A PROJECT ON

"AIRBNB DATA ANALYSIS AND PRICE PREDICTION"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN BIG DATA ANALYSIS

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

Plot no. R/2, Market Yard Road,
Behind Hotel Fulera, Gultekdi,
Pune 411037.
MH-INDIA

SUBMITTED BY –
Abhay Paunikar (56904)
Kaivalya Pande (56547)

UNDER THE GUIDENCE OF –
Mr. Girish Gaikwad
Faculty Member,
Sunbeam Institute of Information Technology,
PUNE.

Certificate

This is to certify that the project work under the title ‘Airbnb Data Analysis and Price Prediction’ is done by Abhay Paunikar (56904) and Kaivalya Pande (56547) in partial fulfillment of the requirement for award of PG - Diploma in Big Data Analysis Course.

Mr. Girish Gaikwad
Project Guide

Mrs. Pradnya Dindorkar
Course Co-Ordinator

Date:

ACKNOWLEDGEMENT

It has been a great opportunity to gain lots of experience in real time projects, followed by the knowledge of how to actually design and analyze real projects. For that we want to thank all the teaching and non-teaching staff members of Sunbeam Institute who made it possible for students like us. Special thanks to Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mrs. Pradnya Dindorkar (Course Coordinator, SIIT, Pune) and Project Guide Mr. Girish Gaikwad for the efforts they did to provide us with all useful information and making the path clear for the students to implement all the education periods in real-time project design and analysis. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Abhay Paunikar

DBDA September 2021 Batch,

SIIT Pune

Kaivalya Pande

DBDA September 2021 Batch,

SIIT Pune

TABLE OF CONTENTS

1. Introduction

1.1. Overview

Airbnb is a home-sharing platform that allows home-owners and renters ('hosts') to put their properties ('listings') online, so that guests can pay to stay in them.

The hosts list their properties on Airbnb by entering all the information like its type, no. of accommodates, information about basic commodities like bedrooms, bathrooms, amenities provided at the facility etc. and the price is decided by the hosts.

Airbnb pricing is important to get right, particularly in big cities like NYC, Boston, Chicago, San Francisco, Los Angeles and Washington DC, where there is lots of competition and even small differences in prices can make the difference between optimum occupancy and high earnings, or being priced out of the market.

This project analyzes the data tries to get some insights about the Airbnb listing as well as to estimate the value of a property as accurate as possible using the different features of the property.

1.2. The Dataset

The dataset used for this project consists of data collected from the Airbnb website in the year 2019. This dataset was used for a Kaggle competition. It contains train.csv and test.csv. The description of all the columns/features is given below:

1) train.csv :

Train.csv has 28 columns and 74111 records.

- log_price – The logarithm of nightly price of the property

- property_type - Type of property, e.g. house or flat

- room_type - Type of listing, e.g. entire home, private room or shared room

- accommodates - How many people the property accommodates

- amenities - List of amenities in the property

- bathrooms - Number of bathrooms

- bed_type- Type of bed, e.g. real bed or sofa-bed

- cancellation_policy - The type of cancellation policy, e.g. strict or moderate

- cleaning_fee – Whether or not cleaning fee (a fixed amount paid per booking) is applicable to the property?

- city – The city to which the property belongs

- description - The description of the property as given by the owner

- first_review - The date of the first review

- host_has_profile_pic - Whether or not the host has uploaded his profile picture

- host_identity_verified - Whether or not the host has been verified with id

- host_response_rate - Proportion of messages that the host replies to

- host_since – Date since the owner started hosting on Airbnb

- instant_bookable - Whether or not the property can be instant booked (i.e. booked straight away, without having to message the host first and wait to be accepted)

- last_review - The date of the most recent review

- latitude

- longitude

- name – Name of the property

- neighborhood – Neighborhood to which the property belongs

- number_of_reviews - The number of reviews left for the property

- review_scores_rating - Guests can score properties overall

- thumbnail_url –URL of the thumbnail image uploaded by the host/owner

- zipcode

- bedrooms - Number of bedrooms

- beds - Number of beds

2) test.csv : It has the same columns as train.csv except ‘log_price’.

2. Problem Definition and Algorithm

2.1. Problem Definition

While listing their properties on Airbnb, the hosts are expected to set their own prices for their listings. Although Airbnb and other sites provide some general guidance, there are currently no free services which help hosts price their properties. Paid third party pricing software is available, but generally you are required to put in your own expected average price ('base price'), and the algorithm will vary the daily price around that base price on each day depending on day of the week, seasonality, how far away the date is, and other factors.

This project aims to use machine learning to predict the base price for properties in the above-mentioned US cities, and also to explore Airbnb listing data, in order to help Airbnb hosts maximize their earnings.

2.2. Algorithms tested

1. Simple Linear Regression

   Linear Regression is a supervised ML algorithm that helps find a suitable approximate linear fit to a collection of points. At its core, linear regression is a linear approach to identifying the relationship between two variables with one of these values being a dependent value and the other being independent. The idea behind this is to understand how a change in one variable impacts the other, resulting in a relationship that can be positive or negative.

2. Lasso Regression (L1 Regularization)

   The word “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. It is a statistical formula for the regularization of data models and feature selection. Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

3. Ridge Regression (L2 Regularization)

   Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

4. K – Nearest Neighbor

   KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same *neighbourhood*. The size of the neighbourhood needs to be set by the analyst or can be chosen using cross-validation (we will see this later) to select the size that minimises the mean-squared error.

   While the method is quite appealing, it quickly becomes impractical when the dimension increases, i.e., when there are many independent variables.

5. Support Vector Machines

   Support Vector Machines are categorized as supervised machine learning algorithms and are primarily used for classification and regression analysis. The algorithm works by building models that assign new examples and data to a category, where these categories are easily distinguishable from one another by a gap. SVM is highly effective in cases
   where the number of dimensions outweighs the number of samples and is extremely memory-efficient.

6. Random Forest

   Random Forests use a variety of algorithms for solving classification, regression, and similar problems by implementing decision trees. The way it works is, it creates heaps of decision trees with random sets of data, and a model is trained repeatedly on it for near-accurate results. In the end, all the results from these decision trees are combined to identify the best suitable result that appears most commonly in the output.

3. Experimental Evaluation:

3.1. Methodology:

The price prediction of Airbnb Listings is done by following the standard techniques used in machine learning like different feature selection techniques considering multicollinearity in the data. The complete methodology will be discussed further.

a) Loading the data using pandas and getting an overview of the data:

```
df = pd.read_csv('train.csv')
df.shape
df.columns
df.describe()
df.info()
```

b) Preprocessing of the data:

Since cleaning and pre-processing the data is the most crucial and time taking task in machine learning, it was given high priority. Data cleaning was done with the effort to avoid the loss of data and to make use of most of it.

Some textual columns in the dataset which were not required for model building could be identified by observing the data. For example, name, id, description, thumbnail_url etc. These columns were removed at an initial glance.

```
df.drop(['id', 'description', 'name', 'zipcode', 'thumbnail_url'], axis=1, inplace=True)
```

The columns like host_since, first_review, last_review, representing dates, were converted to datetime for analysis.

```
df.host_since = pd.to_datetime(df.host_since)

df.first_review = pd.to_datetime(df.first_review)

df.last_review = pd.to_datetime(df.last_review)
```

The dataset had some columns with missing values which can be better understood through the following table and heatmap:

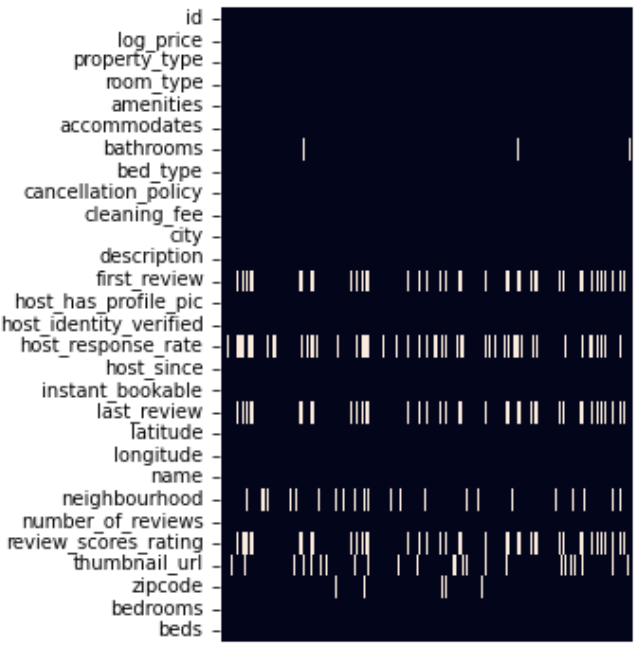

Fig 1: Heatmap representing missing values across the dataset

Feature	Percentage missing values
bathrooms	0.27%
first_review	21.41%
host_has_profile_pic	0.25%
host_identity_verified	0.25%
host_response_rate	24.69%
host_since	0.25%
last_review	21.36%
neighborhood	9.27%
review_scores_rating	22.56%
thumbnail_url	11.09%
zipcode	1.30%
bedrooms	0.12%
beds	0.18%

Table 1: Percentage of missing values across the dataset

The 3 date columns were converted to number of days by subtracting from current date and 3 new columns were created – host_days_active, time_since_first_review, time_since_last_review. Missing values in host_since were filled using the median of that series since it had very fewer missing values.

The count of missing values in first_review and last_review was large in number (more than 21%) so binning was performed here in order to create a new category of null values. The action of converting a numerical feature to categorical is called as binning.

The assigned categories can be seen below:

```
df.time_since_first_review.unique()
```

```
array(['0-4 years', 'nan', '4-6 years', '6-8 years', '8-10 years',
'10+ years'], dtype=object)
```

```
df.time_since_last_review.unique()
```

```
array(['2-4 years', '0-2 years', 'nan', '4-6 years', '6+ years'],
dtype=object)
```

Binning was also performed for review_scores_rating feature in a similar manner.

The features beds and bedrooms are directly related with the no. of accommodates. Hence, nulls in these columns are filled in proportion to the no. of accommodates.

Handling the amenities column:
The amenities column consisted of sets of amenities available in the properties. To get the unique amenities,

```
amenities_list = list(df.amenities)
amenities_list_string = " ".join(amenities_list)
amenities_list_string = amenities_list_string.replace('{', '')
amenities_list_string = amenities_list_string.replace('}', ',')
amenities_list_string = amenities_list_string.replace('"', '')
amenities_set = [x.strip() for x in amenities_list_string.split(',')]
amenities_set = set(amenities_set)
```

The set amenities_set consists of the 131 unique amenities. Some of these 131 amenities were more important than the others with respect to the price. Out of these, 28 amenities were selected and new columns were assigned to them, thus creating 28 new binary features (0, 1).

c) Feature Selection:

Feature selection is also called variable selection or attribute selection. It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modeling problem you are working on.

After the analysis of the amenities, we had 48 features with us. Out of these features we had to select the most important ones. For this, feature selection is done using the following methods and best performing method was selected.

1. By using Random Forest for feature selection:

```
from sklearn.feature_selection import SelectFromModel
sel = SelectFromModel(RandomForestRegressor(n_estimators =
130))
sel.fit(X_scaled, Y)
selected_feat = pd.DataFrame(X_scaled)
selected_feat = selected_feat.columns[(sel.get_support())]
print(selected_feat)
```

2. By filtering the features based of correlation coefficient and Anova test

```
from sklearn.feature_selection import SelectKBest, f_classif
mod = SelectKBest(f_classif,k=10)
new = mod.fit(X,y_ano)

x_1 = new.transform(X)
y_1 = df.log_price.values

columns = new.get_support(indices=True)
df_new = df.iloc[:,columns]_engg="", category_encoders="")
```

3. By using an external package featurewiz

   featurewiz is a python library to find the best features in your dataset if you give it the dataframe and the name of the target variable. It will do the following:

   - Remove highly correlated features automatically (the limit is set to 0.70 but you can change it in the input argument)
   - If multiple features are correlated to each other, which one to remove? In case of such clashes, it will remove the feature with the lower mutual information score
   - Recursively do feature selection using XGBoost algorithm to find the best features using XGBoost.

```
From featurewiz import featurewiz

target = 'log_price'
features, train = featurewiz(df, target, corr_limit=0.4, verbose=2, sep=",",
header=0,test_data="", feature_engg="", category_encoders="")
```

d) **Removing multicollinearity:**

Multicollinearity may not affect the accuracy of the model as much. But we might lose reliability in determining the effects of individual features in your model – and that can be a problem when it comes to interpretability. That is why, an attempt is made to remove the multicollinearity in the data.

```
mul_colinear_cols = []

for col1 in df.columns:

  for col2 in df.columns:

      if (col1 != col2) and (df[col1].corr(df[col2])**2 > 0.25):

        mul_colinear_cols.append((col1,col2))
```

In this way, columns with multicollinearity are identified and some of them are removed, provided they have a weak correlation with the target variable.

```
def remove_multicoll():
    for tp in mul_colinear_cols:
        for col in tp:
            if col == 'log_price':
                mul_colinear_cols.remove(tp)


for i in range(15):
    print("interation",(i+1))
    remove_multicoll()
    print(len(mul_colinear_cols))
```

e) Project Flow Diagram

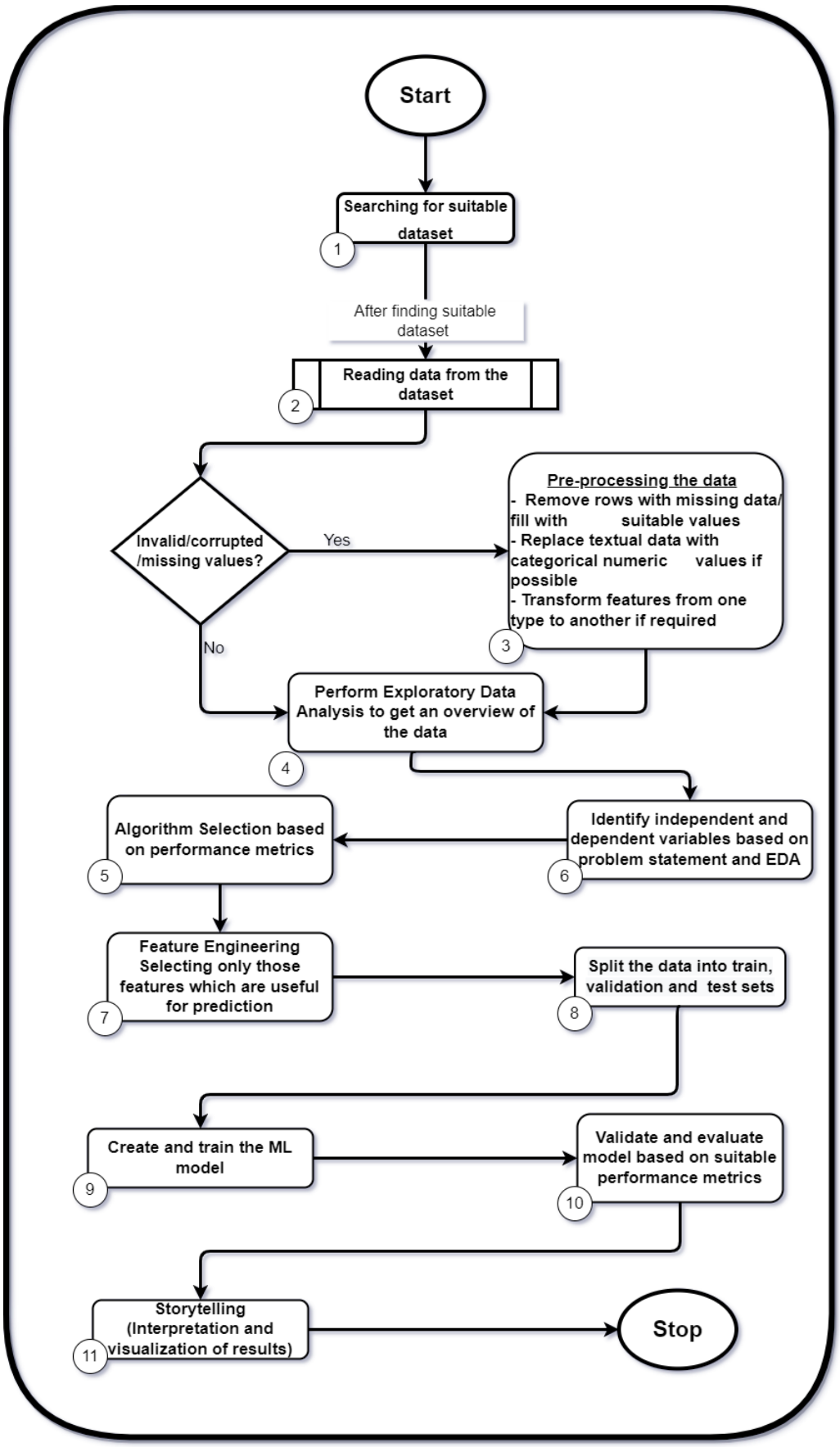

Fig 2: Project Flow Design

3.2. Exploratory Data Analysis

The data consists of following columns:

Type of feature	Features
Categorical Columns	property_type, city, room_type, bed_type, cancellation_policy
Binary Columns	cleaning_fee, host_has_profile_pic, host_identity_verified, 'instant_bookable'
Numerical Columns	accommodates, bathrooms, host_response_rate, number_of_reviews, review_scores_rating, bedrooms, beds
Date Columns	host_since, first_review, last_review

Table 2: Types of features in the dataset

For the categorical columns, countplots are plotted to take a glance at the distribution of the data among the respective categories. For example, the average price plotted against the property_type.

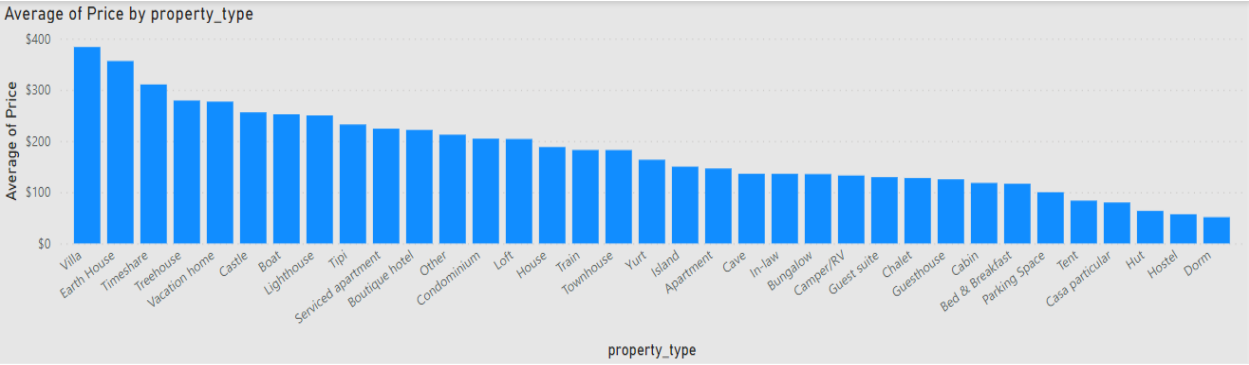

Fig 3: Average Price by Property Type

From this plot we can see the average price by property type. Villa, earth house, treehouse are the costliest property types. Whereas, tent, hut, hostel, dormitories can be considered in affordable category.

Some other important features are bed type, no. of bathrooms and no. of bedrooms.

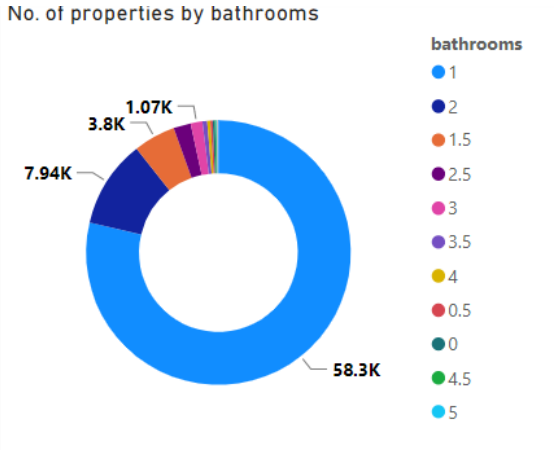

Fig 4: No. of Properties by bathrooms

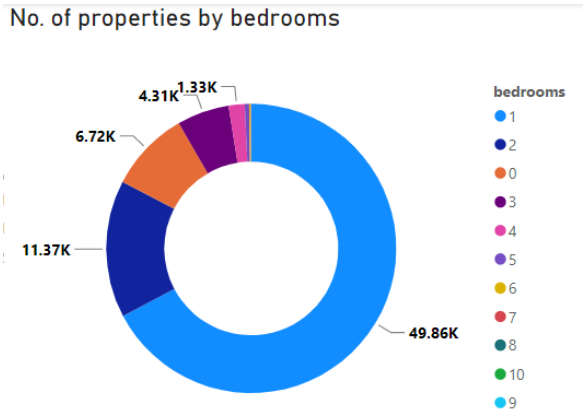

Fig 5: No. of Properties by bedrooms

From the first donut chart, it can be seen that most of the listings have one bathroom and one bedroom.

The below bar-chart of city wise average price gives a clear overview about the variation in property prices in different cities.
The average price is highest in San Francisco, followed by Washington DC. Both of these cities have avg. price more than $200 whereas all other cities have avg. price $165 or less, Chicago being the least at $132.

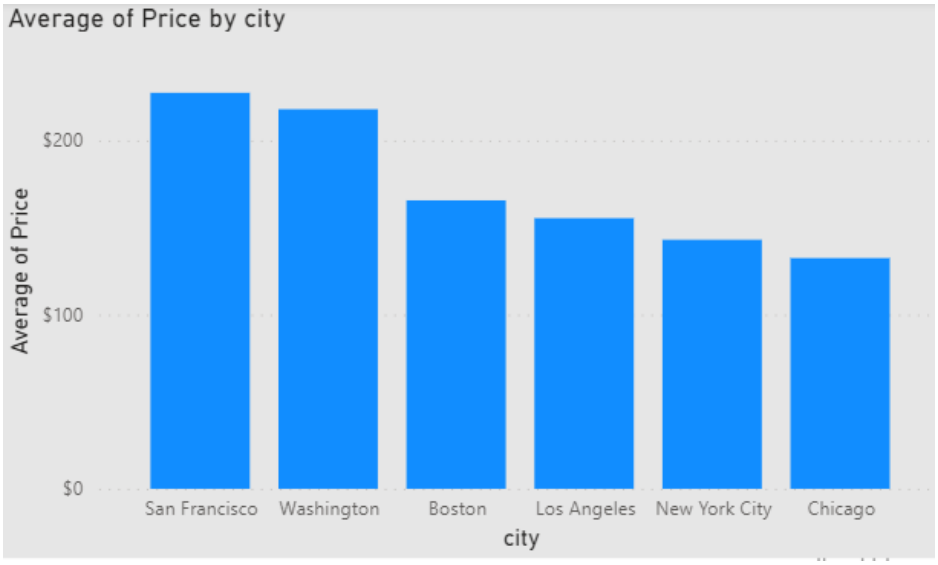

Fig 6: Average Price by City

Time Series:

The following line graph demonstrates the monthly avg no. of reviews given by the tenants. In US, the schools resume in the month of August after summer break of 10 – 11 weeks. That is why we see a dip in no. of reviews in this month. On the other hand, the no. of reviews peaks in the months of February and October which have winter break and fall break respectively and thus most people tend to go on vacation and rent properties to stay in these one-week breaks.

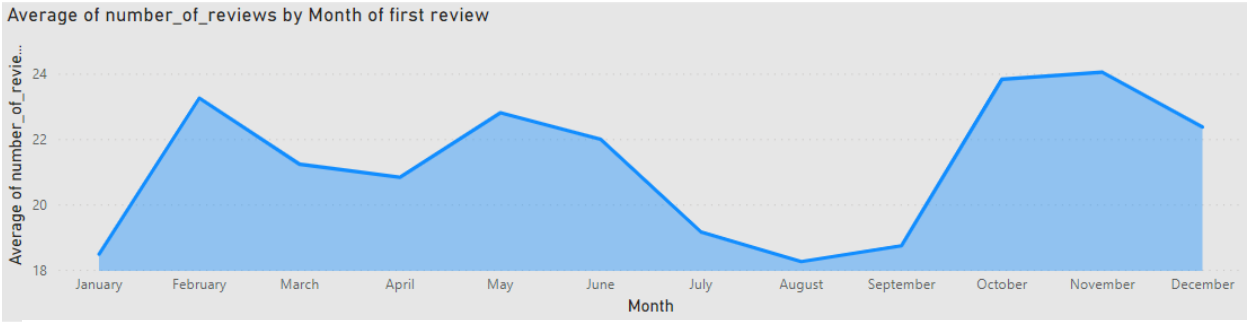

Fig 7: Monthly no. of reviews by customers

Another important chart is the no. of properties listed through the last decade. Airbnb started in 2008 and is presently the most popular platform for the property owners as well as customers who want to rent a property. The below plot shows that the number of properties listed on Airbnb increased from 2008 through 2015 and then decreased slightly.

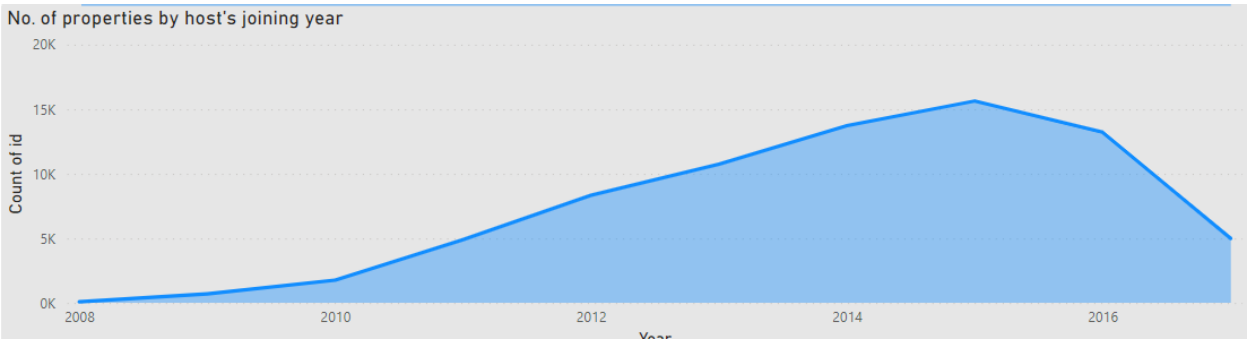

Fig 8: Growth in no. of listings over the years

4. Results and Discussion

For predicting the property price, different algorithms like Simple Linear Regression, Ridge Regression, Lasso Regression, K-Nearest Neighbors, Support Vector Machines and Random Forest were considered. The evaluation parameters of the algorithms were mainly RMSE and R2_Score. The results of all these algorithms are displayed in the table below:

Algorithm	**MAE**	**MSE**	**RMSE**	**R2_Score**
Simple Linear	0.3624	0.2274	0.4769	0.5585
Ridge	0.3624	0.2274	0.4769	0.5585
Lasso	0.5628	0.5151	0.7177	0.0002
KNN	0.3688	0.2371	0.4869	0.5397
SVM	0.3304	0.1968	0.4436	0.6179
Random Forest	0.3061	0.1742	0.4174	0.6417

Table 3: Algorithms used and their results

Out of all these algorithms, the best performer was Random Forest Regressor which is an ensemble learning technique which uses a large number of decision trees for prediction. SVM also gives a value close to Random Forest but the training time for SVM is very high. Random Forest gave an R2_Score of 64.17% and RMSE value of 41.17% with considerably less training time on initial training.

5. GUI

The GUI is developed for this project using Python's Flask framework in order to be used by a non-programming person. Flask is considered a microframework. Micro-frameworks are the frameworks with little or no dependencies to external libraries. This has pros and cons. Flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

6. Future Work and conclusion

6.1. Future Work

This project primarily used machine learning algorithms like regression, decision trees, random forest etc. From the results it can be stated that there is a scope for increasing the predictive accuracy of the model. This can be done by training the model using deep learning algorithms like Neural Networks or LSTM. Secondly, more efforts can be made to reduce the training time of the algorithms.

6.2. Conclusion

- Most of the properties are Apartments, followed by entire houses and condominiums.
- The period of school holidays highly affects the number of bookings and reviews on Airbnb properties per year, as in this period, a significant number of people opt for long vacations.
- The price of Airbnb properties is mainly affected by the no. of accommodates, bedrooms, beds and bathrooms.
- The properties in the cities of San Francisco and Washington DC are a little expensive as compared to Boston, Los Angeles, NYC and Chicago.
- Random Forest Regressor is the best performer out of the various models built on the data.