Comparative Analysis of Milan, Lombardy, Italy AirBnB Listings

Irene "Nhi" Ong
Georgia State University

Binh Le
Georgia State University

Jacob Nguyen
Georgia State University

Abstract

This paper describes the data analysis our team performed on the Milan, Lombardy, Italy AirBnB datasets. We used several different unsupervised learning algorithms to represent and visualize the data sets. We compared the pre-pandemic AirBnB listings of 2019 to the mid-pandemic AirBnB listings of 2020. Following, we attempt to predict the lodging prices post-pandemic.

Keywords—data mining, AirBnB, unsupervised learning, clustering, PCA, pandemic, coronavirus, Milan, Lombardy, Italy, housing

1 Introduction

Earlier this year, the world was seized by the novel coronavirus, formerly known as SARS-CoV-2, and entered into a new pandemic era. The pandemic has forced lifestyle changes for the individual and closures for large businesses and industries. More evidently, it has impacted travel and tourism industries most severely. AirBnB, a vacation rental lodging company, has revolutionized the way people shop for vacation rentals and housing. It has dominated the lodging market for the last few years, even competing with the well established hotel industry. However, dependent on travel and tourism, the lodging rental company was heavily impacted by the pandemic.

We propose a comparative analysis of Milan, Lombardy, Italy AirBnB listings from last year and this current year. We choose this location because Lombardy was one of the COVID-19 hotspots earlier this year in mid-March. We specifically chose Milan because the city has maintained its status as a global fashion capital for many years, attracting millions of tourists each year. We hope to find meaningful patterns that can help the company understand how the pandemic has affected their clients.

Our goal is to seek patterns that provide insight into how the pandemic has impacted the Airbnb lodging in Lombardy, Italy by looking at the pre-pandemic listings (2019) and intra-pandemic listings (2020). With that purpose in mind, we will take an exploratory approach and use unsupervised learning algorithms to represent and visualize the data sets. Following, we will perform a comparative analysis on the results of the two data sets. Using the findings from this analysis, we will predict the lodging prices post-pandemic. This paper will describe the methodologies that we used including the algorithms and technologies. Finally, we describe our final findings with the support of the data visualizations

1.1 Key Algorithms and Technologies

We plan to use unsupervised learning algorithms, or more specifically, clustering algorithms, to analyze the data. Some key algorithms we may use are Principle Component Analysis, K-Means, and Hierarchical clustering. In the case of predicting lodging prices post-pandemic, we will use regression to predict post-pandemic prices.

We will use several Python libraries to help represent and visualize the data, including but not limited to: Pandas, Matplotlib, and Plotly.

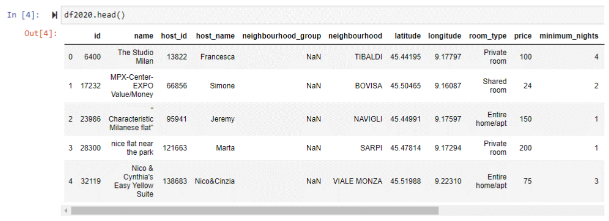
2 Dataset analysis

AirBnB provides free, open data sets on a secondary website. Several csv files including listings, reviews, and neighborhoods are provided. Archived data from last year included. We will use the "Milan, Lombardy, Italy" data set found at: http://insideairbnb.com/get-the-data.html

The "Milan, Lombardy, Italy" listings.csv file for August 31, 2020 has 74 features such as name, amenities, property type, price, and superhost status. The features describe 11,993 data points. The calendar.csv

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

file provides information about availability, price, adjusted price, minimum nights and maximum nights for specific dates. There are more than one hundred thousand data points for the features in the calendar file. In both files, there are both categorical and numerical data values.



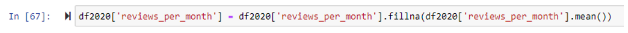
```
In [4]:  df2020.head()
Out[4]:
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights
0	6400	The Studio Milan	13822	Francesca	NaN	TIBALDI	45.44195	9.17797	Private room	100	4
1	17232	MPX-Center-EXPO Value/Money	66856	Simone	NaN	BOVISA	45.50465	9.16087	Shared room	24	2
2	23986	" Characteristic Milanese flat"	95941	Jeremy	NaN	NAVIGLI	45.44991	9.17597	Entire home/apt	150	1
3	28300	nice flat near the park	121663	Marta	NaN	SARPI	45.47814	9.17294	Private room	200	1
4	32119	Nico & Cynthia's Easy Yellow Suite	138683	Nico&Cinzia	NaN	VIALE MONZA	45.51988	9.22310	Entire home/apt	75	3

Figure 1: A snapshot of our data set

2.1 Data Preprocessing

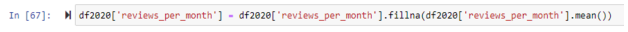
We employed a number of methods to pre-process the data and reduce invalid outputs such as column dropping, value replacement by mean, outlier removal, and variable scaling. Our first goal was to remove noise values. Both datasets, the listings in 2019 and the listings in 2020, contained columns with missing values. The values for the "name", "host name', "id", and "host id" columns contained identifiers unrelated to the price, the variable we were interested in. The "neighbourhood group" column does not contain any values, and the "last review" column simply indicates the date of the last review, baring no relation to the price feature. These values were all removed via the drop function.



```
In [67]:  df2020['reviews_per_month'] = df2020['reviews_per_month'].fillna(df2020['reviews_per_month'].mean())
```

Figure 2: Dropping noise values

There was still a "reviews per month"column that contained some NaN values after deleting unrelated columns and the column with only NaN values. To process this column, we replaced the NaN values with the mean value, using the fillna function.



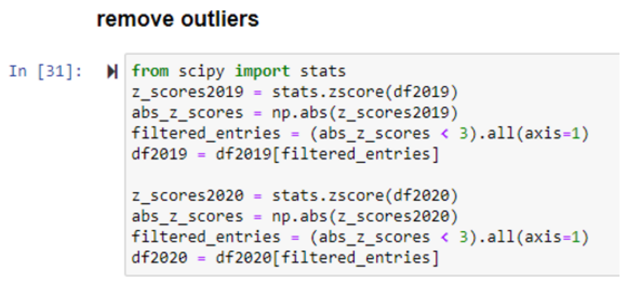
```
In [67]:  df2020['reviews_per_month'] = df2020['reviews_per_month'].fillna(df2020['reviews_per_month'].mean())
```

Figure 3: Using the fillna function

After performing these functions, we removed the outliers and scaled the variables as shown in Figure 4 and 5, respectively.

2.2 Data Exploration

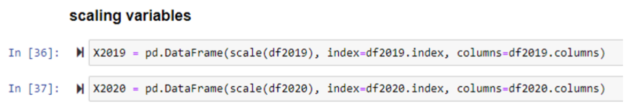
In this section, we summarize the algorithms we performed on the data sets. Then, we discuss the output



remove outliers

```
In [31]:  from scipy import stats
          z_scores2019 = stats.zscore(df2019)
          abs_z_scores = np.abs(z_scores2019)
          filtered_entries = (abs_z_scores < 3).all(axis=1)
          df2019 = df2019[filtered_entries]

          z_scores2020 = stats.zscore(df2020)
          abs_z_scores = np.abs(z_scores2020)
          filtered_entries = (abs_z_scores < 3).all(axis=1)
          df2020 = df2020[filtered_entries]
```

Figure 4: Removing outliers



scaling variables

```
In [36]:  X2019 = pd.DataFrame(scale(df2019), index=df2019.index, columns=df2019.columns)

In [37]:  X2020 = pd.DataFrame(scale(df2020), index=df2020.index, columns=df2020.columns)
```

Figure 5: Scaling variables

results between the two different data sets.

2.2.1 Principle Component Analysis

Principle Component Analysis (PCA) is an unsupervised learning method, and it is the process of calculating principle components from an original data set and using these new components to analyze the original data set. We use PCA when we want to visualize high-dimensional data because it's more difficult to analyze data with multiple variables or axis's. To simplify, PCA works by first projecting the data onto a line and maximizing the distances of the points to the origin to find the best fitting line. This line becomes PC1 and the other variables (PC2, PC3, etc.) can be drawn as a perpendicular line to this one. Once the other variables are accounted for, you rotate the plot until PC1 is horizontal and then you calculate the variation for each of the principal components. Scree plots are typically used to represent the variation percentages of each principle component and we would look at the two PCs that account for most of the variation in the data to analyze.

Some advantages of PCA is that it makes high dimensional data much easier to visualize by reducing the number of features in the data; however, there are some limitations to this algorithm. For example, we lose some parts of the data because we usually only look at the parts that account for most of the variation, leaving out the parts that don't account for very much of the variation. The data must also be quantified numerically before we can apply PCA to it.

Our data sets were an excellent candidate for PCA as they both contained a large number of features and consisted of numerical values. Figure 6 represents the PCA graph for the 2019 AirBnB listings while Figure 7 represents the PCA graph for the 2020 AirBnB listings. We compared the price attribute against the other attributes.

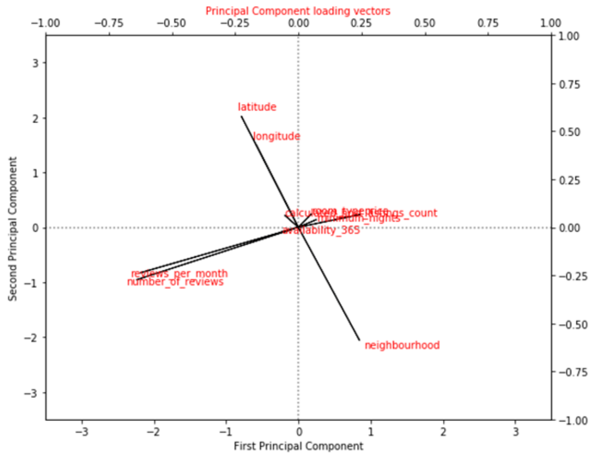


Figure 6: PCA performed on 2019 listings

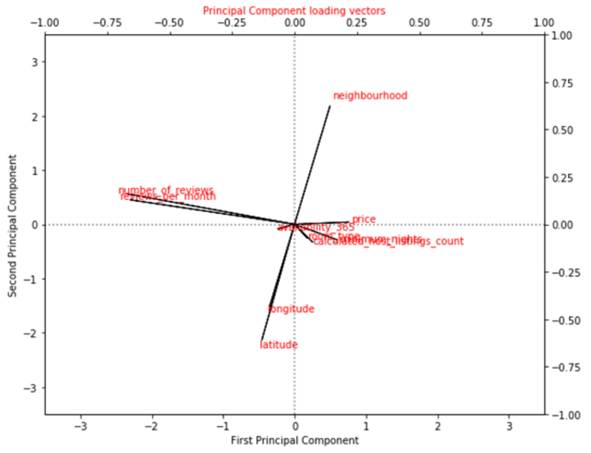


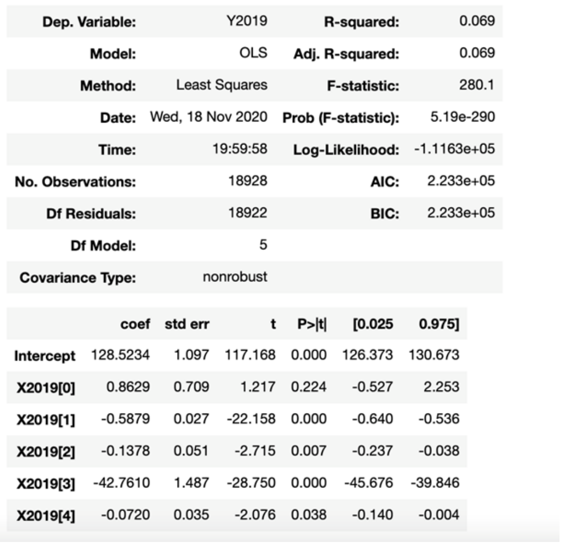
Figure 7: PCA performed on 2020 listings

In both graphs, 'latitude', 'longitude', and 'neighbourhood' did not correlate with the 'price' attribute. We also saw that 'reviews per month', 'number of reviews', and 'minimum nights' are negatively correlated with the 'price' attribute, while 'calculated host listings count' and 'room type' are positively correlated with the 'price' attribute.

2.2.2 Ordinary Least Squares

Ordinary least squares (OLS) is a method used with regression models (typically linear regression models) to estimate unknown parameters. When OLS is performed on a regression model, a straight line is fitted through data points that minimize the sum of squared differences between the observed values and the predicted values. According to the Gauss-Markov Theorem, as long as the standard assumptions are met, the OLS method has the lowest sampling variance compared to other linear unbiased estimating methods.

In the 2019 listings and 2020 listings data sets, the 'review per month', 'number of reviews', 'calculated host listings count', 'minimum nights' and 'room type' attributes are correlated with 'price', so we choose these five attributes as independent variables for the model; the 'price' attribute is the dependent variable.



Dep. Variable:	Y2019	**R-squared:**	0.069
Model:	OLS	**Adj. R-squared:**	0.069
Method:	Least Squares	**F-statistic:**	280.1
Date:	Wed, 18 Nov 2020	**Prob (F-statistic):**	5.19e-290
Time:	19:59:58	**Log-Likelihood:**	-1.1163e+05
No. Observations:	18928	**AIC:**	2.233e+05
Df Residuals:	18922	**BIC:**	2.233e+05
Df Model:	5		
Covariance Type:	nonrobust		

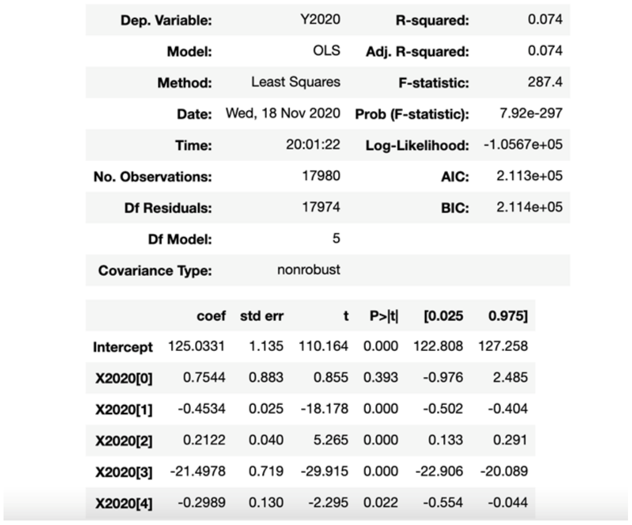
	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	128.5234	1.097	117.168	0.000	126.373	130.673
X2019[0]	0.8629	0.709	1.217	0.224	-0.527	2.253
X2019[1]	-0.5879	0.027	-22.158	0.000	-0.640	-0.536
X2019[2]	-0.1378	0.051	-2.715	0.007	-0.237	-0.038
X2019[3]	-42.7610	1.487	-28.750	0.000	-45.676	-39.846
X2019[4]	-0.0720	0.035	-2.076	0.038	-0.140	-0.004

Figure 8: OLS for 2019 listings

Figure 8 illustrates the results of the OLS estimator performed on the 2019 listing data set. The coefficient for the 'reviews per month' attribute is represented by X2019[0] and has a p-value of 0.224; as this value is higher than 0.05, we failed to reject the null hypothesis of this coefficient and the true value of this coefficient can be 0. The model is represented with:

$$\begin{aligned} price = {} & 128.5234 + 0.8629 * (\text{reviewspermonth}) \\ & - 0.5879 * (\text{numberofreview}) \\ & - 42.7610 * (\text{roomtype}) \\ & - 0.0720 * (\text{minimumnights}) \\ & - 0.1378 * (\text{calculatedhostlistingscount}) \end{aligned} \tag{1}$$

Figure 9 illustrates the results of the OLS estimator performed on the 2020 listing data set. The coefficient for the 'reviews per month' attribute is represented by X2020[0] and has a p-value of 0.393; as this value is higher than 0.05, we failed to reject the null hypothesis of this coefficient and the true value of this coefficient can be 0.This model is represented with:



Dep. Variable:	Y2020	**R-squared:**	0.074
Model:	OLS	**Adj. R-squared:**	0.074
Method:	Least Squares	**F-statistic:**	287.4
Date:	Wed, 18 Nov 2020	**Prob (F-statistic):**	7.92e-297
Time:	20:01:22	**Log-Likelihood:**	-1.0567e+05
No. Observations:	17980	**AIC:**	2.113e+05
Df Residuals:	17974	**BIC:**	2.114e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	125.0331	1.135	110.164	0.000	122.808	127.258
X2020[0]	0.7544	0.883	0.855	0.393	-0.976	2.485
X2020[1]	-0.4534	0.025	-18.178	0.000	-0.502	-0.404
X2020[2]	0.2122	0.040	5.265	0.000	0.133	0.291
X2020[3]	-21.4978	0.719	-29.915	0.000	-22.906	-20.089
X2020[4]	-0.2989	0.130	-2.295	0.022	-0.554	-0.044

Figure 9: OLS for 2020 listings

$$\begin{aligned}
price = 125.0331 &+ 0.7544 * (\text{reviewspermonth}) \\
&- 0.4534 * (\text{numberofreview}) \\
&- 21.4978 * (\text{roomtype}) \\
&- 0.2989 * (\text{minimumnights}) \\
&+ 0.2122 * (\text{calculatedhostlistingscount})
\end{aligned} \tag{2}$$

2.2.3 Comparing the two models

We compared the results of the two models and made the following observations:

- The price of 2019 starts at 128.52 USD and the price of 2020 starts at 125.03 USD. For the 2019 listings' "reviews per month" attribute, when this attribute is raised by one unit, the price increases to 0.86 USD. However, in the 2020 listings, when this attribute is raised by one unit, the price only increases to 0.75 USD.

- For the 2019 listings' "number of reviews" attribute, when this attribute is raised by one unit, the price decreases to 0.59 USD. In the 2020 listings, when this attribute is raised by one unit, the price decreases to 0.45 USD.

- Looking at the "calculated host listings count" attribute, there is an obvious change between the 2019 listings and the 2020 listings. In particular, in the 2019 listings, when this attribute is raised by one unit, the price decreases to 0.14 USD. However, in the 2020 listings, when this attribute is raised by one unit, the price increases to 0.21 USD.

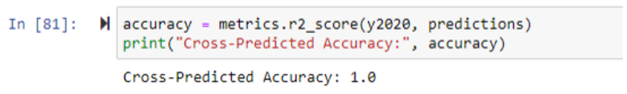
- Similarly, there is a significant change in the "room type" attribute between the 2019 listings and the 2020 listings. In the 2019 listings, when this attribute is raised by one unit, the price decreases to 42.76 USD. However, in the 2020 listings, when this attribute is raised by one unit, the price only decreases to 21.50 USD.

- Looking at the "minimum nights" attribute, in the 2019 listings, when this attribute is raised by one unit, the price decreases to 0.07 USD, and in the 2020 listings, when this attribute is raised by one unit, the price decreases to 0.30 USD.

2.2.4 k-Fold Cross Validation

Model evaluation is a very important step that must be performed to determine if a model is usable. It also informs us whether we need to re-select features or use another model. Cross validation is essentially a model evaluation method or resampling procedure. In a k-Fold cross validation, k refers to the number of groups a data set is split into. To perform this procedure, the data is first shuffled randomly and then, split into k groups where each group is further split into test and training sets. The training set in each unique group is fit to a model and evaluated with the test set. We summarize the data set with the evaluation scores obtained from each of the unique group models.

We decided to use this method because it's cost-effective, and it does not overestimate test error rate as much as other approaches. After performing the k-fold cross validation procedure on our data set, we received a model accuracy score of 1.0 or 100%.



```
In [81]:  accuracy = metrics.r2_score(y2020, predictions)
          print("Cross-Predicted Accuracy:", accuracy)

          Cross-Predicted Accuracy: 1.0
```

Figure 10: k-Fold model score

2.3 Visualization

In this section, we present decision trees as our primary method of visualizing the data sets. Then, we discuss our understanding of the data based on the resulting decision tree.

2.3.1 Decision Trees

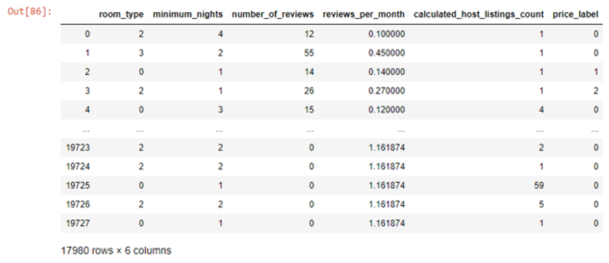
Decision trees are a non-parametric supervised learning method used to visualize and represent tests on the data set attribute (the node or internal node) and the outcomes of each test (the leaf node or terminal node). They are composed of two main parts: nodes and branches. At each node, a feature undergoes a test to evaluate it. After evaluation, the observations are split, and the new intermediate nodes undergoes a test. This process is repeated recursively until it reaches its growth-stop condition. These conditions may be the tree's maximum depth, the leaves' minimum samples, or minimum reduction error.

There are many advantages to using decision trees: they are easy to understand, they can be represented graphically, and they can handle both quantitative and categorical data. However, this method still has disadvantages in that they are not robust and small changes in the data can leave substantial changes in the final results, and decision trees may not have the same predictive accuracy as other regression and classification approaches.

We used decision trees to evaluate which factors affect the price of lodging, and based on that criteria, help customers make decisions about rent lodging in this specific location. We also used the decision tree to predict which classes customers fall under with regard to price. We decided to model our data with decision trees because they are easily interpreted and they tolerate numeric and categorical predictor variables and missing values.

2.3.2 Preparing the data

We eliminated unnecessary attributes to ensure the decision tree can deliver highly reliable information.



Out[86]:

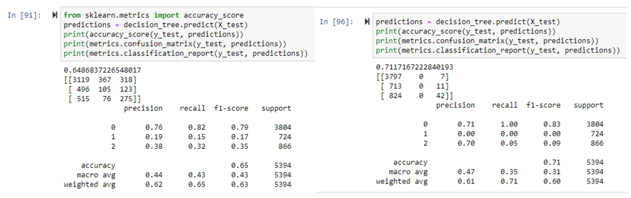
	room_type	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	price_label
0	2	4	12	0.100000	1	0
1	3	2	55	0.450000	1	0
2	0	1	14	0.140000	1	1
3	2	1	26	0.270000	1	2
4	0	3	15	0.120000	4	0
...	...	...	...	...	...	...
19723	2	2	0	1.161874	2	0
19724	2	2	0	1.161874	1	0
19725	0	1	0	1.161874	59	0
19726	2	2	0	1.161874	5	0
19727	0	1	0	1.161874	1	0

17980 rows × 6 columns

Figure 11: Creating the new "price label" column

The price attribute has a large range, so we created a column, "price label," to evaluate the price values based on three categories: 0 (low price), 1 (medium price), 2 (high price). Then, we mapped the prices into three ranges of 0-100, 100-150, 150-10000 respectively with the new "price label" column's categories of 0, 1, 2.

When we explicitly show the "max length" for the decision tree, we saw an increase in the accuracy from 0.65 to 0.71 (The shorter the decision tree, the easier it is to make a decision.)

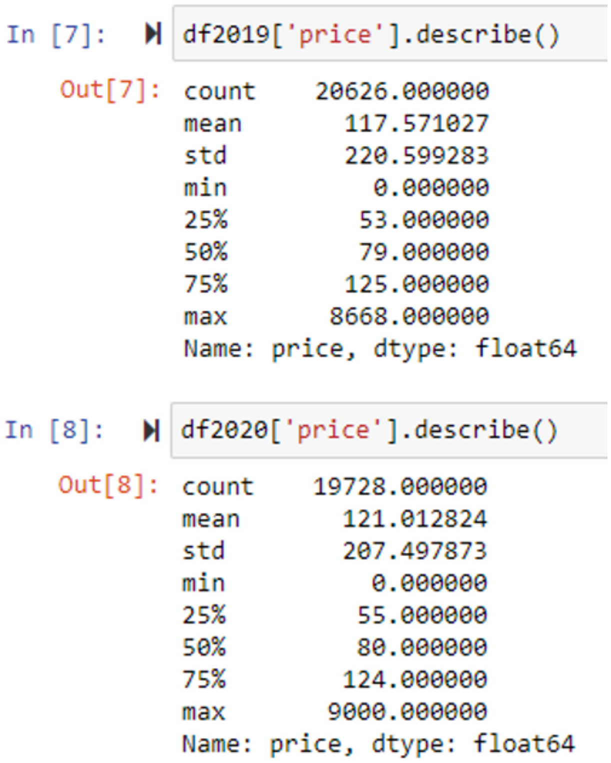


```
In [91]:  from sklearn.metrics import accuracy_score
          predictions = decision_tree.predict(X_test)
          print(accuracy_score(y_test, predictions))
          print(metrics.confusion_matrix(y_test, predictions))
          print(metrics.classification_report(y_test, predictions))

          0.6486837226548017
          [[3119  367  318]
           [ 496  105  123]
           [ 515   76  275]]
                        precision    recall  f1-score   support

                     0       0.76      0.82      0.79      3804
                     1       0.19      0.15      0.17       724
                     2       0.38      0.32      0.35       866

              accuracy                           0.65      5394
             macro avg       0.44      0.43      0.43      5394
          weighted avg       0.62      0.65      0.63      5394
```

```
In [96]:  predictions = decision_tree.predict(X_test)
          print(accuracy_score(y_test, predictions))
          print(metrics.confusion_matrix(y_test, predictions))
          print(metrics.classification_report(y_test, predictions))

          0.7117167222840193
          [[3797    0    7]
           [ 713    0   11]
           [ 824    0   42]]
                        precision    recall  f1-score   support

                     0       0.71      1.00      0.83      3804
                     1       0.00      0.00      0.00       724
                     2       0.70      0.05      0.09       866

              accuracy                           0.71      5394
             macro avg       0.47      0.35      0.31      5394
          weighted avg       0.61      0.71      0.60      5394
```

Figure 12: Before and after max length (left to right)

An observation Even though COVID-19 is still ongoing, the price in the 2020 listings is still higher than in the 2019 listings. Specifically, the price mean in 2019 is only 117.57 USD, while the price mean in 2020 is 121.01 USD. The maximum price in 2019 is $8,668 USD, whereas the maximum price in 2020 is $9,000 USD.



```
In [7]:   df2019['price'].describe()

Out[7]:   count    20626.000000
          mean       117.571027
          std        220.599283
          min          0.000000
          25%         53.000000
          50%         79.000000
          75%        125.000000
          max       8668.000000
          Name: price, dtype: float64

In [8]:   df2020['price'].describe()

Out[8]:   count    19728.000000
          mean       121.012824
          std        207.497873
          min          0.000000
          25%         55.000000
          50%         80.000000
          75%        124.000000
          max       9000.000000
          Name: price, dtype: float64
```

Figure 13: 2019 and 2020 price differences

Understanding We created a graph to chart the features based on their importance score. This helped us understand which features are useful in predicting lodging prices.

Each features' influence is clearly presented in Figure 14. The factors that most affect the price are "reviews per month". The feature that had the least influence on price is "room type". This shows that customers are interested in the number of reviews and reviews

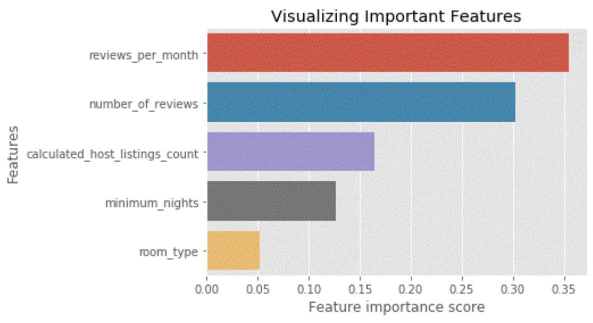


Figure 14: Feature Importance Scores

per month when deciding whether or not to rent an AirBnB lodging arrangement. These factors significantly affect lodging prices.

Our resulting decision tree is seen below in Figure 15. The decision tree begins with the “number of reviews” feature, where the next step is determined by whether the value is less than or equal to 2.5. Our decision tree classifies a listing according to its features to determine what kind of customer it will be suitable for. In our decision tree, “class” corresponds with the “price label” denoting which price range the listing falls under. We can use this information to determine what kind of customers are renting a certain lodging accommodation, or conversely, what lodging accommodation is suitable for a customer of say, a higher economic status.

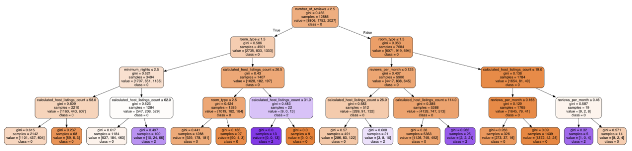


Figure 15: Decision Tree

For example, if “number of reviews” ¡= 2.5 is true and we follow this branch to “room type” ¡= 1.5, then to “minimum nights” ¡= 2.5, and lastly, to “calculated host listings count” is greater than 62.0, we arrive at a result of Class 2. These customers are classified into a group of customers who can rent lodging at a high price.

Conclusion

We compared the pre-pandemic AirBnB listings of 2019 to the mid-pandemic AirBnB listings of 2020 to better understand how the coronavirus pandemic has affected the AirBnB listings. We performed a number of data mining algorithms on our two data sets including: Principle Component Analysis, Ordinary Least Squares regression modeling, k-Fold Cross Validation, and Decision Trees.

Using Ordinary Least Squares (OLS) to model our data to a regression line, we constructed a model that enables the prediction of lodging market values based on various characteristics, thus allowing buyers and sellers to make a preliminary judgment when purchasing or pricing a lodging accommodation. This model allowed us to make several observations about our data sets in regard to pricing. We were able to pinpoint how the other features such as reviews per month, number of reviews, room type, minimum nights, and calculated host listings count are related to the listing price. We also observed how these features changed between the two years.

We used decision trees to evaluate which factors affect the price of lodging, and based on that criteria, help customers make decisions about rent lodging in this specific location. Our decision tree model can help AirBnB hosts determine how to price their lodging accommodation as well as provide more insight into what kind of customer they can expect. Conversely, as a customer, the results can help clients choose the right lodging based on the factors in the decision tree.

Despite the ongoing COVID-19 pandemic, the pandemic has not seemed to significantly affect the Airbnb market in Milan. The prices in the Airbnb market of Milan, Lombardy, Italy seems to have remained the same from 2019 to 2020, indicating that COVID-19 does not seem to be a factor in affecting the price of rental lodging.

References

This section is for our references. [2] [1]

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning data mining, inference, and prediction. *Second Edition*, 12th Printing, 2017.

[2] Jaya Krishna Mandivarapu. Slides. *Data Mining*, 2020.