 non-null  int64
 16  maximum_nights                                19343 non-null  int64
 17  availability_30                               19343 non-null  int64
 18  availability_60                               19343 non-null  int64
 19  availability_90                               19343 non-null  int64
 20  availability_365                              19343 non-null  int64
 21  number_of_reviews                             19343 non-null  int64
 22  number_of_reviews_ltm                         19343 non-null  int64
 23  number_of_reviews_l30d                        19343 non-null  int64
 24  instant_bookable                              19343 non-null  object
 25  calculated_host_listings_count                19343 non-null  int64
 26  calculated_host_listings_count_entire_homes   19343 non-null  int64
 27  calculated_host_listings_count_private_rooms  19343 non-null  int64
 28  calculated_host_listings_count_shared_rooms   19343 non-null  int64
 29  former_city                                   19343 non-null  object
 30  room_type                                     19343 non-null  object
 31  host_since                                    19343 non-null  object
 32  host_response_time                            19343 non-null  object
 33  host_is_superhost                             19343 non-null  object
 34  host_acceptance_rate                          19343 non-null  int64
 35  host_response_rate                            19343 non-null  int64
 36  labels_host_acceptance_rate                   19343 non-null  object
 37  labels_host_response_rate                     19343 non-null  object
 38  host_since_year                               19343 non-null  int64
 39  host_length                                   19343 non-null  int64
dtypes: float64(14), int64(18), object(8)
memory usage: 5.9+ MB
```

Figure 2.

```
#Check correlation of target variables - 'price'
corr_p = df.corr()['price'].abs().sort_values(ascending = False)
corr_p
#Price correlates highest with how many people can stay and how many beds and bedrooms.
#There is also a lesser correlation between whether the room is private (or a full home).
```

```
price                                           1.000000
accommodates                                    0.595983
bedrooms                                        0.540802
beds                                            0.493700
calculated_host_listings_count_private_rooms    0.412672
calculated_host_listings_count_entire_homes     0.339368
latitude                                        0.275433
review_scores_location                          0.147562
availability_30                                 0.092657
host_since_year                                 0.078497
host_length                                     0.078497
review_scores_rating                            0.076440
review_scores_cleanliness                       0.072589
availability_60                                 0.068199
maximum_nights                                  0.059920
availability_90                                 0.055047
calculated_host_listings_count                  0.035810
availability_365                                0.033946
review_scores_checkin                           0.033755
reviews_per_month                               0.032303
review_scores_accuracy                          0.028995
host_acceptance_rate                            0.024670
minimum_nights                                  0.020220
number_of_reviews                               0.013625
review_scores_communication                     0.009418
review_scores_value                             0.008806
longitude                                       0.005433
number_of_reviews_ltm                           0.001903
host_total_listings_count                       0.000012
number_of_reviews_l30d                               NaN
calculated_host_listings_count_shared_rooms          NaN
host_response_rate                                   NaN
Name: price, dtype: float64
```

Figure 3.

```
#Continued for 'review_score_rating'
corr_r = df.corr()['review_scores_rating'].abs().sort_values(ascending = False)
corr_r
#Cleanliness, accuracy of listing and percieved value appear to be the most important factors in determining a review score.
#Communication, checkin perceptions and location (outside circumstantial) are also important, although less so.
```

```
review_scores_rating                            1.000000
review_scores_cleanliness                       0.650398
review_scores_accuracy                          0.645803
review_scores_value                             0.643939
review_scores_communication                     0.559467
review_scores_checkin                           0.480229
review_scores_location                          0.380516
host_total_listings_count                       0.179073
calculated_host_listings_count                  0.170597
calculated_host_listings_count_private_rooms    0.103404
price                                           0.076440
calculated_host_listings_count_entire_homes     0.072359
availability_30                                 0.072297
availability_60                                 0.068570
availability_90                                 0.060145
number_of_reviews_ltm                           0.051054
number_of_reviews                               0.050883
host_length                                     0.049167
host_since_year                                 0.049167
latitude                                        0.047717
availability_365                                0.043711
minimum_nights                                  0.043324
maximum_nights                                  0.038906
host_acceptance_rate                            0.019183
bedrooms                                        0.010735
reviews_per_month                               0.010130
longitude                                       0.004461
accommodates                                    0.003989
beds                                            0.000978
number_of_reviews_l30d                               NaN
calculated_host_listings_count_shared_rooms          NaN
host_response_rate                                   NaN
Name: review_scores_rating, dtype: float64
```

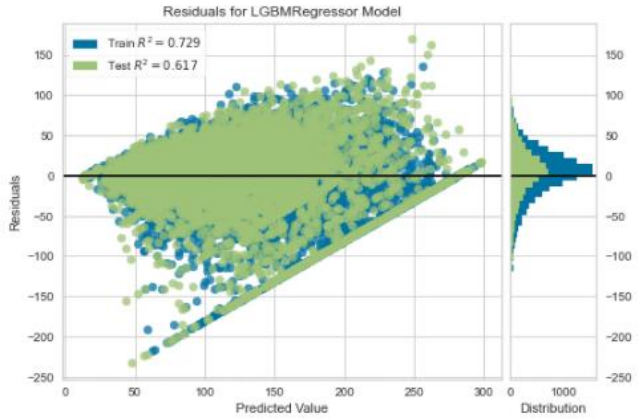
MACHINE LEARNING

Name of Metric	Definition
Mean Absolute Error	How far away predicted values are from observed values
Mean Squared Error	The quality of a predictor based on the average square difference between the observed and predicted values
Root Mean Squared Error	Value difference between the true and predicted values
R2	Proportion of variance in the dependant variable which is predicted from the independent variable

Root Mean Squared Log Error	Ratio difference between the observed and predicted values
Mean Absolute Percentage Error	Prediction accuracy shown as a percentage value

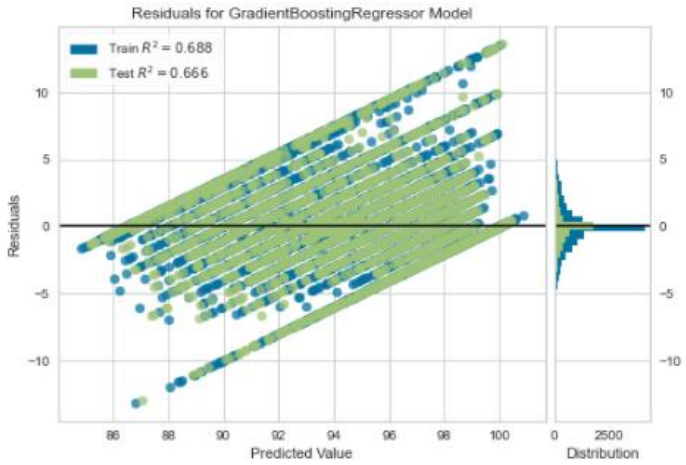
70% training set				
Price_Linear Regression				
Review Ratings Score_Linear Regression				
Price_Random Forest Regression				
Review Ratings Score_Random Forest Regression				
Price_LGBM				
Review Ratings score_LGBM				
Price_GBM				
Review Ratings score_GBM				

In [13]:
```
#Plot of the price model
#Distribution is OK, test r2 is a lower than train set.
#We are looking at RMSE however, so this isn't a complete loss.
plot_model(price_model)
```



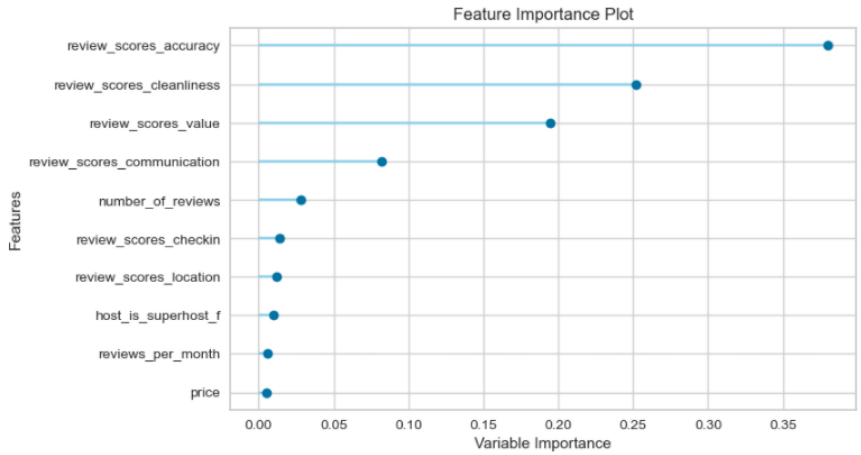
In [9]:
```
#Evaluating the model
#The location appears to be the most important features, likely due to what attractions they are
#close to, and the pricing of the neighbourhood. The number of reviews and their rating directly impacts
#the price, meaning getting as many good reviews is essential to maximizing profit. Beyond that
#availability and the maximum number of nights allowed could be due to 'long term' renters, more research
#may be needed. Finally total listings count is likely due to those with high listings being hotels or other
#corporate entities, which are able to charge a premium for established consistent service.
evaluate_model(price_model)
```

In [20]:
```
#Plotting the score model
#Test set isn't too far off from the train set
#Residuals aren't too excessive and the distribution of the test/train are similar.
plot_model(score_model)
```



In [21]:
```
#Evaluating the model
#The accuracy of a listing and its location have the highest importance in our model.
#This would indicate that customers value the area being to their liking (safety, accessibility, crime levels)
#as well as the listing meeting their expectations after their stay. This is further corraborated by the
#checkin score also being moderately important. It should also be noted that the number of reviews could be
#another factor, more analysis may be required.
evaluate_model(score_model)
```

Plot Type: Hyperparameters | Residuals | Prediction Error | Cooks Distance | Feature Selection
Learning Curve | Manifold Learning | Validation Curve | Feature Importance | Feature Importance...
Decision Tree | Interactive Residuals



In [15]:
```
#Finalizing the model for deployment
finalize_model(price_model)
```

Out[15]:
```
LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
              importance_type='split', learning_rate=0.1, max_depth=-1,
              min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
              n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
              random_state=123, reg_alpha=0.0, reg_lambda=0.0, silent=True,
              subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

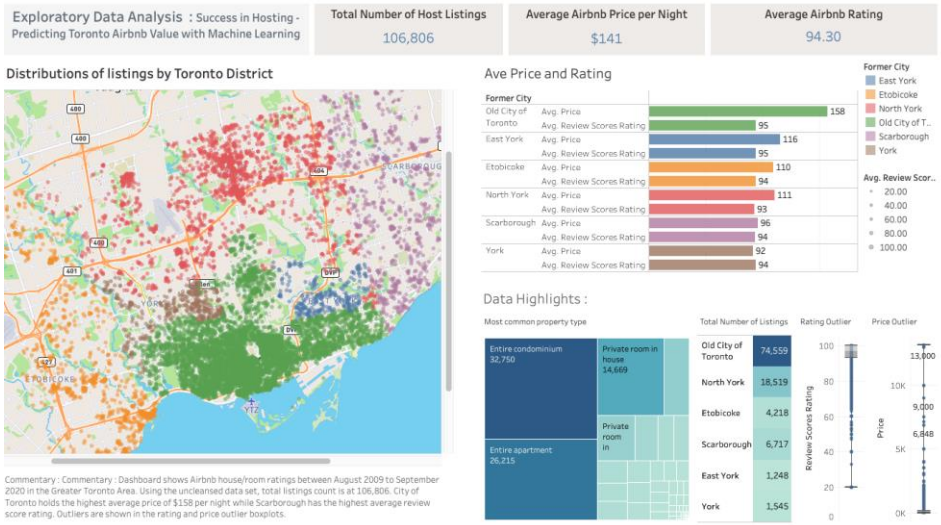
In [16]:
```
#Saving the model for future use
save_model(price_model, 'P3_Price_Model')
```

```
Transformation Pipeline and Model Successfully Saved
```

Out[16]:
```
(Pipeline(memory=None,
          steps=[('dtypes',
                  DataTypes_Auto_infer(categorical_features=[],
                                       display_types=True, features_todrop=[],
                                       id_columns=[], ml_usecase='regression',
                                       numerical_features=[], target='price',
                                       time_features=[])),
                 ('imputer',
                  Simple_Imputer(categorical_strategy='not_available',
                                 fill_value_categorical=None,
                                 fill_value_numerical=None,
                                 numeric_strategy='...
                  LGBMRegressor(boosting_type='gbdt', class_weight=None,
                                colsample_bytree=1.0, importance_type='split',
                                learning_rate=0.1, max_depth=-1,
                                min_child_samples=20, min_child_weight=0.001,
                                min_split_gain=0.0, n_estimators=100, n_jobs=-1,
                                num_leaves=31, objective=None, random_state=123,
                                reg_alpha=0.0, reg_lambda=0.0, silent=True,
                                subsample=1.0, subsample_for_bin=200000,
                                subsample_freq=0)]],
          verbose=False),
 'P3_Price_Model.pkl')
```

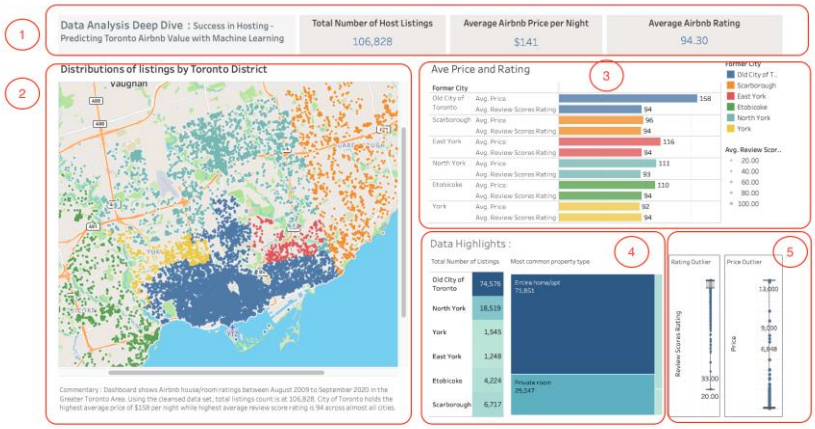
Deliverable 4

Dashboard 1 - Exploratory Data Analysis using uncleansed dataset :



Dashboard shows Airbnb house/room ratings between August 2009 to September 2020 in the Greater Toronto Area. Using the uncleansed data set, total listings count is at 106,806. City of Toronto holds the highest average price of $158 per night while Scarborough has the highest average review score rating. Outliers are shown in the rating and price outlier boxplots.

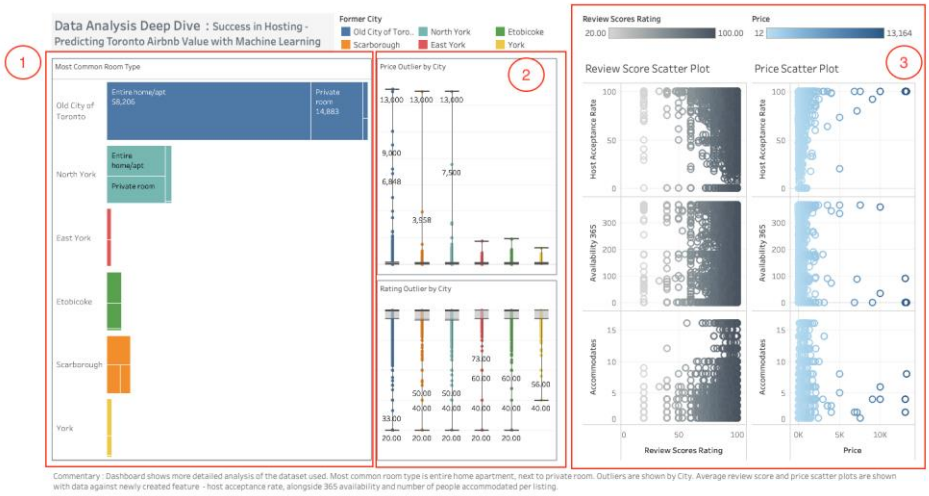
Dashboard 2 - Exploratory Data Analysis using cleansed dataset :



Dashboard shows Airbnb house/room ratings between August 2009 to September 2020 in the Greater Toronto Area. Using the cleansed data set, total listings count is down to 5,051. City of Toronto holds the highest average price of $151 per night while Scarborough and East York have the highest average review score rating.

Parts of the Dashboard:

1. Shows the Total number of Listings and Average Airbnb Price and Rating
2. Shows where most number of Listings are located - circle marks show that most of ratings are on the 80 to 100 range
3. Bar chart shows most and least expensive cities for airbnb rent
4. Tree map shows the most common property types
5. Outliers



Commentary : Dashboard shows more detailed analysis of the dataset used. Most common room type is entire home apartment, next to private room. Outliers are shown by City. Average review score and price scatter plots are shown with data against newly created feature - host acceptance rate, alongside 365 availability and number of people accommodated per listing.

Link to Final Presentation:

Airbnb Toronto

Criteria	Points	Comments
Introduction: Executive summary, business objective, problem description. ☐ Missing exec summary ☐ Not well motivated ☐ Vague/Confusing ☐ Poor description	/5	
Data Description and Cleaning/Wrangling: description of variables, supported with figures and graphs. Any cleaning, preprocessing, and wrangling performed. ☐ Need more descriptive stats and figures ☐ Missing cleaning ☐ Doesn't describe all data ☐ More cleaning should be performed	/5	
Analysis methodology description: Description of approach, models, algorithms, and/or tools. ☐ Missing ☐ Confusing ☐ Vague ☐ Incorrect choice ☐ Too complicated ☐ Too technical	/10	
Decision, Recommendation or Conclusion: Discussion and interpretation of the results. What should the business do, as a result of the analysis? ☐ Missing ☐ Not supported by data ☐ Vague ☐ Unrealistic ☐ Confusing ☐ Trivial	/20	
Organization and Clarity of Presentation: presentations skills, slide organization, oration, flow, storytelling ☐ Too quiet ☐ Text-heavy slide(s) ☐ Mumbled ☐ Unreadable fonts ☐ Shy ☐ Axes not explained ☐ Confusing flow ☐ Bad colors/template	/10	
Q&A: Answered questions from audience with authority, clarity, confidence, and honesty	/5	
Automatic Deductions ☐ More than 20 seconds spent on title slide ☐ Long agenda/outline slide ☐ Too many decimal points ☐ Thank you/Questions slide instead of Summary slide ☐ Went over in time		
Bonus Points ☐ Well-placed meme	/5	
TOTAL	/60	

Commented [1]: First slide is an executive summary.

Problem statement

Final Report

FINAL REPORT

Congratulations! This report is your final assignment for the last course in the data science certificate!

My goal for you is to have a tangible item that you can use to showcase in your interviews. This report is out of 80 marks and is worth 20% of your final grade.

A few key things:

- ✓ Deliver an MS Word (or PDF) document as your final deliverable
- ✓ R/Python code is mandatory
- ✓ Page limit: 15 pages maximum, double spaced, size 11 font
- ✓ Appendix: No page limit. Try to make it useful and not a blob of graphs

Component	Excellent	Required additional effort	Grade
Problem statement and framing according to the proposed data analytics solution *A summary of what it is you are trying to solve and why, stakeholder review and ROI analysis (if applicable)*	Problem statement effectively summarizes the opportunity you are tackling, the desired result and KPIs. Discussion of the 5W's of the problem. Financial measure of success discussed where appropriate.	Many major elements missing, or section is missing completely	/15
Research Methodology & Ethics *A descriptive review on the steps taken to create the problem statement, cited literature reviews conducted, and a discussion on ethical techniques used while handling the*	Thorough research performed to understand the industry (or data set) that you are working with. Well cited references and literature reviews. A respectful discussion of ethical treatment of data,	Major elements missing including no literature review or the section is missing completely.	/5

data and developing your model	highlighting caveats or challenges you encountered.		
Model creation *An overview of the techniques, the variance explained, influential features and what this means to the business or stakeholder*	Describe your modelling approach, the algorithm you chose and any tools you used. A description on the metric you used to evaluate your model and tradeoffs Discuss the interpretation of your results and how this can be applied to the business or stakeholder	Major elements missing, incorrect technique used and no application to the business	/30
Visualization with Dashboard *A link to your published Tableau dashboard and/or a screen shot of the dashboard, with a description on how the business can use this tool in an interactive way*	Dashboard looks professionally made with effort to add in formatting (such as $ vs % where appropriate) Good use of Tableau's functionality such as maps and embedded formulas (where appropriate) Link included with the Tableau workbook published and error free	Major UI misses, Tableau section missing entirely	/10

Recommendations & Conclusion *Wrap up your report with recommendations*	Provide clear recommendations on what the next steps would be, with business recommendations and a solid conclusion.	Section missing completely.	/20
Organization, clarity, and cohesiveness	A well thought out report with a contents page, all graphs labelled, axes labelled with business sense, page numbers and general effort put in. Your final result should be professional and something you should be proud to showcase at interviews and on your LinkedIn page!		
Deductions	All past reports just pasted under each other A document not submitted, rather just code No/Shallow reference to business insights No real recommendations to the business Over the page limit		

https://docs.google.com/document/d/13UyWqoWfUXPimtIFh8sMW5dAoZtwnGUDA6z9em9rDPo/edit#