IS424: Data Mining and Business Analytics

Final Project Report

Project Title:

Airbnb Price Prediction Analysis In Singapore

Team Name:	**G2T5**
Team Members:	Sanofer
	Heng Si Kai
	Leow Jia Xiang
	Kowsalya Ganesan
	Sithi-Rukaiyaa Arssath
	Tanisha Kiran Bhagwanani
	Luqman Juzaili Bin Muhammad Najib

1. Introduction

Our project is focused on Airbnb Price Prediction Analysis in Singapore. This project aims to predict the prices of Airbnb listings in Singapore using various machine learning techniques. The project uses a dataset of Airbnb listings in Singapore that includes information on the listing's location, number of rooms, amenities, and other features. The project uses a number of regression models to predict the price of a listing based on the features of the listing, and eventually attempts to conclude with which model is the best in its price prediction capabilities using multiple measure criteria.

1.1. Problem Background

Airbnb has become a popular choice for travellers looking for affordable and unique accommodation options. As a result, Airbnb listings have become a significant part of the hospitality industry. In Singapore, the demand for Airbnb listings has increased in recent years due to the country's growing tourism industry. However, one of the biggest challenges for both hosts and guests is determining the appropriate price for a listing. The question still stands: How should Airbnb hosts price their properties to maximise profits? Should they be pricing their properties at high markup prices to attempt to derive higher profit margins at the expense of potential customers, or should they be more aggressive and price competitively to attract more potential tenants? Many Airbnb hosts struggle with this pricing problem: they don't know if they're undercutting potential profits through suboptimal pricing or deterring customers with expensive quotes.

The goal of this project is to propose optimal prices using an AI model while clearly explaining the derivation of such prices for hosts to be more objective in their various pricing decisions. This will be achieved by developing a machine-learning model that can accurately predict the price of an Airbnb listing in Singapore based on various features.

1.2. Problem Statement

The challenge in predicting the price of an Airbnb listing in Singapore is due to the many variables at play that can affect its price. These variables include the location of the listing, the number of rooms, the type of property, the amenities provided and the list goes on. The aim of this project is to develop a model that can accurately predict the price of a listing based on these variables.

2. Motivation

The motivation for this project is to help Airbnb hosts in Singapore to set appropriate prices for their listings. By accurately predicting the price of a listing, hosts can optimize their earnings while providing fair pricing to guests, helping them with their pricing decisions. In addition, guests can benefit from accurate price predictions by being able to find listings that fit their budget. This would help to reduce occurrences of underpriced or overpriced listings as well.

3. Literature Review

There have been several studies on Airbnb price prediction using machine learning techniques. Most of these studies have focused on the US and European markets, with limited research on the Singapore market. Some studies have used linear regression models to predict Airbnb prices, while others have used tree-based models and neural networks. One recent study used a deep learning model to predict Airbnb prices in Singapore. However, there is still a need for more research on this topic, especially in the Singapore market.

3.1 Location Data Analysis

The first literature review is sourced from Towards Data Science and focuses on the prediction of Airbnb prices using machine learning and location data (Carrillo, 2020).

This project uses varying listing features to try and predict price, with an added feature of a predictor based on space: the property's proximity to specific venues. With this, the model can place an implicit price on aspects of a listing such as living closer to a supermarket, bar, or pub.

This project used the data scraped from Airbnb in July 2019 that contains information on all Airbnb listings in Edinburgh, Scotland at that time.

Some important features this project looked into were the number of guests the listing could accommodate, the number of bedrooms and bathrooms in a listing, the number of nights stayed at a listing, etc.
The approach this review focused on was log-transforming the data in order to make "price" more normally distributed, hot-encoding categorical features, and then standardizing them using StandardScaler().

The models used were the Spatial Hedonic Price Model (OLS Regression), with the LinearRegression from Scikit-Learn, and the Gradient Boosting method, with the XGBRegressor from XGBoost.

In the end, the review concluded that XGBoost performs better than the Spatial Hedonic Price Model, however, it only predicts 66% of the variation in price. Furthermore, this model was built without the reviews column.

As a result, the team decided to include reviews in the dataset to understand better how positive and negative reviews can potentially impact price listings overall.

3.2 Dynamic Price Strategy

The second literature review is sourced from ResearchGate and focuses on the use of dynamic pricing strategies by Airbnb hosts (Gibbs et al., 2017).

This project goes over dynamic pricing as a tool to maximize profit by adjusting the price of a product or service continuously to meet the fluctuation in demand. This study does so by using attribute and sales information from 39,837 Airbnb listings and hotel data from 1,025 hotels across five markets to test their varying hypotheses.

Airbnb has a unique position in the sharing economy, with hosts being able to set daily, weekly, and monthly room rates and control prices as time goes by.

While dynamic pricing is considered to be a strategic revenue management tool to maximize profit through the adjustment of price, the project mentions how Airbnb hosts seem to lack the motivation and skills that are associated with dynamic pricing.

Thus, the review notes that Airbnb hosts make limited use of dynamic pricing overall. As a result, there is a push for greater host motivation in order to better understand the determinants of a listing when pricing in order to better adhere to the fluctuations in price.

3.3 Machine Learning and Sentiment Analysis

The final literature review is sourced from Cornell Univeristy and focuses on Airbnb price prediction using machine learning and sentiment analysis (Kalehbasti et al., 2021).

This project focuses on the challenges and issues in regard to the pricing of the property and the evaluation of the price of a property that customers/owners face when it comes to gauging the value of a property and ensuring optimal pricing.

The paper dived into a couple of approaches such as sentiment analysis - in order to further investigate customer reviews, and feature importance analysis -in order to manually select features in order to reduce the model variance overall. The paper then utilizes models such as linear regression, tree-based models, SVR, KMC, and NNS.

For sentiment analysis, the review uses TextBlob sentiment analysis library to analyze the reviews of the Airbnb listings to assign sentiment scores to each review from -1 to 1, where -1 being the most negative to the 1 being the most positive sentiment. This is to further understand the possible correlation between customer reviews in regards with the price.

For feature importance analysis, the paper looked into methods to reduce the high dimensionality data of 764 columns. Several methods such as the manual, p-value and lasso cross-validation were used to obtain the best dimensionality method. Ultimately, the best method was through lasso cross-validation in which they were able to reduce 764 features to 78 features with non-zero values.

The models that were used were then linear regression models, with the standard linear regression and ridge regression which is Linear Regression with L2 regularisation to introduce a penalising term to squared error cost function to prevent overfitting. Tree-based models were also used to handle non-linear relationships between the variables and the price. K-means clustering was also leveraged to cluster features with Ridge Regression on each cluster, with the focus of handling non-linearity of data as well. Support Vector Regression (SVR) was then used to model the non-linear relationship between covariates.

The review concluded that excessive amounts of features will end up resulting in high variance and a weak performance of the model. However, using the SVR method produced the best results, with an R Square that was equal to 69%, and a MSE of 0.147.

4. Dataset

4.1 Types of Dataset

There were a total of 6 datasets used in the project which would be the Airbnb listings, the reviews for the Airbnb listings, the translated reviews for the Airbnb listings, Singapore MRT coordinates, Singapore Tourist Attractions coordinates and Singapore Mall coordinates.

Dataset 1: Airbnb Singapore Listing Dataset
Dataset 1 is sourced from Kaggle's website (ks4s, 2019) and contains information on Airbnb listings in Singapore, including the listing's location, number of rooms, amenities, and other features.

The dataset contains 8293 rows and 98 columns. The team further explored the columns that were available and could identify many columns that may hold insignificant information towards the prediction of an Airbnb listing price. Hence, there is a need to reduce the number of columns which will be explored more in section 4.3.

Dataset 2: Airbnb Singapore Listing Reviews Dataset
Similar to Dataset 1, Dataset 2 is also obtained from Kaggle (ks4s, 2019) and contains the review information regarding the Airbnb listings in Dataset 1. Each review will be tagged to a listing ID that can be traced back to Dataset 1. This dataset contains 10,919 rows and 6 columns. This means that for every listing, there could be zero, one or more than one review. The team will be using this dataset to obtain the sentiment scores of each review and provide an average sentiment score for each listing. The detailed process will be explained in section 5.1.3.

Dataset 3: Airbnb Singapore Listing Translated Reviews Dataset
Upon examining Dataset 2, the team noticed that some reviews are not in the English language which could interfere with the sentiment scores assigned to it. Hence, the team decided to utilise deep_translator from the GoogleTranslator library to translate such reviews into the English language, giving us Dataset 3. The team also found out that there were 77 rows with null reviews which the team then decided to drop. Additionally, there were also some reviews with empty strings ‘’, which the team decided to drop as well. Therefore, the team will have a reduced number of rows for Airbnb translated reviews with 10,893 and 7 columns, with the additional column to store sentiment scores.

Dataset 4: Singapore MRT Station Exits Coordinates Dataset

Dataset 4 is from the Data.gov.sg website (Data.gov.sg., 2019), containing the coordinates of MRT and LRT station exits in Singapore. The dataset contains 474 rows and 6 columns.

Dataset 5: Singapore Tourist Attractions Coordinates Dataset

Dataset 5 is from the Data.gov.sg website (Data.gov.sg, 2017), containing the coordinates of Tourist Attractions in Singapore. The dataset contains 106 rows and 23 columns.

Dataset 6: Singapore Malls Coordinates Dataset

Dataset 6 is from a GitHub online project (Lim, 2021) where they scraped the data for coordinates of shopping malls in Singapore. The dataset contains 183 rows and 3 columns.

4.2 Exploratory Data Analysis

To gain a better understanding of the Airbnb listings in Singapore, the team carried out Exploratory Data Analysis to better understand the data and the steps required to build the best model for price prediction. The team explored the number of neighbourhoods and regions in Singapore, together with the types of rooms and other factors that may potentially affect the pricing of a listing.

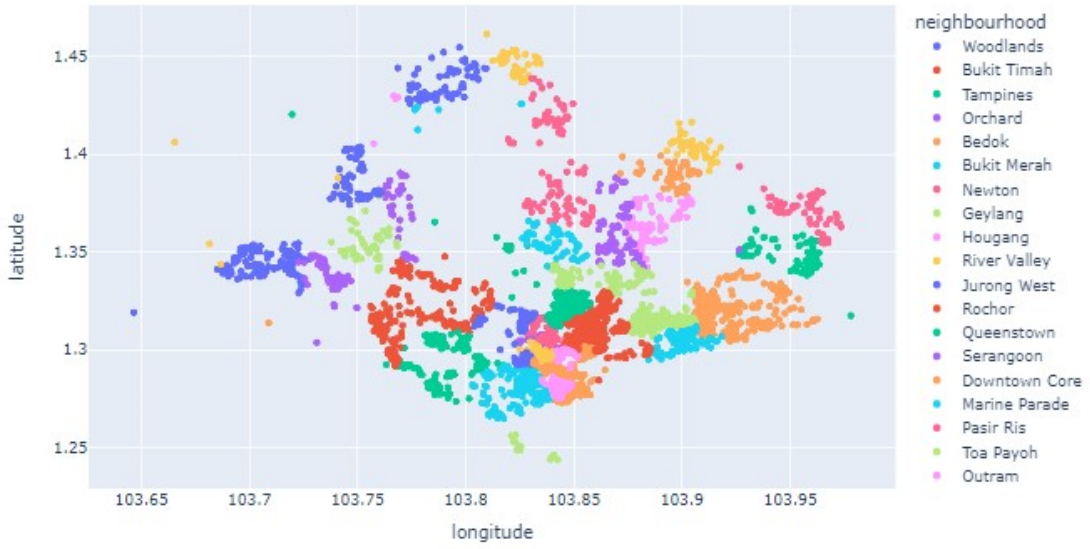

Figure 4.2(a) Number of Neighbourhoods in Singapore

Figure 4.2(a) shows the number of neighbourhoods that exist in Singapore. The team is able to identify 45 unique neighbourhoods - "Woodlands", "Bukit Timah", "Tampines", "Orchard", "Bedok", "Bukit Merah", "Newton", "Geylang", "Hougang", "River Valley", "Jurong West", "Rochor",

"Queenstown", "Serangoon", "Downtown Core", "Marine Parade", "Pasir Ris", "Toa Payoh", "Outram", "Punggol", "Tanglin", "Kallang", "Novena", "Bukit Panjang", "Singapore River", "Mandai", "Ang Mo Kio", "Bukit Batok", "Museum", "Sembawang", "Choa Chu Kang", "Clementi", "Central Water Catchment", "Jurong East", "Sengkang", "Bishan", "Yishun", "Southern Islands", "Sungei Kadut", "Western Water Catchment", "Tuas", "Marina South", "Lim Chu Kang", "Paya Lebar", "Boon Lay".

The team will continue to find more information about the neighbourhood such as the number of listings each neighbourhood has and the price amount to see any distinct differences between them.

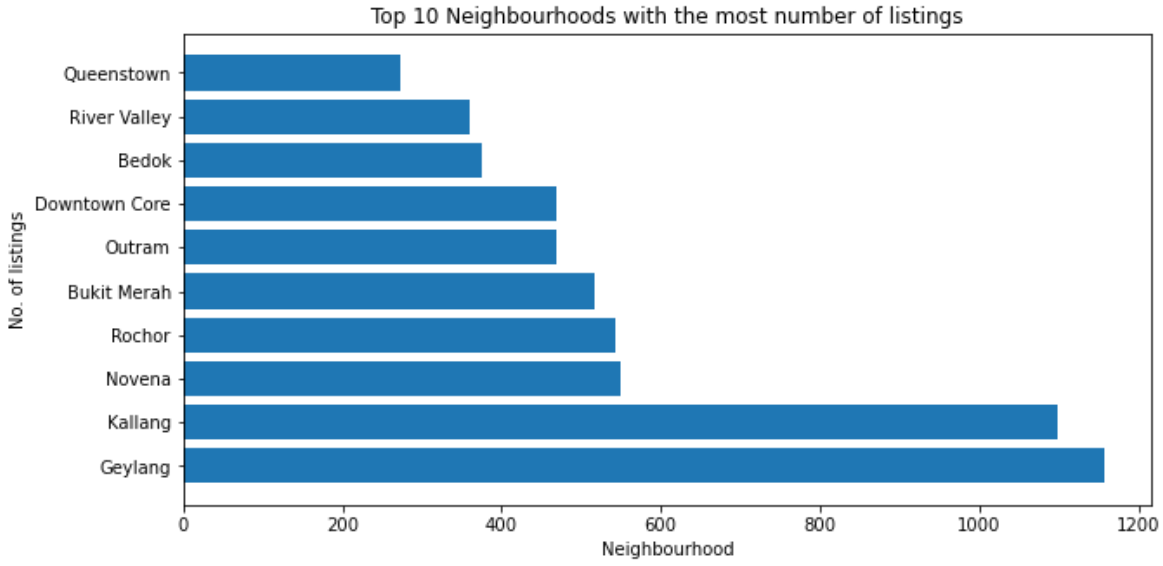

Figure 4.2(b) Top 10 neighbourhoods based on the number of listings

Figure 4.2(b) shows the top 10 neighbourhoods based on the number of listings. Geylang and Kallang are the top 2 on the list, while Queenstown has the lowest number of listings among the top 10. The team is able to observe that the top two listings have a significantly higher number of listings as compared to the other neighbourhoods. Hence, the team could expect that Geylang and Kallang would be the most popular neighbourhoods and could expect a huge amount of listings in these areas. This could also reflect that there could be more demand for listings in such areas where pricing could be more competitive. On the other hand, listers would know to avoid adding more listings in the least popular areas as there might be a chance it would be difficult to find renters.

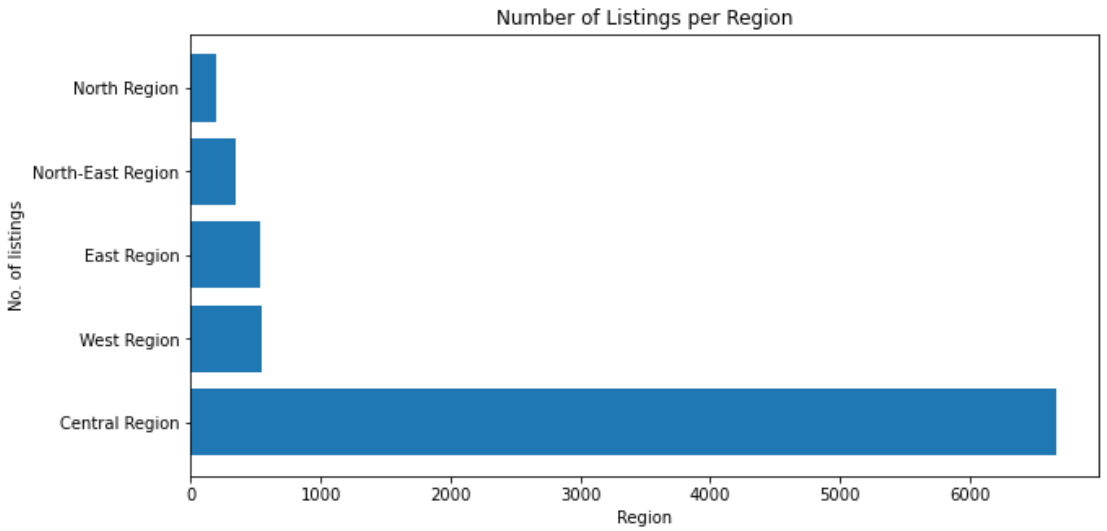

Figure 4.2(c) Number of Listings per Region

Figure 4.2(c) shows the number of listings per region. The team will be analysing 5 main regions in Singapore - North Region, North-East Region, East Region, West Region and Central Region. The team was able to notice that the Central Region has the highest number of listings whereas the North Region has the least number of listings. The team can also observe that the Central Region has a significantly higher number of listings than the other regions. Hence, the team could expect a Central Region listing to have a more expensive price due to the demand it has. Rationally, it is possible that the other regions have a cheaper price due to the significant differences in demand as compared to the Central Region.

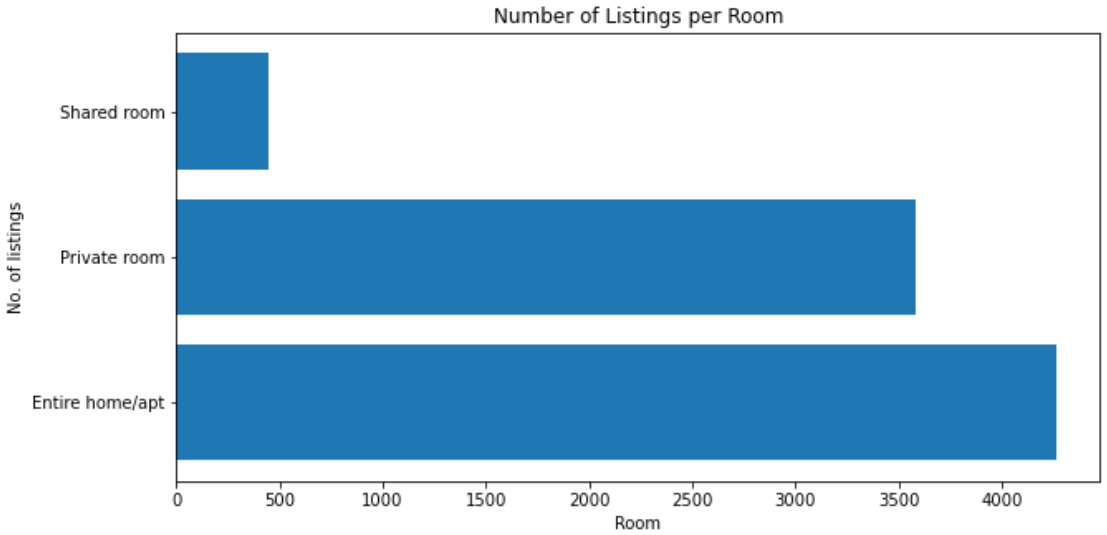

Figure 4.2(d) Number of Listings per Room

Figure 4.2(d) shows the number of listings per room type. In total, there are 3 types of rooms in Singapore - Shared Room, Private Room and Entire home/apartment. The team is able to identify the private room and entire home/apartment to have the highest number of listings. Hence, the team is able to have a glimpse of customer preferences for having privacy in their own space. This could be a potential factor that would influence the price of a listing.

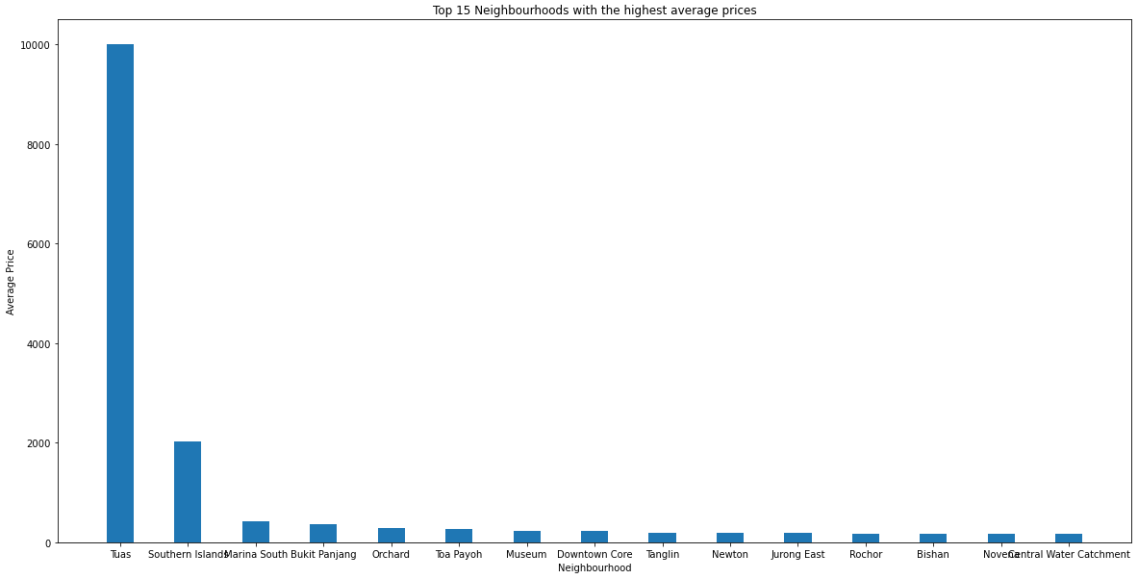

Figure 4.2(e) Top 15 neighbourhoods with the highest average price

From Figure 4.2(e), the team is able to obtain the Top 15 neighbourhoods with the highest average price. The team is able to note that most neighbourhoods have the same average price of listings, except Tuas, which has a very high average price range. Therefore, the team could expect the price range between the other neighbourhoods excluding Tuas and Southern Island (the second neighbourhood with the highest average price) to be the representative value of the average listing price. The team also noted that when they observed the average price in the neighbourhood level, they identified Tuas, a West Region neighbourhood to have the highest average price despite the Central Region dominating the number of listings. This may suggest that although a popular region might have the most number of listings which suggests a higher demand, there is still a need to take a look at the average listing price per neighbourhood as the area with the highest price listing might not necessarily be from the Central Region Hence, this shows the importance of analysing the data from the perspective of both the overall region and the individual neighbourhood of each listing.

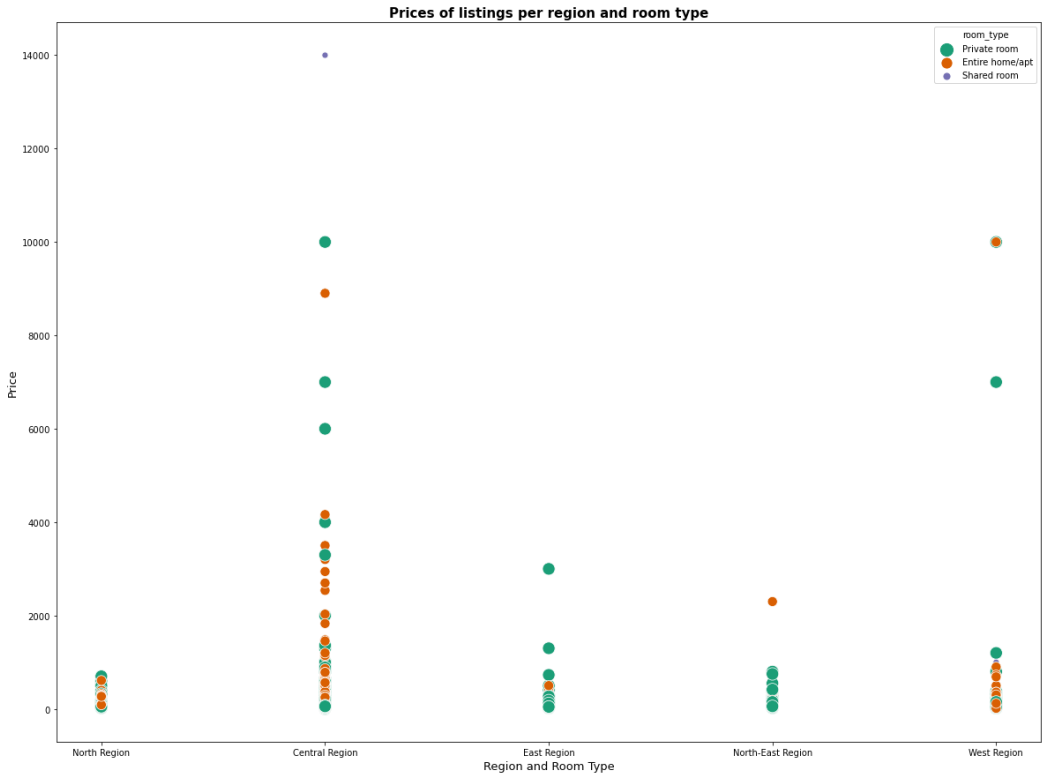

Figure 4.2(f) Price of listings per region & room type

Figure 4.2(f) shows the price of listings per region and room type. The team can see that the Central Region has the highest price range whereas the North Region has the smallest price range. There are also noticeable outliers in the Central and West Region. For the latter, it is possible that the Tuas neighbourhood is the outlier which had such a high price listing as analysed in an earlier section. The team can also for certain regions, the demand for the room type differs. For the central region, there seems to be a higher demand for the entire home/apartment whereas, for areas like the East and North-East region, there seems to be a higher demand for private rooms. This gives us insights to the different room type demands given the region.

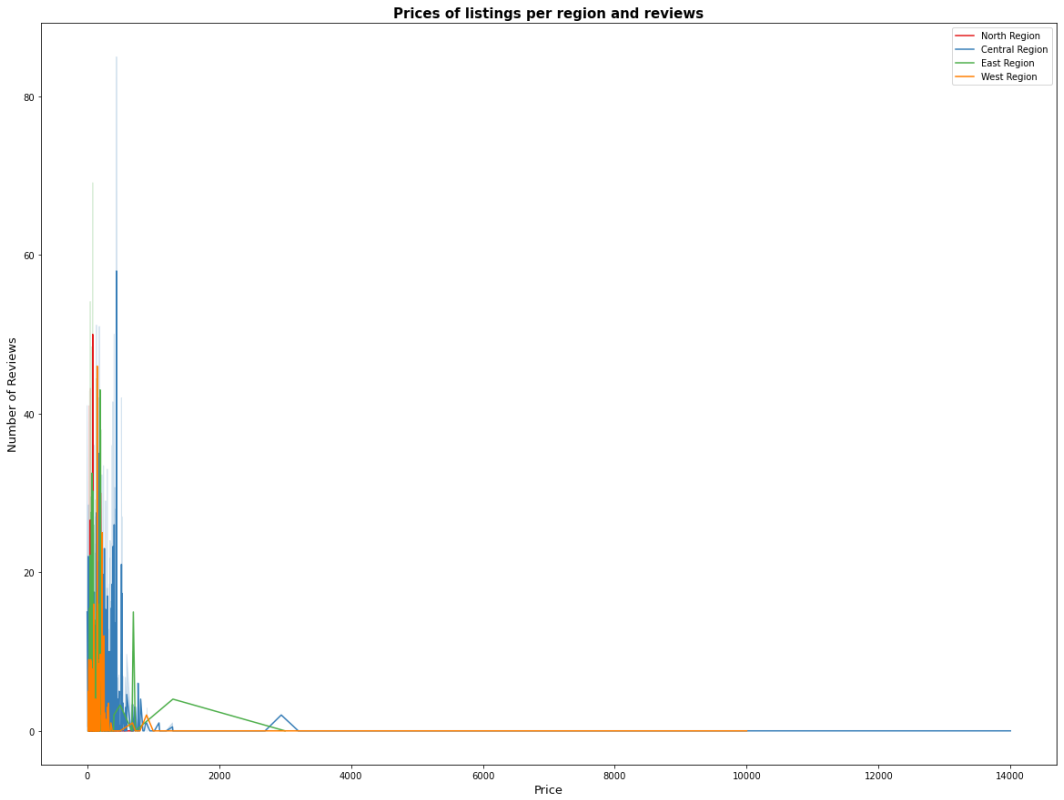

Figure 4.2(g) Price of listings per region and reviews

Figure 4.2(g) shows the price of listings per region and reviews. As the team will be conducting sentiment analysis on the reviews of the listings, is it important for us to understand the number of reviews received by the listings in the dataset. The team is able to notice that the listings with the lower prices have a higher number of reviews. This could be possible due to the fact that cheaper listings may potentially have more renters. Therefore, it is important to note the lack of reviews of certain listings with higher price.

4.3 Data Pre-processing

Before fitting the data into the models, the team decided to pre-processed the data through manual feature selection, handling missing data and outliers, transformation of variables and label encoding categorical variables.

4.3.1 Manual Feature Selection

The listings dataset had 98 columns however some of the columns were not of much use such as URLs. Therefore the team removed any columns that were URLs. The team also removed all IDs except for the listing ID. Columns with too many null values were also dropped. Some categorical

variables did not provide much insight as all the values were the same, in that case the team dropped such variables as well.

4.3.2 Handling Missing Data and Outliers

There were some missing values for binary variables such as ‘Host_is_superhost’ and ‘host_identity_verified’, so the team imputed the missing values with ‘u’ so as to differentiate the missing values from true and false values. There were also some missing values for ‘reviews_per_month’, so the group made the assumption that there were no reviews for those listings, hence the team imputed the null values with 0. For missing values in the ‘price’ variable after dropping the outliers the team imputed the price using the mean price as there were only 34 missing values.

The team found that the ‘price’ variable had outliers. Most prices were concentrated around the 100−400 range but a few listings were more than $12000. The team then filtered out any excessive prices or undercharge prices by using the three SD range rule which removes values that are three standard deviations below or above the mean.

4.3.3 Transformation of Variables

From the price distribution plot, the team is able to observe that the price distribution is skewed to the right. The team would like to have a normal distribution so as to have a stable learning process and avoid having a large spread of values. Hence the team applied log transformation to the ‘price’ variable.

4.3.4 Label Encoding Categorical Variables

For categorical variables in the dataset such as 'host_is_superhost', 'neighbourhood', 'region', 'host_identity_verified', 'is_location_exact', 'property_type', 'room_type', 'bed_type', the team applied label encoding using LabelEncoder() from sklearn. The team chose to use label encoding as some of the variables such as neighbourhood have many unique values and if one-hot encoding were to be used it would result in a high-dimensional feature space, which can increase the risk of overfitting.

4.3.5 Correlation Matrix

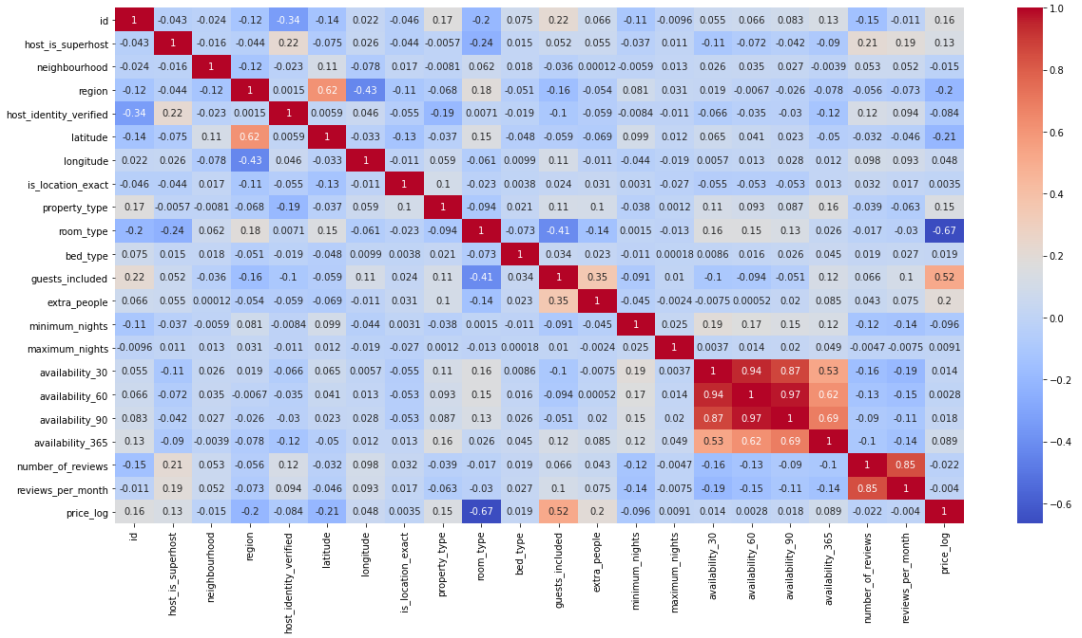

Figure 4.3.5(a) Correlation Matrix

The team plotted a correlation matrix (Figure 4.3.5(a)) in order to check which features are most correlated to price and should be included in the final model.

No features have a very high correlation to the attribute price. The top 3 features with the highest correlation were guests_included (0.52), extra_people (0.2), and property_type (0.15).

Since there existed some multicollinearity between variables, 5 availability features were reduced to 2, i.e, the team only chose availability_30 and availability_365 to be included in the final model as only availability_30 and availability_365 had lower collinearity to the other availability features. The rest were removed.

4.3.6 Residual Plots

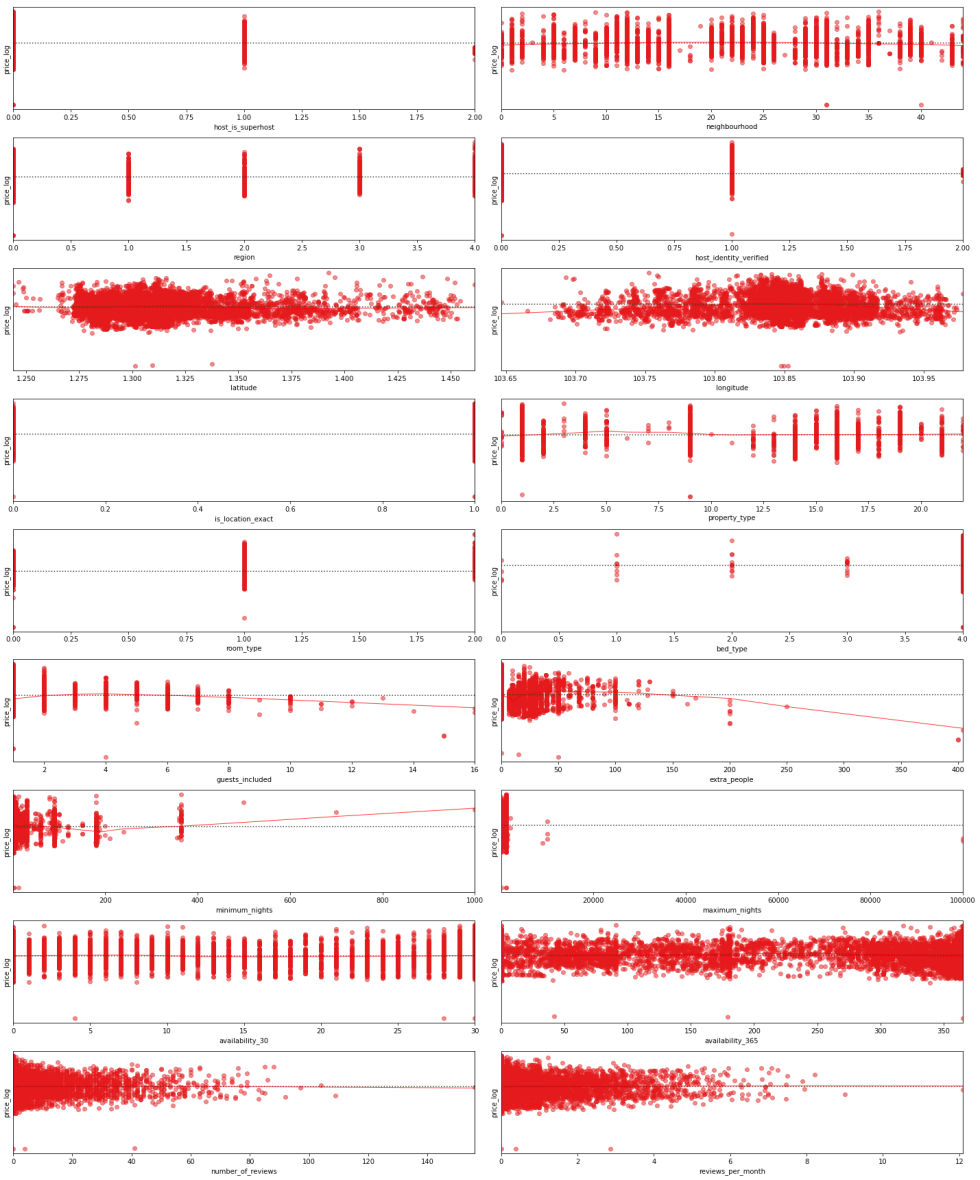

Figure 4.3.6(a) Residual Plots

Residual plots (Figure 4.3.6(a)) were plotted to check for non-linearity in the data and to detect any outliers that still exist in the data. The plots above indicate the presence of a non-linear relationship between the dependent variable (price) and other independent variables. This helped the team understand that a linear model should not be used to predict price. The plots also show that there are no independent variables that have a very strong relationship, individually, with the price. Hence, the final model must make use of multiple variables to determine an optimal price for a listing.

5. Methodology

For the process of feature engineering and model building, the team utilised an AI personal communication tool, ChatGPT March 2023 version to assist them in debugging, generating codes and understanding the concepts and key metrics that will be used during the project.

5.1 Feature Engineering

Feature Engineering is the process of selecting and transforming raw data into features that can be used as inputs to a machine learning algorithm. The goal of feature engineering is to extract meaningful and relevant information from the data that will improve the accuracy and performance of the model. In this project, the team has utilised 3 types of feature engineering - K-Means Clustering, Distance Calculation and Sentiment Analysis.

The goal of each feature engineering can be summarised as follows:

1) **K-Means Clustering:** To group listings with MRT station exits and tourist attractions based on their coordinates
2) **Distance Calculation:** To calculate the distance between the listings and the nearest MRT, the number of malls within a 2km radius of a listing, and the number number of tourist attractions within a 5km radius of listing,
3) **Sentiment Analysis:** To obtain the average sentiment score of each reviews of a listing.

5.1.1. K-Means Clustering

The K-means algorithm is a simple and widely used clustering method that partitions data into k clusters based on their similarity. The algorithm iteratively assigns each data point to the cluster whose centroid is closest to it and recalculates the centroids until convergence. (Jin & Han, 2011)

The team decided to perform K-means clustering so as to add additional information to the dataset regarding which cluster each listing belongs to. The team believes the cluster analysis helps the models which the team built later on to identify similar listings more easily.

In this feature engineering method, the coordinates of MRT station exits and tourist attractions will be added to a temporary dataframe together with the coordinates of the Airbnb listings. K-Means clustering will be performed on the temporary dataset based on the coordinates and generate the cluster grouping.

Firstly, the team created 2 dataframes (mrt_df & att_df) that contain the MRT and tourist attractions coordinates. After combining the dataframes into the overall temporary dataframe, the team also added a 'type' column which identifies whether a particular coordinate is an Airbnb listing, an MRT exit or an attraction. Specifically, type= 0 indicates an Airbnb listing, type = 1 indicates an MRT exit and type = 2 indicates an attraction.

The team plotted an elbow curve to identify the value to use for the number of clusters. From the plot the team decided to go with 5 clusters.

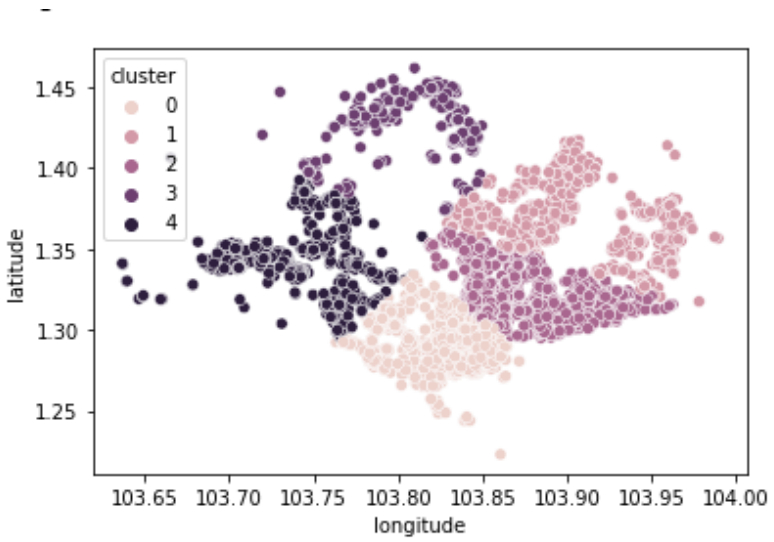

Figure 5.1.1.(a) Scatter plot of the K-Means Cluster grouping

When the team visualised the clusters created on a map (Figure 5.1.1(a)), they could see the clusters almost resembling different regions of Singapore such as North, South, West and East with the East side containing 2 clusters.

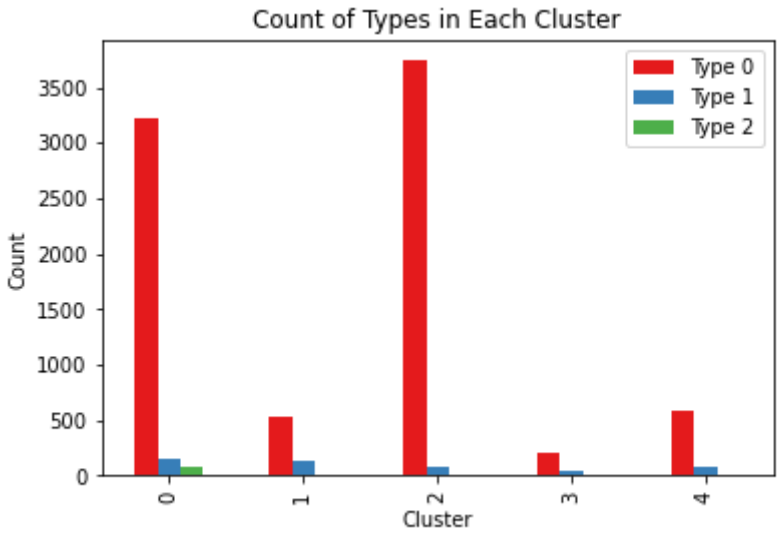

Figure 5.1.1(b) The number of Listings, MRT exits and Tourist Attractions in each cluster

The team further analysed the clusters by counting the number of MRT exits, attractions and listings in each cluster. From the bar chart (Figure 5.1.1(b)), the team can see that most of the listings are in cluster 0 and cluster 2 which are the south and east regions respectively. Furthermore the team noticed that a large majority of the attractions were located in cluster 0 which is the south region.

5.1.2. Distance Calculation

The team used the Haversine formula to calculate the distances to places of interest such as MRT stations, tourist attractions, and shopping malls.

For MRT stations, the team decided to find the distance to the nearest station.

For tourist attractions, the team set a radius of 5km from a particular listing's location and counted the number of attractions within the set distance.

For shopping malls, the team set a radius of 2km from a particular listing's location and counted the number of malls within the set distance. The team decided to use 2km so that they are able to find malls that are within walking distance to each listing.

5.1.3 Sentiment Analysis

The Airbnb Singapore Listing Reviews dataset (Dataset 2) contains multiple reviews for each listing. The team decided to carry out sentiment analysis and assign sentiment scores to each listing, to help with predicting the price for that listing.

Data cleaning:
As the first step, the team analyzed the dataset to look for any null values or missing values in the reviews column. 80 rows out of 100919 contained missing values, and hence these were removed from the dataset. The team also carried out the manual deletion of rows that had integer values or were not words or sentences. After these steps, the dataset contains 100835 rows with 6 columns.

Translation of Reviews to English:
During the analysis of the dataset, the team noticed that not all reviews were written in English. This can affect the sentiment analysis model, thus affecting the final model to predict price. Hence, the team carried out translation on all 100835 rows using Google Translate, from the Deep Translator package. As Google Translate has a limit of 5000 characters per input, the first 5000 characters were extracted from reviews that were longer.

TextBlob for Sentiment Analysis:
Once every review was translated to English, the team used TextBlob to assign a sentiment score (between -1 and 1) for each review in the dataset.

Grouping by Listing ID and Analysis:
The dataset was then grouped by listing ID and the mean sentiment score for that listing. The team performed some EDA on the grouped reviews dataframe and got the following insights:

- *Number of listings with negative reviews (score < 0): 63*
- *Number of listings with neutral reviews (score = 0): 199*
- *Number of listings with positive reviews (score > 0): 4894*
- *Listing ID with the most positive review (highest score): 8247580*
- *Listing ID with the most negative review (lowest score): 33915290*

As a final step for sentiment analysis, the sentiment scores were appended to Dataset 1 (Airbnb Singapore Listing Dataset) for scaling.

5.2 Min-Max Scaling and K-Fold Cross Validation & GridSearchCV hyperparameter Tuning

Before trying out the machine learning models, the dataset is further prepared by scaling the variables to ensure all variables are on the same scale. Due to a limited number of samples, K-Fold Cross Validation was also utilized to further utilize every data point to act both as training and test sets.

Min-Max Scaling

The team decided to perform scaling on the variables due to the differences of scale it has. At this stage, all of the variables are of the int64 types as all of the categorical variables have been label encoded. Figure 5.2(a) summarises the variables and the range of values it has. The label encoded variables - "neighbourhood", "region", "host_identity_verified", "is_location_exact", "property_type", "room_type", "bed_type", are not included in the table as scaling does not have any effect on them as they do not contain any meaningful numerical relationships with one another.

	latitude	longitude	guests_included	extra_people	minimum_nights	maximum_nights	availability_30	availability_365
0	1.44255	103.79580	1	14	180	360	30	365
1	1.33235	103.78521	2	20	90	730	30	365
2	1.44246	103.79667	1	14	6	14	30	365
3	1.34541	103.95712	4	27	1	1125	25	353
4	1.34567	103.95963	1	20	1	1125	25	353
5	1.34702	103.96103	1	20	1	1125	25	348
6	1.34348	103.96337	4	34	1	365	9	155
7	1.30175	103.83609	9	120	7	1000	30	365
8	1.32304	103.91383	1	0	30	1125	0	229
9	1.32458	103.91163	1	0	30	1125	0	206

Figure 5.2(a) Values before scaling for the first eight numerical variables

number_of_reviews	reviews_per_month	cluster	nearest_mrt	num_attractions	num_malls	sentiment_scores	price_log
0	0.01	3	0.570182	4	6	0.210000	4.406719
0	0.28	4	0.409590	4	2	0.434563	4.394449
0	0.21	3	0.477753	3	6	0.415570	4.234107
2	0.13	1	0.477559	1	5	0.329667	5.303305
0	0.21	1	0.433381	1	5	0.266197	4.532599
8	0.35	1	0.557524	1	4	0.326727	4.634729
3	0.23	1	0.255565	1	2	0.295912	5.318120
0	0.16	0	0.285156	77	43	0.432629	4.795791
1	1.92	2	0.225091	4	2	0.407484	3.951244
3	2.13	2	0.406443	4	2	0.400790	4.077537

Figure 5.2(b) Values before scaling for the last eight numerical variables

From Figure 5.2(a) and Figure 5.2(b), the team is able to see the difference in scale between the variables. For instance, the team is able to see variables like “minimum_nights” to have values as large as 180 while variables like “availability_30” have values as low as 0. According to Loukas 2020, having variables of different ranges may affect the model-fitting process as every variable would not be weighted equally. This may result in a bias where some variables may affect the model performance more despite not necessarily having a greater impact. Therefore, the team is unsure of how much the difference in scale will affect the model performance and decided to conduct scaling of variables to ensure all the variables are in the same range.

Therefore, the team performed scaling on only the numerical variables of the dataset such as 'guests_included','extra_people','minimum_nights','maximum_nights','availability_30','availability_365','number_of_reviews','reviews_per_month','price_log','nearest_mrt','num_attractions','num_malls','sentiment_scores' because it is not recommended to scale the label encoded categorical variables.

The scaling was performed using MinMaxScaler from sklearn. With the Min-Max scaler, the variables will be converted to a range of 0 to 1 (Loukas, 2020). This will be desirable for the project as variables such as price will not go below 0, as it is impossible to have such a value.

The team decided to utilize such a scale as it prevents variables from having negative values as it would be impossible to have negative values for variables such as price. The team also removed the ID of the listings which was a numerical variable from the dataframe to be used for model building as it may hinder the model performance.

K-Fold Cross-Validation

K-fold cross-validation is a technique used in machine learning to evaluate the performance of a predictive model. It provides a more accurate estimate of the performance of a model than simply using a single train/test split. By repeating the process with different folds, the team can get a better sense of the model’s generalisability (Pandian, 2022). The team used 5 fold cross validation for our project.

GridSearchCV Hyperparameter Tuning

GridSearchCV is a method for hyperparameter tuning that performs an exhaustive search over a specified range of hyperparameters and returns the combination that results in the best performance according to a specified metric. The method works by constructing a grid of hyperparameters and

evaluating the model performance for each combination of hyperparameters using cross-validation. (Sharma, 2020). The team decided to use GridSearchCV for hyperparameter tuning of the models.

5.3 Regression Models

For the regression models, the team has utilized a couple of regression models such as Linear, Lasso, Ridge, Elastic Net (ENET), and K-Nearest Neighbours (KNN) Regression for baseline models. Due to the underwhelming performance of the mentioned models which will be expanded later, the team then decided to explore bagging and boosting methods to obtain a better model performance.

With a guidance with ChatGPT, the evaluation metrics that the team used for evaluation the models performance are:

- **Root Mean Squared Error (RMSE):** Measures the average error between the predicted price and the actual price. The lower the value, the better the model performance is.
- **Mean Absolute Error (MAE):** Measures the average error between the predicted price and the actual price through the use of absolute values. Additionally, it is less sensitive to outliers than RMSE.
- **R-squared (R2):** Measures the proportion of variance in the dependent variable that is explained by the independent variables in the model. A higher value of R2 would indicate a better model performance.
- **Adjusted R-squared (Adj R2):** Slightly different from R-squared value as it takes into account the number of independent variables and will get affected when more independent variables are added to it.

5.3.1 Linear Regression

Linear Regression is a supervised learning method that aims to predict a continuous target variable based on labelled training data. The aim of the model will be to find the linear line that would best describe the relationship between the predictors and the target variable (Deepanshi, 2021). There are two types of linear regression - Simple and Multiple. The former would only require one dependent variable whereas the latter would have more than one dependent variable. Additionally, the linear regression model would aim to minimize the error which is the difference between the actual and predicted values. (Deepanshi, 2021).

Implementation of Linear Regression

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.0682863988309986
Average MAE: 0.04932973142224716
Average R2: 0.5658213994492831
Average Adjusted R2: 0.564310733278852


TEST SET SCORES:
---------------------
Average RMSE: 0.06887071557309285
Average MAE: 0.04965631611246924
Average R2: 0.557835463246364
Average Adjusted R2: 0.5516138714556833
```

Figure 5.3.1(b) Airbnb Prediction Linear Regression Model results

With reference to Figure 5.3.1(b), When we conducted linear regression on the dataset we realised that the average RMSE and MAE depicts a good score as it typically means that the model's predicted values are only off by 0.07 or 0.05 units away from the actual values. However, the average R squared and adjusted R squared value suggested otherwise with the model only able to account for 55.2% percent of the variation in the target variable with regards to the relationship between the predictor variables.

5.3.2 Lasso Regression

Lasso Regression is a regularisation technique which is used over regression models to have better performance. This regularisation method introduces a penalty term that equals the absolute value of the line's coefficient. Hence, models with coefficient values closer than zero will have a bigger penalty (Great Learning Team, 2023).

Implementation of Lasso Regression

```
TRAIN SET SCORES:
--------------------
Average RMSE: 0.07845380993166931
Average MAE: 0.060368178603370806
Average R2: 0.42690954747253346
Average Adjusted R2: 0.4249155564041067


TEST SET SCORES:
--------------------
Average RMSE: 0.07999421084218376
Average MAE: 0.060458584195423014
Average R2: 0.42375139264826633
Average Adjusted R2: 0.4156401821408674
Best alpha value: 0.01
```

Figure 5.3.2(a) Airbnb Prediction Lasso Regression Model results

With reference to Figure 5.3.2(a), the team received even poorer average R squared and average adjusted R squared values compared to Linear Regression when we conducted lasso regression on the dataset.

However, the average R squared and adjusted R squared value suggested otherwise with the model only able to account for about 41.6% percent of the variation in the target variable with regards to the relationship between the predictor variables.

Based on hyperparameter tuning, 0.01 was the best value to set for the regularisation parameter alpha.

5.3.3 Ridge Regression

Ridge Regression is a linear regression algorithm that uses L2 regularization to prevent overfitting. It adds a penalty term to the cost function to reduce the magnitude of the coefficients, making the model more generalizable to new data. (Mohan, 2021)

Implementation of Ridge Regression

```
TRAIN SET SCORES:
--------------------
Average RMSE: 0.0711409901713128
Average MAE: 0.05655722697750075
Average R2: 0.5280062988010742
Average Adjusted R2: 0.47563980833407465


TEST SET SCORES:
--------------------
Average RMSE: 0.07139538646575988
Average MAE: 0.051838679864137045
Average R2: 0.524267100611781
Average Adjusted R2: 0.528920091898252


Best alpha value: 0.1
```

Figure 5.3.4(a) Airbnb Prediction Ridge Regression Model results

With reference to Figure 5.3.4(a), when we conducted Ridge regression on the dataset, the team received poor average R squared and average adjusted R squared values where the scores were lower than the Linear Regression model. From our results, the average R squared and adjusted R squared value suggests that the model is only able to account for 52.9% percent of the variation in the target variable with regard to the relationship between the predictor variables.

Based on hyperparameter tuning, 0.1 was the best value to set for the regularisation parameter alpha.

5.3.4 Elastic Net (ENET) Regression

Elastic Net Regression is a linear regression algorithm that combines L1 and L2 regularization. It adds both the L1 and L2 penalties to the cost function to balance between sparsity and reducing the magnitude of the coefficients. The strength of the penalties is controlled by two hyperparameters called the mixing parameter and regularization parameter (Mohan, 2021).

Implementation of ENET Regression

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.07399430894499234
Average MAE: 0.05896611908947689
Average R2: 0.49020735070146665
Average Adjusted R2: 0.4460882362679998


TEST SET SCORES:
---------------------
Average RMSE: 0.07409022732540092
Average MAE: 0.05295100269252763
Average R2: 0.4884682490062359
Average Adjusted R2: 0.5164422596427898


Best alpha value: 0.01
Best l1_ratio value: 0.2
```

Figure 5.3.3(a) Airbnb Prediction ENET Regression Model results

With reference to Figure 5.3.3(a), when we conducted Elastic Net regression on the dataset, the team received poor average R squared and average adjusted R squared values where the scores were lower than the Linear Regression model. From our results, the average R squared and adjusted R squared value suggests that the model is only able to account for 51.6% percent of the variation in the target variable with regard to the relationship between the predictor variables.

Based on hyperparameter tuning, 0.01 was the best value to set for the regularisation parameter alpha and 0.2 for the l1_ratio_value. The l1_ratio_value determines the balance between L1 (Lasso) and L2 (Ridge) regularization

5.3.5 K-Nearest Neighbours (KNN) Regression

K Nearest Neighbors (KNN) regression model is a machine learning algorithm used for predicting numerical values, like Airbnb prices. In KNN, the algorithm finds the k closest data points to a new data point, based on certain features. Then, it averages the target variable of those k data points to predict the target value for the new data point. (Singh, 2023)

Implementation of KNN Regression

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.07021887618687633
Average MAE: 0.0551109973454372
Average R2: 0.5402377856324628
Average Adjusted R2: 0.49314908545121594


TEST SET SCORES:
---------------------
Average RMSE: 0.07037524386597345
Average MAE: 0.05121054354074016
Average R2: 0.5377727931854185
Average Adjusted R2: 0.5363551618314737


Best hyperparameters: {'n_neighbors': 10}
```

Figure 5.3.5(a) Airbnb Prediction KNN Regression Model results

With reference to Figure 5.3.5(a), When we conducted K-Nearest Neighbours regression on the dataset, the team received poor average R squared and average adjusted R squared values where the scores were lower than the Linear Regression model. From our results, the average R squared and adjusted R squared value suggests that the model is only able to account for 53.6% percent of the variation in the target variable with regard to the relationship between the predictor variables.

Based on hyperparameter tuning, 10 neighbours was the best value to set for the number of neighbours in the model.

5.3.6. Decision Trees Regression

A decision tree regression model is a type of machine learning algorithm used for predicting numerical values. In the context of Airbnb price prediction, it involves creating a decision tree based on the features to predict the price of a listing. The model recursively splits the data based on the most significant features until a leaf node is reached, which represents the predicted price. (Prasad, 2021)

Implementation of Decision Trees Regression

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.0697578191946581
Average MAE: 0.05414684425739482
Average R2: 0.546353529048157
Average Adjusted R2: 0.5048219368626434


TEST SET SCORES:
---------------------
Average RMSE: 0.06986517256608024
Average MAE: 0.05083366174670203
Average R2: 0.5445256394722374
Average Adjusted R2: 0.5408162037914067


Best hyperparameters: {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_
split': 20}
```

Figure 5.3.6(a) Airbnb Prediction Decision Trees Regression Model results

With reference to Figure 5.3.6(a), when we conducted Decision Trees regression on the dataset, the team received poor average R squared and average adjusted R squared values where the scores were lower than the Linear Regression model. From our results, the average R squared and adjusted R squared value suggests that the model is only able to account for 54.1% percent of the variation in the target variable with regard to the relationship between the predictor variables.

Based on hyperparameter tuning, the best values for the parameters were 5 max depth, 1 minimum samples leaf, and 20 minimum sample split.

5.4 Bagging and Boosting Methods

Bagging is a technique that creates an ensemble of multiple decision trees to predict the price of an Airbnb listing based on its features. Each decision tree is built on a random subset of the training data and the features. Bagging improves the accuracy of the model by reducing the variance and overfitting. The final prediction is the average of the predictions of all the decision trees in the ensemble (Amrit, 2020).

5.4.1 Standard Bagging Method

Implementation of Standard Bagging Method

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.06948118499932715
Average MAE: 0.0534581634802217
Average R2: 0.5500229750975736
Average Adjusted R2: 0.5131596878708059


TEST SET SCORES:
---------------------
Average RMSE: 0.06955912978614431
Average MAE: 0.05058240721734327
Average R2: 0.5485773472443287
Average Adjusted R2: 0.5437902317646954


Best hyperparameters: {'base_estimator__max_depth': 20, 'base_estimator__mi
n_samples_leaf': 4, 'base_estimator__min_samples_split': 2, 'n_estimators':
50}
```

Figure 5.4.1(a) Airbnb Prediction Standard Bagging Model results

With reference to Figure 5.4.1(a), when we conducted the standard bagging method on the dataset, the team received poor average R squared and average adjusted R squared values where the scores were lower than the Linear Regression model. However, the bagging method did perform better than other models such as K Nearest Neighbors and the Decision Tree model. This could be due to the fact the bagging method is an ensemble method and hence its ability to reduce variance and increase stability. From our results, the average R squared and adjusted R squared value suggests that the model is able to account for 54.4% percent of the variation in the target variable with regard to the relationship between the predictor variables.

Based on hyperparameter tuning, the best values for the parameters were 20 base estimator max depth, 4 base estimator minimum sample leaf nodes, 2 base estimator minimum sample splits, and 50 n_estimators.

5.4.2 AdaBoost

Adaboost is a boosting algorithm that uses an ensemble of weak learners, such as decision trees, to predict the price of an Airbnb listing based on its features. It assigns weights to the training instances based on their error and focuses on the misclassified instances in subsequent iterations. Adaboost

improves the accuracy of the model by combining multiple weak models into a strong one. (Brownlee, 2020)

Implementation of AdaBoost Method

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.06963830416551722
Average MAE: 0.0531356744912058
Average R2: 0.5480550659095665
Average Adjusted R2: 0.5160908272401796


TEST SET SCORES:
---------------------
Average RMSE: 0.06966488213312055
Average MAE: 0.05043765493546368 4
Average R2: 0.5498646400614613
Average Adjusted R2: 0.5446852912915595


Best hyperparameters: {'learning_rate': 0.01, 'n_estimators': 10}
```

Figure 5.4.2(a) Airbnb Prediction AdaBoost Model results

With reference to Figure 5.4.2(a), when we applied the AdaBoost algorithm to the dataset, the team received poor average R squared and average adjusted R squared values where the scores were lower than the Linear Regression model. The AdaBoost model performed similarly to the Bagging model. From our results, the average R squared and adjusted R squared value suggests that the model is able to account for 54.5% percent of the variation in the target variable with regard to the relationship between the predictor variables.

Based on hyperparameter tuning, the best values for the parameters were 0.01 learning rate and 10 n_estimators.

5.4.3 XGBoost

XGBoost is a gradient-boosting algorithm that uses an ensemble of decision trees to predict the price of an Airbnb listing based on its features. It is known for its high accuracy and ability to handle complex non-linear relationships between the features and the target variable. XGBoost incorporates techniques like regularization, tree pruning, and boosting to prevent overfitting and improve the generalization of the model (Brownlee, 2021).

Implementation of XGBoost Method

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.04316442861206191
Average MAE: 0.02993097587765613
Average R2: 0.8245130022424455
Average Adjusted R2: 0.8239024221774841
TEST SET SCORES:
---------------------
Average RMSE: 0.056645086001764
Average MAE: 0.037679257713994094
Average R2: 0.7007761925484249
Average Adjusted R2: 0.6965659132221249
Best hyperparameters: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimator
s': 100}
```

Figure 5.4.3(a) Airbnb Prediction XGBoost Model results

With reference to Figure 5.4.3(a), when we applied XGBoost algorithm on the dataset, the team received good average R squared and average adjusted R squared values where the scores were the highest of all the models. From our results, the average R squared and adjusted R squared value suggests that the model is able to account for 69.7% percent of the variation in the target variable with regard to the relationship between the predictor variables.

Based on hyperparameter tuning, the best values for the parameters were 0.1 learning rate, 5 max depth and 100 n_estimators.

5.5 Feature Importance

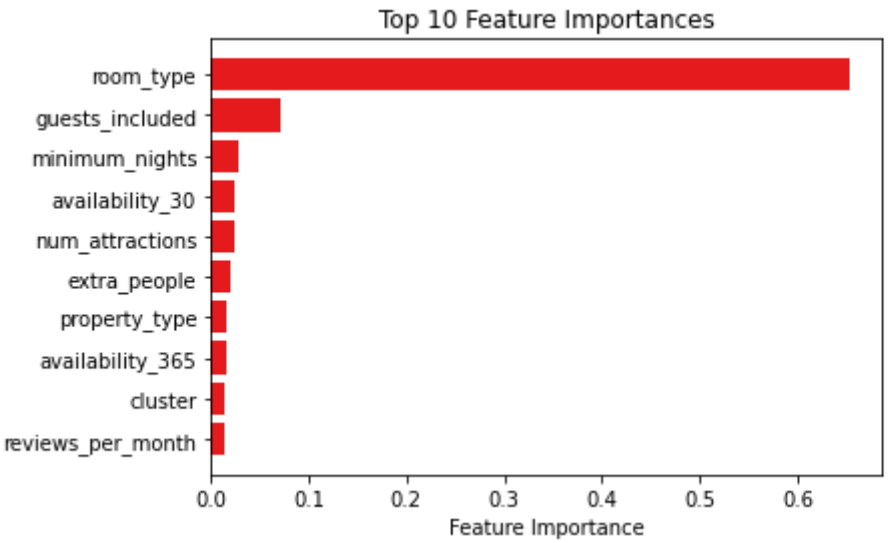

Figure 5.5(a) Feature Importance on XGBoost model

From our feature importance plot that was plotted from our XGBoost model (Figure 5.5(a)), we can see that room_type is the most important. By doing a feature importance analysis we can find out which features are contributing the most to the model's prediction and in the context of our project it is also able to give us some insight into which aspects of the listing contribute to the price of the listing. This helps us to achieve one of our motivations for doing this project which is to clearly explain the derivation of prices to be more objective in pricing decisions.

5.6 Principal Component Analysis (PCA)

```
TRAIN SET SCORES:
---------------------
Average RMSE: 0.05654596232648192
Average MAE: 0.04210899928890642
Average R2: 0.7018842750158526
Average Adjusted R2: 0.5654288437085959


TEST SET SCORES:
---------------------
Average RMSE: 0.06979707256684084
Average MAE: 0.05024465189295757
Average R2: 0.5514737560828766
Average Adjusted R2: 0.5458787039940449


Best hyperparameters: {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 5
0}
```

Figure 5.6(a) Airbnb Prediction XGBoost Model on PCA Dataset

PCA (Principal Component Analysis) is a technique used for dimensionality reduction in machine learning and data analysis. The main goal of PCA is to transform a high-dimensional dataset into a lower-dimensional space while retaining as much of the variation in the data as possible.

We performed PCA on our best-performing model which was the XGBoost model, however, the results were poorer than the results we got for the XGBoost model (refer to Figure 5.6(a)). The adjusted R2 value dropped from 0.6966 to 0.5459. This could be due to the fact that there were not many variables correlated to the price variable as observed from our correlation map (refer to Figure 4.3.5(a)).

Based on hyperparameter tuning, the best values for the parameters were 0.1 learning rate, 7 max depth, and 5 n_estimators.

6. Results and Discussion

6.1 Summary of Results

All our dimensionality reduction models such as Lasso Regression, Ridge Regression, Elastic Net Regression, and PCA performed poorly when compared to the Linear Regression model.

The best-performing model was XGBoost. The test scores can be summarised in Figure 6.1(a):

Models	RMSE	MAE	R2	Adj R2
Linear Regression	0.07	0.05	0.56	0.55
Lasso Regression	0.08	0.06	0.42	0.42
ENET Regression	0.07	0.06	0.49	0.52
Ridge Regression	0.07	0.05	0.52	0.53
KNN Regression	0.07	0.05	0.54	0.54
Decision Trees Regression	0.07	0.06	0.54	0.54
Standard Bagging Method	0.07	0.05	0.55	0.54
AdaBoost Method	0.07	0.05	0.55	0.54

XGBoost Method (PCA)	0.07	0.05	0.55	0.55
XGBoost Method	**0.06**	**0.04**	**0.70**	**0.70**

***Best Performing Model**

Figure 6.1(a) The evaluation scores of each model

7. Conclusion and future work

7.1 Conclusion

Based on the results of our project, the team found that XGBoost was the best model for predicting airbnb listing price with an R2 score of 0.6966 and an RMSE score of 0.057. Although it does not achieve an extremely high score, the team believed that this is the best optimal solution due to the nature of the dataset which has variables that do not have a strong correlation with the price. This is apparent by the correlation matrix in the above section where it is observed that there are no correlations of more than 0.52 between variables and the price. Additionally, the models used show signs of underfitting due to poor training scores in most models.

7.2 Limitations

Throughout the study, the team found a couple of limitations that could be addressed in future studies in the pursuit of better results. The limitations are the method of calculating distances, lack of interpreting a listings' amenities availability and the lack of interpreting the

Currently the distances to places of interest are calculated using the haversine formula, however in reality the distances could be different based on road networks and walkways in Singapore.

There is information in the dataset such as amenities available in a particular listing which includes information about whether things like Television, Internet, Air-conditioning and more are available in the listing. However, our project currently does not take this into account.

Furthermore, the dataset also provides other data such as a written description of the listings and house rules of listings which could provide valuable information on whether someone finds the listing

suitable. However, since this information was textual data, there was no standardized way where the team could take this information into consideration when building our prediction model.

7.3 Future Works

The team could explore using deep learning, specifically neural networks to predict the listing price. This is because neural networks would be able to make better sense of non-linearity in the data.

The dataset also provides the listing image for the listings and with that data, the team could explore using computer vision to find out if certain features of the images have any impact on the listing price.

We could also perform a time series analysis as prices can be influenced by various factors that change over time, such as seasonal trends, days of the week, holidays, and events happening in the local area.

8. References

Amrit, A. (2020, October 13). Bagging on low variance models. Medium. https://towardsdatascience.com/bagging-on-low-variance-models-38d3c70259db

Brownlee, J. (2020, May 1). How to develop an ADABOOST ensemble in python. MachineLearningMastery.com. https://machinelearningmastery.com/adaboost-ensemble-in-python/

Brownlee, J. (2021, March 6). XGBoost for regression. MachineLearningMastery.com. https://machinelearningmastery.com/xgboost-for-regression/#:~:text=XGBoost%20is%20an%20efficient%20implementation,a%20prediction%20on%20new%20data

Carrillo, G. (2020, January 29). Predicting airbnb prices with machine learning and location data. Medium. https://towardsdatascience.com/predicting-airbnb-prices-with-machine-learning-and-location-data-5c1e033d0a5a#788c

Data.gov.sg. (2017, January 9). Tourist attractions. https://data.gov.sg/dataset/tourist-attractions

Data.gov.sg. (2019, October 9). LTA MRT station exit. https://data.gov.sg/dataset/lta-mrt-station-exit

Deepanshi. (2021, May 25). All you need to know about your first Machine Learning model – Linear Regression. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/#:~:text=In%20the%20most%20simple%20words,the%20dependent%20and%20independent%20variable

Gibbs, Chris & Guttentag, Daniel & Gretzel, Ulrike & Yao, Lan & Morton, Jym. (2017). Use of dynamic pricing strategies by Airbnb hosts. International Journal of Contemporary Hospitality Management. 30. 00-00. 10.1108/IJCHM-09-2016-0540.

Great Learning Team. (2023, January 12). Understanding of Lasso Regression. Great Learning Blog. https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/

Jin, X., Han, J. (2011). K-Means Clustering. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_425

Kalehbasti, P. R., Nikolenko, L., Rezaei, H. (2021). Airbnb Price Prediction Using Machine Learning and Sentiment Analysis. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2021. Lecture Notes in Computer Science(), vol 12844. Springer, Cham. https://doi-org.libproxy.smu.edu.sg/10.1007/978-3-030-84060-0_11

ks4s. (2019). Airbnb Singapore listing. Kaggle. https://www.kaggle.com/datasets/sarvasaga/airbnb-singapore-listing

Lim, V. (2021). Mall-Coordinates-Web-Scraper [Computer software]. GitHub. https://github.com/ValaryLim/Mall-Coordinates-Web-Scraper

Loukas, S. (2020, May 28). Everything You Need to Know About Min-Max Normalization in Python. Toward Data Science. https://towardsdatascience.com/everything-you-need-to-know-about-min-max-normalization-in-python-b79592732b79

Mohan, P. (2021, December 29). Ridge, Lasso & Elastic Net Regression. Medium. https://blog.devgenius.io/ridge-lasso-elastic-net-regression-2ea752186e51

Pandian, S. (2022, August 24). K-fold cross validation technique and its essentials. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/

Prasad, A. (2021, August 8). Regression trees: Decision tree for regression: Machine Learning. Medium. https://medium.com/analytics-vidhya/regression-trees-decision-tree-for-regression-machine-learning-e4d7525d8047

Sharma, M. (2020, March 21). Grid search for hyperparameter tuning. Medium. https://towardsdatascience.com/grid-search-for-hyperparameter-tuning-9f63945e8fec

Singh, A. (2023, March 13). KNN algorithm: Introduction to K-nearest neighbors algorithm for regression. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/