Predicting Prices of the Property Listings using Supervised Machine Learning Techniques

Ritika Bhatia
Department of Computer Science,
Carleton University,
K1S5B6,Ottawa,ON,Canada,
Email:ritikabhatia@cmail.carleton.ca

Herna Viktor
Department of Computer Science,
University of Ottawa,
Ottawa,ON,K1N6N5,
Email:hviktor@uottawa.ca

***Abstract*— The major aspect of the hospitality industry includes the facilities it provides and its pricing offered to the consumers. This project focuses on the application of various machine learning algorithms, in the domain of online operational hotel business models specially focusing on the problem to predict the prices for the property listing. Airbnb is one of the largest platform that contributes in the online hotel industry and provides opportunity to hosts as well as customers to post and rent the properties. In this project, a prototype has been built using several machine learning algorithms like Decision Tree, XGBoost, Linear Regression, Dummy Regressor, K Nearest Neighbor. that utilizes several features of the properties to get the estimate prices of the properties. Various features and their combinations were investigated to inspect the accuracy and the error in the predictions. Statistical testing of different algorithms has been performed to validate the results of the prediction as well several features were taken into consideration to visualize the pattern of the information provided.**

***Keywords*—Business Intelligence, Price Prediction Model, Regression, Ensembling, Tree Regressor, Statistical Testing**

I. INTRODUCTION

Since the establishment of Airbnb, it has become the epitome of the shared economy[1], and has gained popularity among the travellers. Trust and reliability are the major factors that establish the ground for peer to peer business models. Airbnb listings not only provides unique revelations into the essential qualities of the city and its surroundings but also provides visitors to contribute towards the “shared economy”[1]. Shared economy is the group that makes use of internet for renting and getting the services. Airbnb is one of the platform that runs on the basis of shared economy and enables its users to communicate, rent, book, or host properties or similar services via internet.

The characteristics that make Airbnb a favorite choice for travellers is the opportunity to take advantage of the local authenticity, affordable pricing rather than hotels, and privacy Since its foundation, the user base of this website is rapidly increasing and the amount of services provided to the users have Concerns[2]. With the large number of services provided, like setting the rate by the potential hosts himself, also improving the customized searching for the property, Airbnb has successfully set its base. also been improved over couple of years and therefore, increasing the revenue of the website.

This project focuses on the prediction of accurate pricing of the property listings that could help people to explore the real benefits of using Airbnb. Considering the scenario that a visitor or a person has to shift or relocate, but want to keep its original flat in its own native city. So the person prefers to rent his or her property to some other person or visitor. So instead of random guessing of the prices, machine learning helps to evaluate the price of property by taking into consideration, the list of near and similar property listings and helps the host or user to find the real worth of their property[3], so that user gets the initial idea while thinking of renting his or her space. The extra advantages that price prediction brings in the business model of the shared economy includes increase in revenue, which is the key intention of any business model. Second is the customer retention and satisfaction. More benefits and profit shares, the customer gets, more is the customer retention rate, therefore enhancing the networking of the business model. Third is the affordable pricing range customized according to the customers. The property features are not just limited its physical characteristics but also includes the services provided and the amenities available inside and nearby the property listings.

Proposed Solution: With the help of machine learning, and exploiting the features of the dataset, and the understanding the ingredients that are impacting the pricing of the different types of property including condo, big or small houses, etc., the proposed model will able to yield effective pricing of the property listings. The model will be able to gain insights from the ground real-world dataset and helps to yield the results that can be proved promising for the new hosts.

II. CASE STUDY

Airbnb is American company acting as an intermediator between customers and the potential hosts. The Airbnb can be accessed by any website or by using the android application.

In many cases the price prediction, of Airbnb price prediction is always missed in the literature and previous researches. The reason for choosing the Berlin city is due to the increased number of visitors as well as the increase in the number of people interested to rent the properties. This could lead to increase in the number of potential hosts as well as number of commuters.

For any business model to operate, the revenue is the key concern. Websites like Airbnb earn their profits by acting as a medium between different users. The application of machine learning techniques will not only yield results helpful for the customers but also for the revenue generation of the firm. For instance, in [4] the authors used regression techniques involving panels to visualize and analyze the impact of hotel industry on the growth of economy of the country.

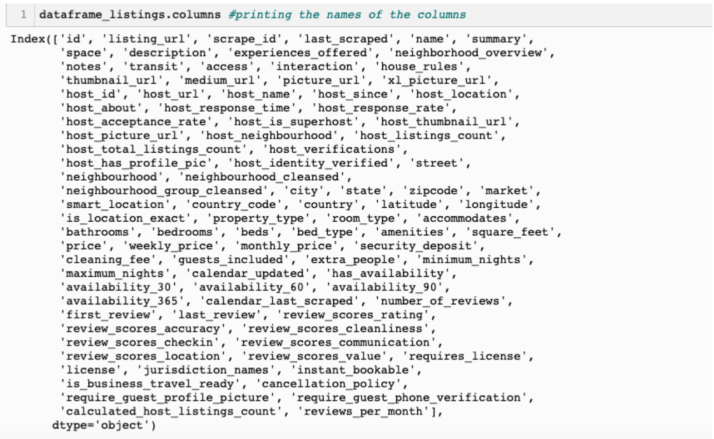
For the Berlin Airbnb dataset, there are total of 96 characteristics available for the dataset that need to be explored. In the research paper[1], the authors worked on the similar type of problem, to predict the prices of San Francisco Airbnb Data by using five features and performed feature scaling and then applied Support Vector Machine for the prediction purposes. In [5], the authors discussed the importance of the effective pricing and how it could impact the customer base of the website. In [5], they used multiclass clustering to find the associated patterns within the dataset. In [6], the authors considered several features like availability, amenities, reviews, description, etc. in order to predict the accurate price listings of the San Francisco Airbnb dataset. In [7], the authors applied several machine learning algorithms like XGBOOST, Neural networks, ridge regression in order to obtain the useful results. These studies sets the baseline to calculate prices using machine learning approaches.

III. EXPERIMENTAL SETUP AND EVALUATION

A. The Dataset

The dataset includes the real time data of Airbnb website and it was scraped in the year 2018. The file used for this prediction project is named listings_summary. This data was created by Murray Cox and can be found on[8]. The dataset consists of 22500 rows and 96 columns initially. The raw dataset consists of many unnecessary instances and variables and was refined, so that machine learning models can work extensively with data.

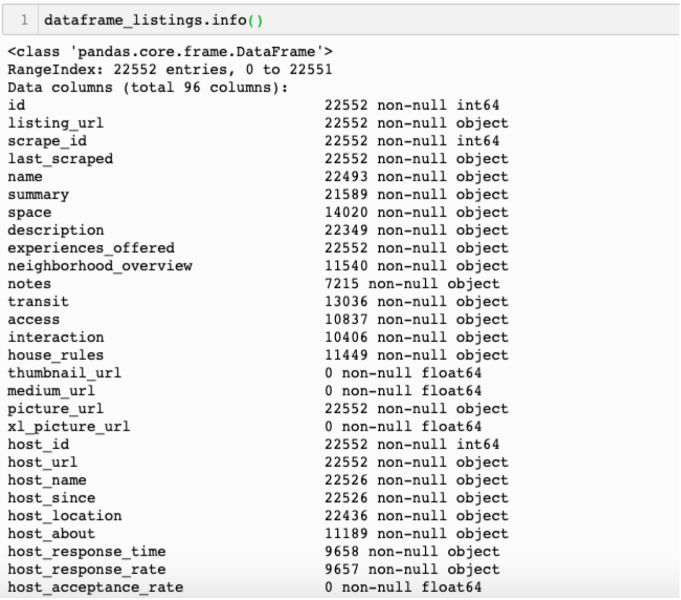
The dataset consists of the attributes column id, listing url, scrape id, last scraped, name, summary, space, description, experiences offered, neighborhood overview, notes, access, interaction, url, host id, reviews score, reviews accuracy, etc. Figure 1 describes the total number of columns present in the dataset before any feature synthesis. Figure 2 describes the attributes and the types of values present inside the attributes. The dataset is loaded using the pandas library that considers the csv file as a dataframe and could help in multiple manipulations that are required to be performed simultaneously for many data values present inside the table. The output of the pandas framework is in tabular form that helps to analyze all the feature values easily and at a very fast pace.



```
dataframe_listings.columns #printing the names of the columns
```

```
Index(['id', 'listing_url', 'scrape_id', 'last_scraped', 'name', 'summary',
       'space', 'description', 'experiences_offered', 'neighborhood_overview',
       'notes', 'transit', 'access', 'interaction', 'house_rules',
       'thumbnail_url', 'medium_url', 'picture_url', 'xl_picture_url',
       'host_id', 'host_url', 'host_name', 'host_since', 'host_location',
       'host_about', 'host_response_time', 'host_response_rate',
       'host_acceptance_rate', 'host_is_superhost', 'host_thumbnail_url',
       'host_picture_url', 'host_neighbourhood', 'host_listings_count',
       'host_total_listings_count', 'host_verifications',
       'host_has_profile_pic', 'host_identity_verified', 'street',
       'neighbourhood', 'neighbourhood_cleansed',
       'neighbourhood_group_cleansed', 'city', 'state', 'zipcode', 'market',
       'smart_location', 'country_code', 'country', 'latitude', 'longitude',
       'is_location_exact', 'property_type', 'room_type', 'accommodates',
       'bathrooms', 'bedrooms', 'beds', 'bed_type', 'amenities', 'square_feet',
       'price', 'weekly_price', 'monthly_price', 'security_deposit',
       'cleaning_fee', 'guests_included', 'extra_people', 'minimum_nights',
       'maximum_nights', 'calendar_updated', 'has_availability',
       'availability_30', 'availability_60', 'availability_90',
       'availability_365', 'calendar_last_scraped', 'number_of_reviews',
       'first_review', 'last_review', 'review_scores_rating',
       'review_scores_accuracy', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value', 'requires_license',
       'license', 'jurisdiction_names', 'instant_bookable',
       'is_business_travel_ready', 'cancellation_policy',
       'require_guest_profile_picture', 'require_guest_phone_verification',
       'calculated_host_listings_count', 'reviews_per_month'],
      dtype='object')
```

Figure 1: Displays the names of different attributes



```
dataframe_listings.info()
```

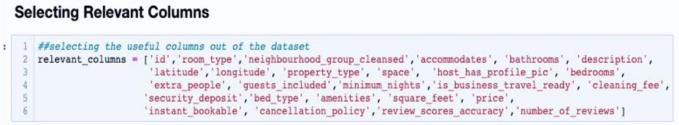
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22552 entries, 0 to 22551
Data columns (total 96 columns):
id                                  22552 non-null int64
listing_url                         22552 non-null object
scrape_id                           22552 non-null int64
last_scraped                        22552 non-null object
name                                22493 non-null object
summary                             21589 non-null object
space                               14020 non-null object
description                         22349 non-null object
experiences_offered                 22552 non-null object
neighborhood_overview               11540 non-null object
notes                               7215 non-null object
transit                             13036 non-null object
access                              10837 non-null object
interaction                         10406 non-null object
house_rules                         11449 non-null object
thumbnail_url                       0 non-null float64
medium_url                          0 non-null float64
picture_url                         22552 non-null object
xl_picture_url                      0 non-null float64
host_id                             22552 non-null int64
host_url                            22552 non-null object
host_name                           22526 non-null object
host_since                          22526 non-null object
host_location                       22436 non-null object
host_about                          11189 non-null object
host_response_time                  9658 non-null object
host_response_rate                  9657 non-null object
host_acceptance_rate                0 non-null float64
```

Figure 2: Displays the attributes and datatypes.

B. Feature Selection

Feature Selection can be referred to as the selection of the attributes or columns that could best represent the dataset without containing the inconsistent and redundant values. The feature selection plays an important role while dealing with large number of features. The main goals of feature selection is to select the relevant attributes that are useful in defining the predictions. The feature selection leads to the reduced number of features, therefore the complexity of the model is reduced and also the chances of overfitting. It enhances the learning capability of the model. There are various inbuilt functions that can be used for detecting the relevant features like Chi-square test, Pearson Correlation Test, and the output predicts the most suitable features.

In this model, various features and their combinations were tried and tested and hence, 25 out of 96 columns were find out to be relevant for the dataset. Figure 4 describes the selection of the relevant columns from the dataset. The relevant columns include id, room_type, neighbor_hood_group_closed, accommodates, description, latitude, longitude, is travel friendly, cleaning fees, pricing, security deposits, cancellation policy, number of reviews, instant bookable, amenities, bed type, host profile pic.



Selecting Relevant Columns

```
##selecting the useful columns out of the dataset
relevant_columns = ['id','room_type','neighbourhood_group_cleansed','accommodates', 'bathrooms', 'description',
                    'latitude','longitude', 'property_type', 'space',  'host_has_profile_pic', 'bedrooms',
                    'extra_people', 'guests_included','minimum_nights','is_business_travel_ready', 'cleaning_fee',
                   'security_deposit','bed_type', 'amenities', 'square_feet', 'price',
                   'instant_bookable', 'cancellation_policy','review_scores_accuracy','number_of_reviews']
```

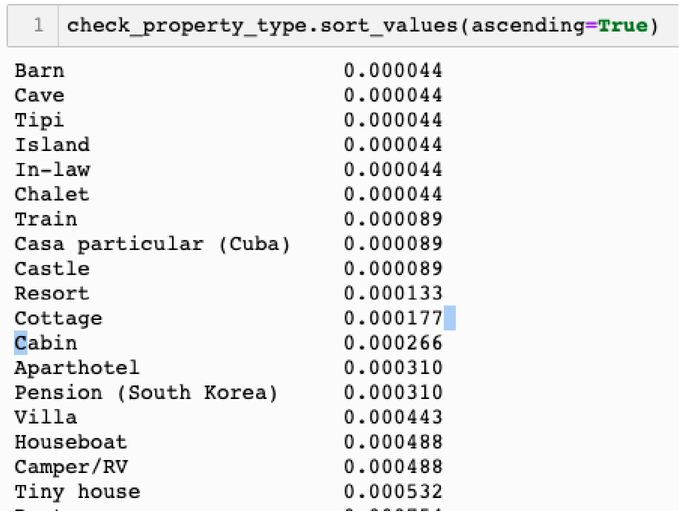
Figure 4: Showing the selection of the relevant columns

C. Data Cleaning and Preprocessing

The performance of the model directly or indirectly depends on the procedure of data cleaning. The raw data contains many outliers, inconsistent values and improper format that could mislead the results. Inconsistent and redundant values may lead to untrue conclusions and could be hard to map to the real time scenario. In any business model, the decisions are made on the reliable data which can be trusted and could not deflect the interests of the company. Unreliable datasets can prove costly to the entire firm as well as impacts the trust of its customers. Therefore, before any model implementation data cleaning process is an inclusive and intensive step. The merits of data cleaning include the consistent data with removal of false values or the removal of the values that could not define the dataset.

Regarding this dataset, attributes containing unnecessary values were cleaned and transformed for further analysis.

Beginning with the attribute property type, first its value were checked and normalize to see what type of values are contained within this attribute. The property type mentioned in the dataset are apartment, condominium, hotels, guest house, townhouse, hostel, Villa, Boat, Bungalow, Barn, Cave, Chalet, etc.



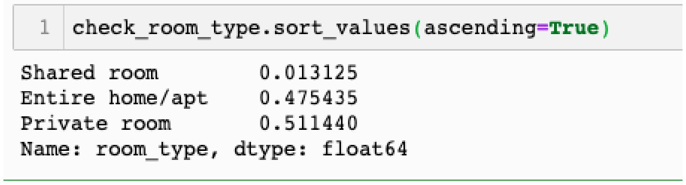
```
check_property_type.sort_values(ascending=True)
```

```
Barn                        0.000044
Cave                        0.000044
Tipi                        0.000044
Island                      0.000044
In-law                      0.000044
Chalet                      0.000044
Train                       0.000089
Casa particular (Cuba)      0.000089
Castle                      0.000089
Resort                      0.000133
Cottage                     0.000177
Cabin                       0.000266
Aparthotel                  0.000310
Pension (South Korea)       0.000310
Villa                       0.000443
Houseboat                   0.000488
Camper/RV                   0.000488
Tiny house                  0.000532
```

Figure 5: Showing the type of the property types present in the dataset

Figure 5 explains the type of properties available with their normalized values.

Next attribute that is inspected is the room type. The host on the Airbnb website has the option to provide information about the type of room, it has to offer to its customers. The potential type of room available for the customers are Shared room, entire room or apartment. Figure 6 shows the output of the type of rooms that are available for renting.

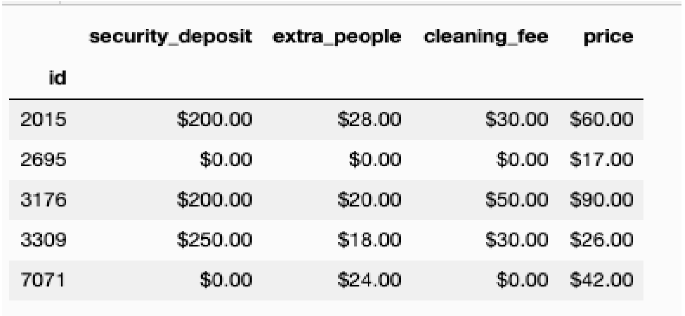


```
check_room_type.sort_values(ascending=True)
```

```
Shared room        0.013125
Entire home/apt    0.475435
Private room       0.511440
Name: room_type, dtype: float64
```

Figure 6: Types of rooms available

The next feature that is accounted for this dataset is price. The actual price of the property depends on many factors. Hence, all the attributes indicating variable amount of prices were taken into consideration. The attributes of the pricing includes the price of the security deposit that people have to pay initially as a token amount, price while accommodating extra people, the cleaning fees, and the actual fees that constitutes the price of the property listing. Figure 7 describes the type of pricing included in the dataset.

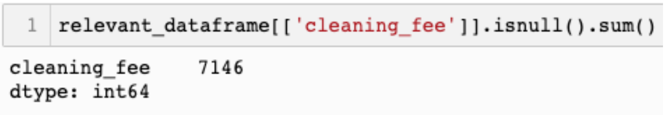


	security_deposit	extra_people	cleaning_fee	price
id				
2015	$200.00	$28.00	$30.00	$60.00
2695	$0.00	$0.00	$0.00	$17.00
3176	$200.00	$20.00	$50.00	$90.00
3309	$250.00	$18.00	$30.00	$26.00
7071	$0.00	$24.00	$0.00	$42.00

Figure.7: Showing the categories of different pricings

(i) Removal of Null Values

The real world data includes inconsistent and heterogenous data. There might be scenarios where there are no values present for any datatype or the values are missing, and they are considered as null values. The null values if fed to the machine learning model, may not be able to predict the desired results. Therefore the null and missing values should be handled and removed accordingly. In this project, null values are checked in different attributes. Figure 8 shows the checking of null values present inside the attribute cleaning fee. The output shows the 7146 records with missing values of cleaning fee.

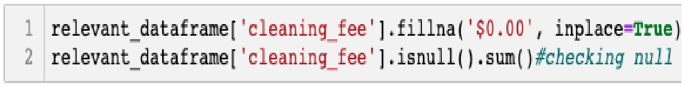


```
relevant_dataframe[['cleaning_fee']].isnull().sum()
```

```
cleaning_fee     7146
dtype: int64
```

Figure 8: Checking of null values in the attribute cleaning fee

The null values in the cleaning fee is replaced by the integer 0, assuming that the missing values for this numerical feature corresponds to no cleaning fees. Figure 9 shows the imputation of null values in cleaning fee attribute with 0. After the imputation, rechecking of null values returns 0.



```
relevant_dataframe['cleaning_fee'].fillna('$0.00', inplace=True)
relevant_dataframe['cleaning_fee'].isnull().sum()#checking null
```

Figure 9: Displaying the imputation of null values with 0

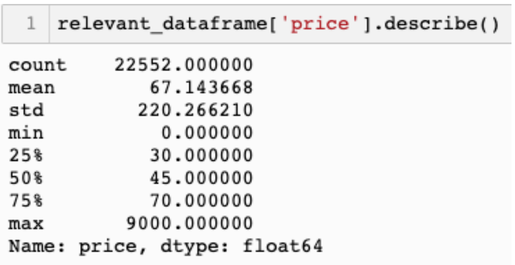
Following the similar procedure all the attributes related to pricing, were inspected for the null values and imputed with the value 0.

(ii) Cleaning Numeric Values

The numeric values contained in the pricing attribute must be cleaned in order to be utilized for calculations. The attributes contained '$' sign in it which could obstruct numerical manipulations. Therefore, the numeric values are cleaned and '$' sign is stripped from the values.

(iii) Checking the Ranging of Price Attribute

The price attribute was examined to see what is the most dominant pricing range. This range was inspected using boxplot so that the values surrounding the mean were taken into consideration and the outliers that could impact the result will be dropped. Figure 11 displays the characteristics of the price attribute. Figure 12 displays the variance in the price attribute and how the values in the attribute are spread and while moving away from mean, there are very less number of price rangings therefore, not giving any useful information. In this scenario, the values upto 600 are considered for the prediction model and the values above 600 are assumed as outliers and dropped from the dataset.

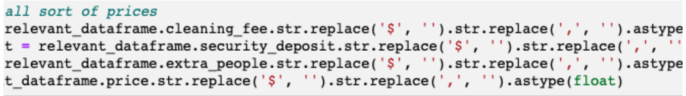


```
relevant_dataframe['price'].describe()

count    22552.000000
mean        67.143668
std        220.266210
min          0.000000
25%         30.000000
50%         45.000000
75%         70.000000
max       9000.000000
Name: price, dtype: float64
```

Figure 11: Displaying the information about price attribute

(iv) Dropping Attributes with most null values and dropping rows with maximum missing values

The attributes present in the dataset space and square feet has approximately 9k and 22k missing values. Therefore, imputation could lead to faulty results. In this scenario, these columns were dropped from the dataset. On the other hand, the rows which contain maximum amount of missing values of the attributes bedrooms, bathrooms, accuracy score, etc. are also removed using dropna python function. The reason behind these removal is that more the null values, more are the chances of the inaccurate results. Another categorical attribute, naming 'host_has_profile_pic' is taken into consideration and all its missing values are replaced with the keyword false, assuming that the person who does not have any value of profile picture might not have posted his or her information on the website.



```
all sort of prices
relevant_dataframe.cleaning_fee.str.replace('$', '').str.replace(',', '').astype
t = relevant_dataframe.security_deposit.str.replace('$', '').str.replace(',', ''
relevant_dataframe.extra_people.str.replace('$', '').str.replace(',', '').astype
t_dataframe.price.str.replace('$', '').str.replace(',', '').astype(float)
```

Figure 10: Showcases the cleaning of numeric values

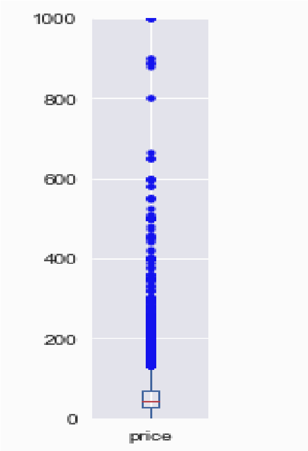


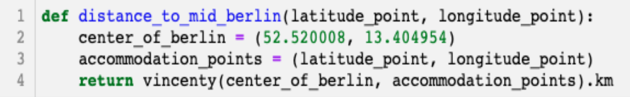
Figure 12: Boxplot of the price range where mean value is between 0 and 200 and most of the values lies in the range upto 600

D. Feature Engineering

Along with the feature selection and data cleaning and preprocessing, feature engineering is also the main aspect of any machine learning algorithm. It is basically done to build the explanatory features[9] and can significantly enhance the performance of predictive models. The process of feature engineering removes the complex details of the dataset and these complex details can be expressed in simpler form, that can be digested and learned by the machine learning algorithm easily.

(i) Constructing feature 'Distance' by using Latitude and Longitude values

The feature 'distance from the center of the city' was constructed by calculating the distance from the property's location latitude and longitude values. The library geopy, python library that induces geography was used to construct this feature and the function used was vincenty. The vincenty function yields more accurate real time distance values rather than any other function like great circle. The function was built in which latitude and longitude values of the center of the city was passed and also the latitude and longitude values of the property listing was passed and the actual distance was obtained as output. Figure 13 shows the glimpse of the function using vincenty to record the distances.



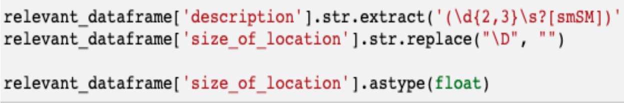
```python
def distance_to_mid_berlin(latitude_point, longitude_point):
    center_of_berlin = (52.520008, 13.404954)
    accommodation_points = (latitude_point, longitude_point)
    return vincenty(center_of_berlin, accommodation_points).km
```

Figure 13: Function displaying the calculation of distance

(ii) Constructing Feature Size by exploiting the attribute Description

The description attribute in the dataset is the textual attribute which contains characteristics of the property listings. maximum amount

Initially, the attribute size was dropped from the dataset due to large number of null values. Therefore, the description attribute is explored to obtain the values of the size attribute. The procedure involves in the extraction of the size attribute includes the stripping of the integers present in the description attribute in the first initial phrasings and loading those values in new attribute named ‘size’. Figure 14 shows the function of extracting the numerical values in the description column.



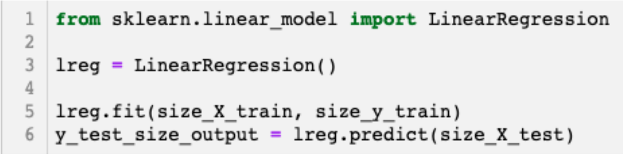
```python
relevant_dataframe['description'].str.extract('(\d{2,3}\s?[smSM])'
relevant_dataframe['size_of_location'].str.replace("\D", "")

relevant_dataframe['size_of_location'].astype(float)
```

Figure 14: Extraction of integer values from the description attribute to build the size attribute

(iii) Predicting missing values for size attribute using Regression

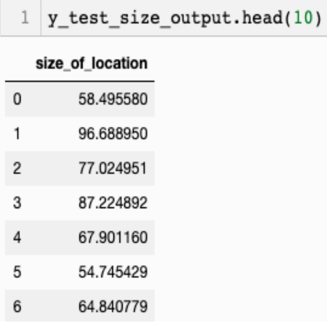
The earlier extraction function was able to strip off the values from the description, but there are still some values, approximately 51% left that are interpreted as missing values. For finding out the missing values, the technique utilized was Linear regression technique. The textual features were dropped from the training dataset and fed into the model and used for the prediction. The approach for direct imputation of values might not be useful in this scenario because it is quite tedious task to impute values that could match the ground reality of the dataset. Therefore, regression was implemented in this scenario.

Figure 15 showcases the implementation of the regression function to calculate the values of size of the location. Figure 16 showcases the output of the size attribute obtained after the application of the function.



```python
from sklearn.linear_model import LinearRegression

lreg = LinearRegression()

lreg.fit(size_X_train, size_y_train)
y_test_size_output = lreg.predict(size_X_test)
```

Figure 15: Linear Regression for predicting the values of size attribute



```
y_test_size_output.head(10)

   size_of_location
0         58.495580
1         96.688950
2         77.024951
3         87.224892
4         67.901160
5         54.745429
6         64.840779
```

Figure 16: Results of size attribute after prediction

The output dataframe is then concatenated back to the original dataset to build as a whole.

(iv) Constructing Categorical Features from attribute Amenities

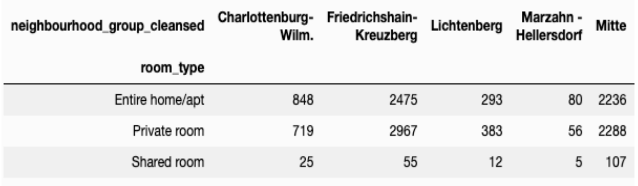
Amenities attribute in the dataset contains the names of the facilities offered by the hosts in their properties. The basic amenities included Wifi, TV, Cable, Kitchen, Gym, Heating, Elevator, Smoking allowed, Breakfast, Street Parking, Shampoo, Pet Friendly, Hot water, iron, etc. The facilities offered were analyzed and the most suitable facilities were isolated and special columns were added to existing dataframe that could indicate the presence of those utilities. The facilities selected for the construction of the new columns were laptop Friendly Space, TV, Smoking allowed, Host greets you, Family kid friendly.

E. DETAILED DATA ANALYSIS

In this section, many attributes corresponding to the prediction of the pricing and their impact on the pricing is investigated by doing exploratory data analysis.

E.1. Linking of Room Type and its Pricing

Figure 17 presents the detailed analysis of the impact of the type of the group and the type of the room and its interconnection with the pricing. With the type of the room and depending upon its nearby locality, the pricing of the room changes. It is observed that the several areas like Friedrichshain having locality Kreuzberg has more pricing than the other areas.



neighbourhood_group_cleansed	Charlottenburg-Wilm.	Friedrichshain-Kreuzberg	Lichtenberg	Marzahn - Hellersdorf	Mitte
room_type					
Entire home/apt	848	2475	293	80	2236
Private room	719	2967	383	56	2288
Shared room	25	55	12	5	107

Figure 17: Linking of the rooms and the closed neighborhood

Figure 18 describes the price range of different types of houses along with the closed neighborhood groups using stacked bar plot.

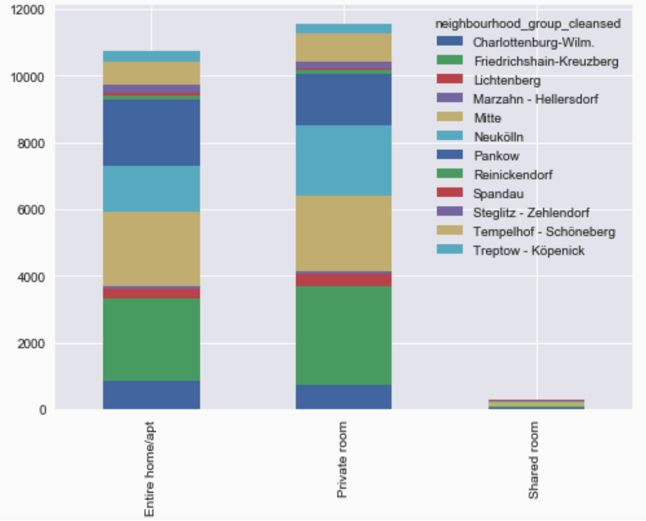


Figure 18 : Stacked plot of locations and neighboring groups

E.2. Linking of Latitude and Longitude

The points of latitude and longitude are plotted to visualize the spread of the property listings within the city. It helps to visualize the area covered by the property listings and the analyze the number of properties located within the city and its central points and the points which are located far away from the center of the city.

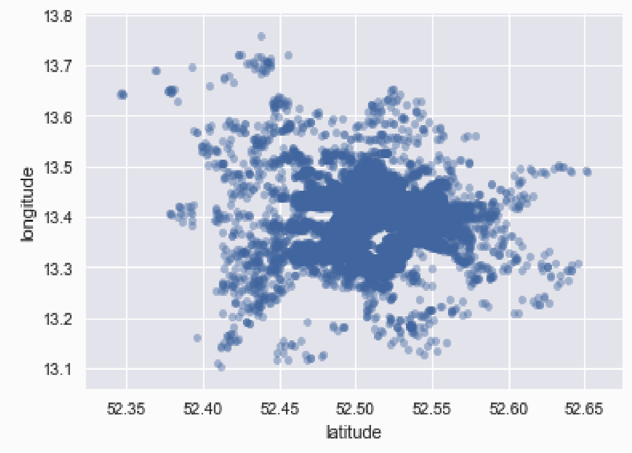


Figure 19: spread of the property points within the city

E.3. Linking of Prices with the number of Bedrooms

The prices of the property listings vary with the change in number of bedrooms and also the distances of the neighborhoods. The interesting part to gain insight is that the combination of the two variables impacting the price of the property. The number of bedrooms matters when it is chosen to proximity locations and therefore, the pricing of the properties fluctuate. Figure 20 represents the change of the median pricing with respect to the number of bedrooms and closed neighborhoods.

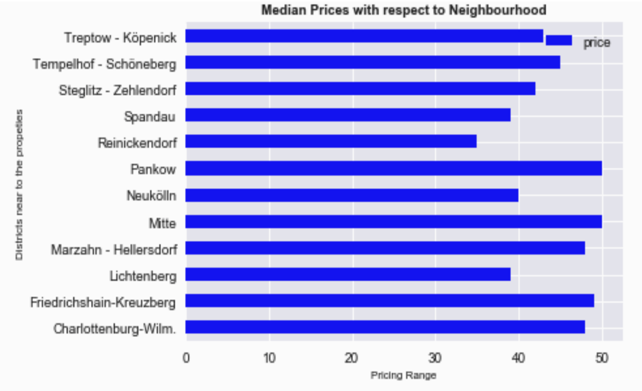


Figure 20: Linking of price rangings when compared to the neighborhoods

E.4. Linking of Prices with the Cancellation Policy

The prices are correlated to the type of cancellation policy it has. It is observed that the type of cancellation that has super_strict_60 has more price ranging whereas the property listings that have flexible and moderate cancellation policies have lower price rangings. The cancellation policy with grace period of 14 days has moderate range of pricing. Figure 21 indicates the scenario of the cancellation policies and their price rangings.

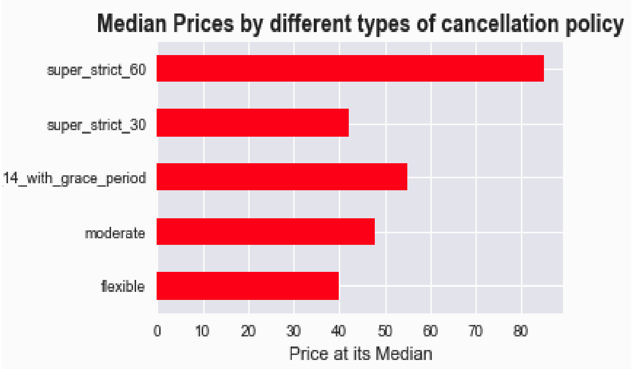
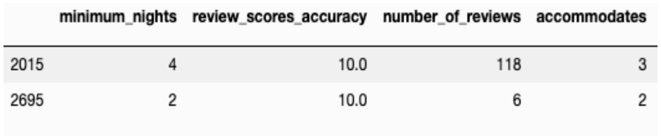


Figure 21: Correlation between price rangings and cancellation policy

F. DATA MODELING AND TRANSFORMATION

Machine learning algorithms are not efficient to handle the raw data. Therefore, it is necessary to transform the features into specific type. For this data, there are several features that need to be converted into categorical and numerical features specifically. For instance, all the numeric features needs to be transformed into the proper integer datatypes like float, etc for the proper mathematical manipulations.

The features which can be described using the categories need to be transformed into the categorical type. The type converted should be compatible with the machine learning algorithm and can be converted using the inbuilt convert functions.



	minimum_nights	review_scores_accuracy	number_of_reviews	accommodates
2015	4	10.0	118	3
2695	2	10.0	6	2

Figure 22: Glimpse of Features after transformations

For the accurate prediction and performance for machine learning algorithms, the data needs to be in proper format. Data modeling is necessary so that all the features can be represented in similar type of dimensions. For this the procedure of the normalization is carried out. The normalization is based on the mathematical procedure:

$$X(new) = X - X(min) / X(max) - X(min) \quad [10]$$

It represents the features in the same dimensions so that one feature does not overpower or impact the dependency of the other feature.

G. Final Feature Selection

The table represents the final list of the features that were obtained using the feature engineering and the feature selection. The class and Price Class are the dependent variable that is supposed to be predicted by the application of several machine learning algorithms. The number of rows were reduced to 18.5k.

S.No	Feature	Type
1	Minimum nights	Int 64
2	Review scores accuracy	Float 64
3	Number of reviews	Int 64
4	Accommodates	Float 64
5	Bathrooms	Float 64
6	Bedrooms	Float64
7	Cleaning Fees	Float64
8	Distance from center	Float 64
9	Extra People	Float 64
10	Guests Included	Int 64
11	Security Deposit	Float 64
12	Size of Location	Float 64
13	Laptop_friendly_workspace	Boolean
14	TV	Boolean
15	Family kid friendly	Boolean
16	Host greets you	Boolean
17	Smoking allowed	Boolean
18	Entire room	Unit-8
19	Private Room	Unit-8
20	Shared Room	Unit-8
21	Host has profile pic t	Unit-8
22	Host has profile pic f	Unit-8
23	Business ready	Unit-8
24	Airbed travel	Unit-8
25	Bed type couch	Unit-8
26	Bed type futon	Unit-8
27	Pull out sofa	Unit-8
28	Real bed	Unit-8
29	Instant bookable false	Unit-8
30	Instant bookable true	Unit-8
31	Flexible cancel policy	Unit-8
32	Moderate cancel policy	Unit-8
33	Strict 14 cancel policy	Unit-8
34	Super 30 cancel policy	Unit-8
35	Super strict 60 cancel policy	Unit-8

Table.1: Showing the extracted features in Airbnb Berlin Dataset

H. MODEL APPLICATION

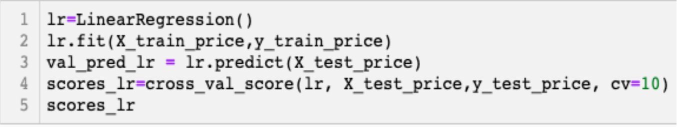
The different types of the model are being evaluated on this dataset. The dataset is split in 80-20 ratio which means 80% of the data remains in the training set, with which the algorithms will learn a mapping towards the target value and 20% of the dataset remains in the test data, means that the model will be evaluated by checking on that 20% of the data. The implementation of the model ranges from the nearest neighbor, tree learners, rule based learners, ensembling methods, and linear classifier. The cross validation technique is also used in order to enhance the prediction power of the regressor algorithm. All the algorithms, use cross validation value 10 for the implementation purpose.

H.1. Linear Classifier

The Linear regression is implemented on the Berlin Airbnb dataset. The linear regression learns the values by mapping a linear function from input values to the target value. The cross validation value used in 10 in his scenario. Simple Linear Regression basically estimates the coefficients of the equation x1 and x2 which predicts the target value(y)[11]. This can be expressed by the mathematical notation

$$Target(y)=x1+c.x2$$

Figure 22 shows the application of the Linear Regression Model onto the Berlin Airbnb dataset.



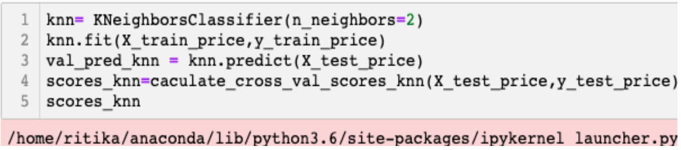
```
lr=LinearRegression()
lr.fit(X_train_price,y_train_price)
val_pred_lr = lr.predict(X_test_price)
scores_lr=cross_val_score(lr, X_test_price,y_test_price, cv=10)
scores_lr
```

Figure 23: showing the implementation of linear regression on the dataset.

H.2. Nearest Neighbor Regressor

K nearest neighbor is one of the most essential and simplest supervised learning method and becomes one of the finest choices for the model application[12]. K nearest neighbor was built to perform discriminant analysis when probability densities of parameters are unknown. It basically operates on the calculation of the eucledian distances among the data points which considers the squares of the difference between the two points.

The K nearest neighbor regressor is applied on the Berlin Airbnb datatset. The different values of the K is tried and tested on the models but the most suitable value for this dataset turns out to be 2. Figure 24 describes the application of K nearest neighbor on the Berlin dataset.



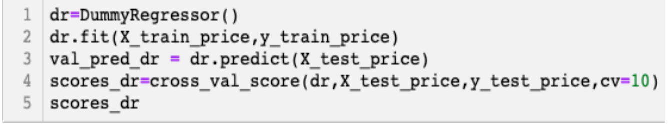
```
knn= KNeighborsClassifier(n_neighbors=2)
knn.fit(X_train_price,y_train_price)
val_pred_knn = knn.predict(X_test_price)
scores_knn=caculate_cross_val_scores_knn(X_test_price,y_test_price)
scores_knn
```

```
/home/ritika/anaconda/lib/python3.6/site-packages/ipykernel launcher.py
```

Figure 24: Describes the application of Distance Based Learning Algorithm on the dataset

H.3. Rule Based Learners

Dummy Regressor is used for the rule based learner. Dummy Regressor is the type of algorithm that does not yield any insights from the data, but rather uses the simple rules to predict the values. It basically works on random guessing and probably finds difficult to learn values in regression algorithms. It acts as the baseline for other classifiers and regressors and can operate in cases where the values are most frequent, stratified, uniform or constant. [13]. The cross validation value used in 10 in this case also. Figure 25 showcases the application of Dummy Regressor on the Berlin Airbnb Dataset.

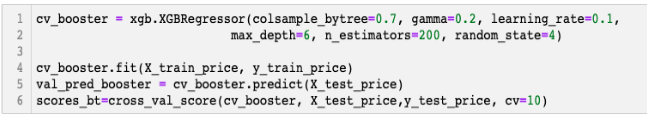


```
dr=DummyRegressor()
dr.fit(X_train_price,y_train_price)
val_pred_dr = dr.predict(X_test_price)
scores_dr=cross_val_score(dr,X_test_price,y_test_price,cv=10)
scores_dr
```

Figure 25 showcasing the implementation of Dummy regressor

H.4. Ensembling Method

The ensembling method used is XGBoost. It is one of the powerful algorithms that converts weak learners into strong learners. Several parameters were tried and tested for the XGBoost and Grid Search CV was applied for the hyperparameter tuning, and the values for the number of estimators were chosen was 200 with the learning rate of 0.1. The learning rate was kept low so that the model could not overfit the data. Figure 26 showcases the application of XGBoost algorithm on the Airbnb Dataset.

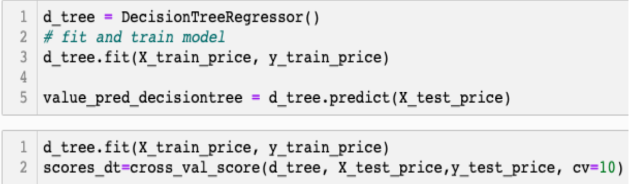


```
cv_booster = xgb.XGBRegressor(colsample_bytree=0.7, gamma=0.2, learning_rate=0.1,
                              max_depth=6, n_estimators=200, random_state=4)

cv_booster.fit(X_train_price, y_train_price)
val_pred_booster = cv_booster.predict(X_test_price)
scores_bt=cross_val_score(cv_booster, X_test_price,y_test_price, cv=10)
```

Figure 26 Implementation of XGBoost

H.5. Tree Based Regressor

The tree based Regressor used here is Decision Tree Regressor. The decision tree regressor can be expressed as the recursive partition algorithm that consists of the root nodes, internal nodes, and leaf nodes[14]. The default values were chosen as hyperparameters for the application of the decision tree. The value of cross validation used here is 10. Figure 27 illustrates the implementation of decision tree regressor on the Berlin Airbnb Dataset.



```
d_tree = DecisionTreeRegressor()
# fit and train model
d_tree.fit(X_train_price, y_train_price)

value_pred_decisiontree = d_tree.predict(X_test_price)
```

```
d_tree.fit(X_train_price, y_train_price)
scores_dt=cross_val_score(d_tree, X_test_price,y_test_price, cv=10)
```

Figure 27: Illustration of application of decision tree regressor on Airbnb Dataset

H.6. Cross Validation Technique

Cross Validation is the techniques that is used for the validation of the model and reduce the bias of the models. It basically helps to understand or estimate the new data [15] by providing input to the model by reshuffling and splitting the dataset. It is basically a method of resampling that is used to validate the performance of machine learning algorithm on the limited amount of dataset. The basic value for cross validation is considered to be 10 in most of the cases.

I. MODEL EVALUATION

There are different metrics for testing the evaluation and performance of the models.

R Score: Based on such metric, R score was used to see the fit of the predicted values according to the algorithms. Higher the value of R score, more the model is able to explain the variability within the dataset. Figure 28 represents the value of r score corresponding to the different algorithms. The ensembling method XGboost was able to get the highest value of r score among all the algorithms applied and hence, the most suitable machine learning algorithm for this dataset.

RMSE: Another metric that was used for the evaluation of the model is root mean squared error. Its numeric quantity describes how much the expected value fluctuates from predicted value. Lower the rmse rate, better is the prediction performance of the model. Figure 29 explains the range of root mean square error calculated on different algorithms.

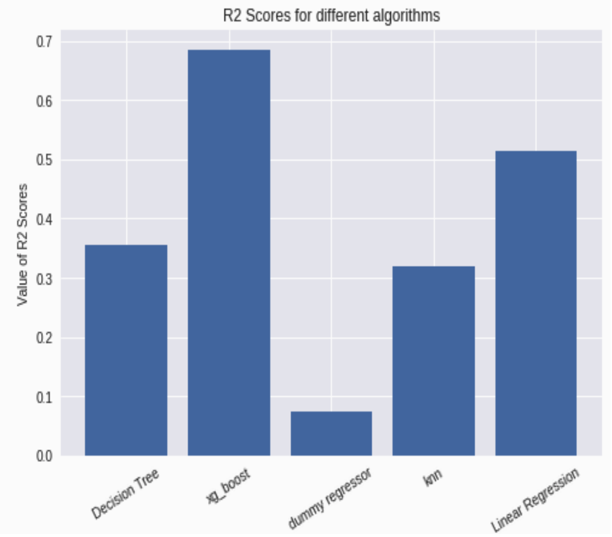


Figure 28: R scores of different algorithms, with the highest at XGboost and lowest rate at Dummy Regressor

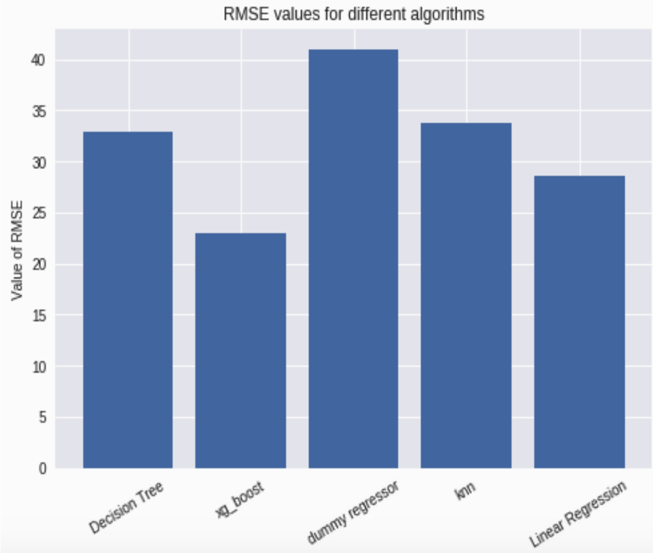


Figure 29: Rmse values for different algorithms with the highest point at dummy regressor and lowest at xg boost

J. STATISTICAL TESTING

The statistical testing is carried out to see whether the results of the different algorithms are statistically significant and the underlying distribution of the two result values vary. The statistical difference is being carried out on the combinations of the different algorithms. The methodology for carrying out the statistical testing used is Paired T-Test. For the statistical testing of all the algorithms with XGBoost, the results were statically significant from the other algorithms like linear regression, decision tree, and k nearest neighbors. In all of the cases, the p-value was below than the alpha set 0.05, therefore the results proved to be statistically significant.

For the statistical test of results of dummy regressor, only the results of decision tree were find to be statistically significant.

For the model, Linear regression, when performed t test with the results of Decision Tree yields the p-value 0.002, which means that the results are statistically significant. Another paired t test was performed on the results of KNN and dummy regressor which yielded the p-value less than 0.05, describing the results to be statistically significant.

IV. CONCLUSION AND FUTURE WORK

In the nutshell, different techniques and types of methodologies were applied on the Airbnb Berlin dataset to see the price prediction. There are many interesting factors that are unearthed during the implementation of the project. The combinations of the features and the variability of the features enhances and decreases the performance of the model. Therefore, the right feature selection should be performed correctly. Several features were tried and tested but, able to construct 35 features out of the 96 present in the original dataset. The dataset matches the standards of the ground reality, therefore the inference mapping of property listings with their was easier when features like space, size, number of bedrooms were taken into consideration.

The best metric suitable for this type of dataset is r score as the dataset fails to yield high accuracies. Therefore, to check the residual values with the predicted values is checked r score.

The most impacting features on the price range includes the availability and proximity to the center of the city, cleaning fee, the number of bedrooms, the number of review. So, the person looking for room in Berlin or a potential host trying to offer his or property on the site can consume these earlier predictions to get the idea of the actual rate scenario.

In the direction of the future research work, there can be many methodologies followed to improve the work. First can be the use of automated feature selection tools that could reduce the manual overhead while refining the features. Further, the application of Neural Networks and Deep Learning could help to yield better accuracy scores as compared to simple machine Learning algorithms. For the better prediction purposes, the actual customer reviews can be taken into consideration and can be transformed using several bag of words like word2vec and tfidf, to gain insights of the sentiments of the users and that could lead to the better prediction for the upcoming hosts and customers.

V. REFERENCES

[1]. Emily Tang, Kunal Sangani, “Neighborhood and Price Prediction for San Francisco Airbnb Listings” (2015)

[2] Daniel Gutentag, Stephen Smith, Luke Potwarka, “Why tourists choose Airbnb: A motivation based Segmentation Study”, Journal of Travel Research, Vol 57, Issue 3, pp. 342-359,2017.

[3] Predicting Prices using XGboost https://www.kaggle.com/brittabettendorf/predicting-prices-xgboost-feature-engineering

[4] Ki-Hong Choi, Joohyun Jung2, Suyeol Ryu3, Su-Do Kim4 and Seong-Min Yoon1” The relationship between Airbnb and the hotel revenue: In the case of Korea”, Indian Journal of Science and Technology, Vol 8, ISSN: 0974-6846, 2015

[5] Li, Y., Pan, Q., Yang, T., & Guo, L. (2016, July). Reasonable price recommendation on Airbnb using Multi-Scale clustering. In 2016 35th Chinese Control Conference (CCC) (pp. 7038-7041). IEEE.

[6] Choudhary, Paridhi, Aniket Jain, and Rahul Baijal. "Unravelling airbnb predicting price for new listing." arXiv preprint arXiv:1805.12101 (2018).

[7] Kalehbasti, Pouya Rezazadeh, Liubov Nikolenko, and Hoormazd Rezaei. "Airbnb Price Prediction Using Machine

Learning and Sentiment Analysis." arXiv preprint
arXiv:1907.12665 (2019).

[8]The link for the dataset
http://insideairbnb.com/get-the-data.html

[9] Why feature selection is important
https://www.rittmanmead.com/blog/2019/02/the-importance-of-feature-engineering-and-selection/

[10] How the values are normalized
https://www.statisticshowto.datasciencecentral.com/normalized/

[11] Simple Linear Regression
https://www.researchgate.net/publication/275018503_Simple_Linear_Regression

[12] Characteristics of K-Nearest Neighbor
http://scholarpedia.org/article/K-nearest_neighbor

[13] Working of Dummy Regressors
https://www.geeksforgeeks.org/ml-dummy-classifiers-using-sklearn/

[14] Tree based Regressors and classifiers
https://www.researchgate.net/publication/225237661_Decision_Trees

[15]Cross Validation Technique Sampling
https://machinelearningmastery.com/k-fold-cross-validation/