Sentimental Analysis applied at Airbnb review

Guilherme da S. Pereira[1], Fernando H. M. de Paula[1]

[1]Faculdade de Economia do Porto, Universidade do Porto, Porto, Portugal.

Abstract. The presented paper aims to make sentimental analysis with dataset provided from Airbnb where evaluations from clients were provided from room of Boston, US. For this intent, libraries from python in this niche were used, and further, correlation between characteristics from room were established to infer cause and effect.

Keywords: Text Mining, Natural Language Processing, Sentiment Analysis, Tourism, Airbnb, Tourist accommodation.

1 Introduction

"Natural language processing (NLP) is a theory-motivated range of computational techniques for the automatic analysis and representation of human language." Cambria (2014).

In other words, NLP is a way that machines and computers understand our language. In this context, the project will explore a case study about Text Mining which nowadays is a very powerful technique to evaluate large collections of documents and discover new information and answer questions. Considering the amount of information present on the World Wide Web. However, the most of this text data is produced for human consumption, leading to unstructured data which requires pre-processing techniques to make the data understandable to the machines.

In the scope of this project, sentiment analysis was done, which it is the process used to determine the emotional tone for of a piece of text. This analysis It is an important tool for businesses in order to understand customer feedback, social media sentiment, and brand reputation due to their sentiment analysis has several diverse applications in the real world.

Intended to make this the sentiment analysis, we used the NLTK (Natural Language Toolkit) for that. It consists of a package from Python which provides a wide range of libraries for text analysis, furthermore, it is widely used, well-documented and has a large community of users and contributors. NLTK includes pre-trained models for sentiment analysis, which was used to classify the sentiment of Airbnb reviews (a piece of text) as positive, negative, or neutral.

The data used was collected from insideairbnb.com which is a project that provides data about Airbnb services along the world. Within the data available on the site, we chose to analyze the data concerning the guest reviews and the data about the accommodation's characteristics.

The goal of this project is to demonstrate with real data the challenges of transforming a business problem into a data problem, through a huge quantity of data available from guest reviews for Airbnb accommodations at the Boston City between 2015 and 2023.

Leading to how text mining and sentiment analysis can be used to extract insights from a large volume of unstructured data, and how it can help businesses make data-driven decisions that improve customer satisfaction.

2 Background and problem description

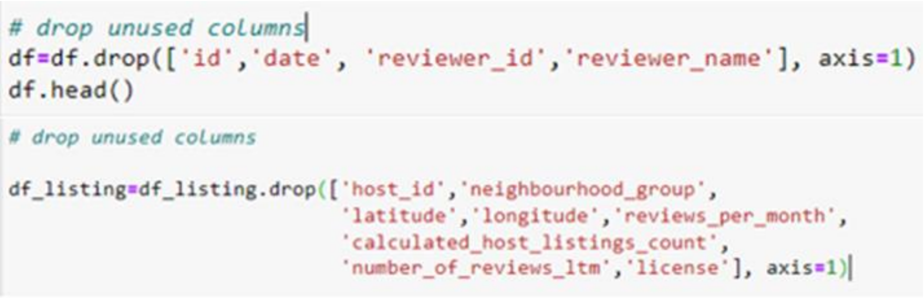
The tourism industry has always been heavily connected to customer satisfaction. With the rise of online platforms such as Airbnb, the volume of guest reviews has increased exponentially. Considering that manually analyzing this vast amount of data can be time-consuming and error prone, there is a growing need for automated tools and techniques that allow the extraction of valuable insights from this data.

Sentiment analysis is a powerful tool for analyzing customer feedback and sentiment in the tourism industry, using NLP and machine learning techniques, sentiment analysis can automatically classify guest reviews as positive, negative, or neutral, and extract valuable insights into customer satisfaction and preferences. We will then use the insights gained from the sentiment analysis to identify trends, patterns, and opportunities, which can be valuable for improving customer satisfaction.

2.1 Data pre-processing

After pre-processing data, the module *PC* from *pgmpy.estimator* were used intended to model the data and to obtain the direct acyclic diagram as can be seen in Fig. 2:

Firstly, we started by deleting any unnecessary columns from both datasets, the review dataset and the listings dataset implementing the algorithm presented in Fig. 1 and 2. We also decided that we should consider only reviews with the length higher than 30 characters because for the sentiment analysis longer reviews are more useful than shorter ones.

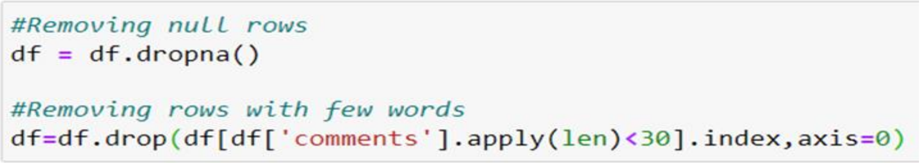


```
# drop unused columns
df=df.drop(['id','date', 'reviewer_id','reviewer_name'], axis=1)
df.head()
```

```
# drop unused columns

df_listing=df_listing.drop(['host_id','neighbourhood_group',
                            'latitude','longitude','reviews_per_month',
                            'calculated_host_listings_count',
                            'number_of_reviews_ltm','license'], axis=1)
```

Fig. 1. Python code used to delete unused columns.

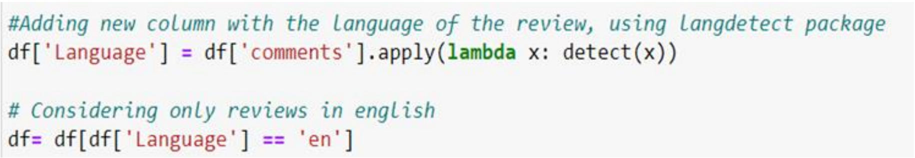


```
#Removing null rows
df = df.dropna()

#Removing rows with few words
df=df.drop(df[df['comments'].apply(len)<30].index,axis=0)
```

Fig. 2. Python code used to delete unused rows.

Considering that our data is real and there are reviews from several languages, in order to simplify the computational effort and reduce the noise in data, we decided to consider only reviews in English. The function *langdetect* from *detect* package was used to detect the language from each review. Firstly, the DataFrame available at http://insideairbnb.com/ has 162.916 rows and after dropping the unused rows the DataFrame has 144969 rows, which is a huge number of reviews to analyze.



```
#Adding new column with the language of the review, using langdetect package
df['Language'] = df['comments'].apply(lambda x: detect(x))

# Considering only reviews in english
df= df[df['Language'] == 'en']
```

Fig. 3. Python code to consider only reviews in English.

We used the NLTK library to manipulate the data in order to tokenize, remove the stop words and count the frequency of each word on each review, with the functions word_tokenize, FreqDist, stopwords. Finished the Data pre-processing, we perform the sentiment analysis using the nltk.sentiment.vader.SentimentIntensityAnalyzer, it is a pre-trained model for sentiment analysis in English text being part of the VADER (Valence Aware Dictionary and sentiment Reasoner) module from NLTK library.

The VADER Sentiment Analyzer is designed for sentiment analysis in social media context, where traditional sentiment analysis models may struggle because of the informal language used. VADER is trained on a vast amount of social media data and uses a combination of lexical and grammatical heuristics to estimate the sentiment intensity of a given text. (Keita,2022)

For the following analysis, we consider the compound score given by polarity_scores() method, which represents an aggregated sentiment intensity ranging from -1 (extremely negative) to 1 (extremely positive). To have a better view of the result, a histogram and a boxplot were made, as can be seen in Fig. 4 and Fig. 5, respectively. The figures showed that the mean of the compound from we can see that from the reviews tend to be positive, which is good information for Airbnb, but it creates some difficulties in our analysis, more specifically how can we detect the similar sentiment scores once it has a very low variance and tends to be very positive.

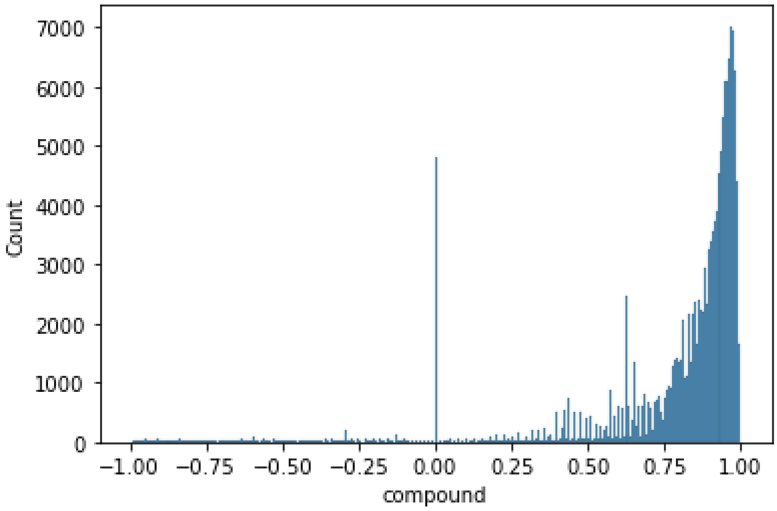


Fig. 4. Histogram of the distribution of "compound"

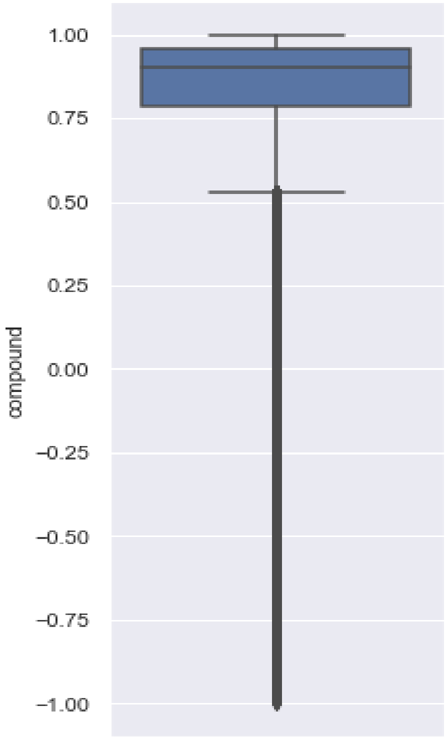
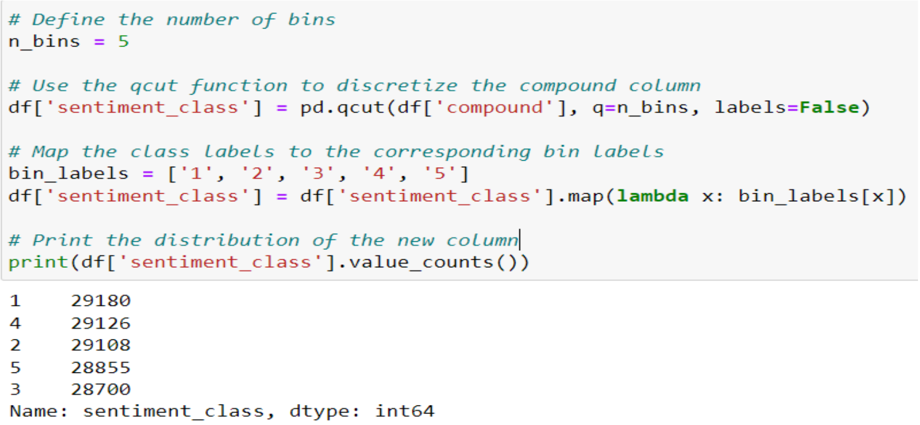


Fig. 5. Boxplot of the distribution of "compound"

To avoid this similarity between the compound of the reviews, we decided to discretize the “compound” value, creating a new column named “sentiment_class” using the qcut function from Pandas library. The function computes the quantiles based on the distribution of the data, then splits the data in the number of bins defined where each bin will have approximately the same number of observations. The code implemented and the distribution after implementation, can be seen in Fig. 6 and 7, respectively.



```
# Define the number of bins
n_bins = 5

# Use the qcut function to discretize the compound column
df['sentiment_class'] = pd.qcut(df['compound'], q=n_bins, labels=False)

# Map the class labels to the corresponding bin labels
bin_labels = ['1', '2', '3', '4', '5']
df['sentiment_class'] = df['sentiment_class'].map(lambda x: bin_labels[x])

# Print the distribution of the new column
print(df['sentiment_class'].value_counts())
```

```
1    29180
4    29126
2    29108
5    28855
3    28700
Name: sentiment_class, dtype: int64
```

Fig. 6. Python code to use *qcut* function to discretize the “compound”

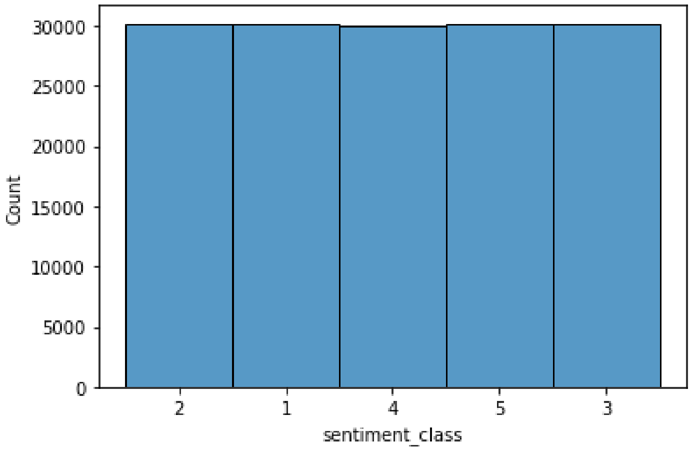


Fig. 7. Histogram of the distribution of “sentiment_class” generated by *qcut* function

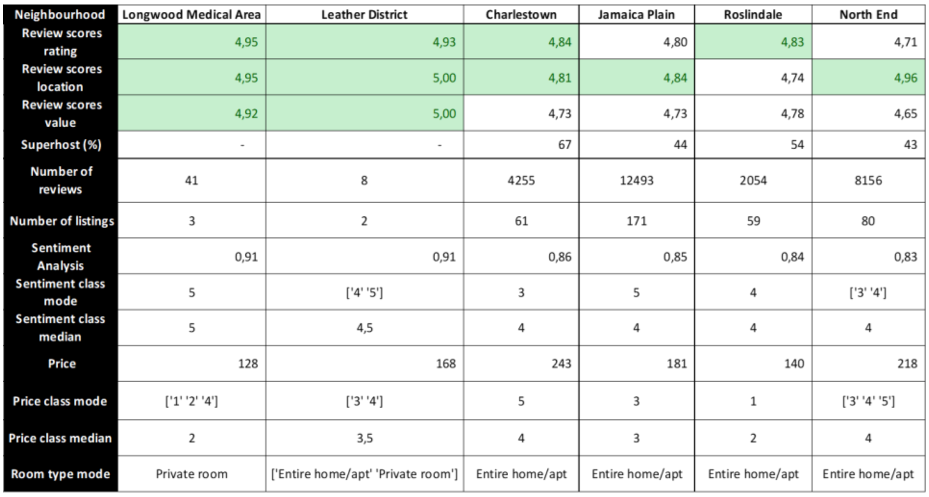
3 Result and Discussion

In this section, the sentimental analyses made in previously will be the based for further analysis in order to estimate cause and effect analysis.

3.1 Grouped analysis

I. Neighbourhood:

From the sentimental analysis, the mean of the compound taken for each announcement was correlated with other information provided from the dataset. The table presented in Fig. 8 includes the top 6 neighborhoods in Boston with the best sentiment analysis results, as there are 25 different neighborhoods. The full table is available at the Appendix.



Neighbourhood	Longwood Medical Area	Leather District	Charlestown	Jamaica Plain	Roslindale	North End
Review scores rating	4,95	4,93	4,84	4,80	4,83	4,71
Review scores location	4,95	5,00	4,81	4,84	4,74	4,96
Review scores value	4,92	5,00	4,73	4,73	4,78	4,65
Superhost (%)	-	-	67	44	54	43
Number of reviews	41	8	4255	12493	2054	8156
Number of listings	3	2	61	171	59	80
Sentiment Analysis	0,91	0,91	0,86	0,85	0,84	0,83
Sentiment class mode	5	['4' '5']	3	5	4	['3' '4']
Sentiment class median	5	4,5	4	4	4	4
Price	128	168	243	181	140	218
Price class mode	['1' '2' '4']	['3' '4']	5	3	1	['3' '4' '5']
Price class median	2	3,5	4	3	2	4
Room type mode	Private room	['Entire home/apt' 'Private room']	Entire home/apt	Entire home/apt	Entire home/apt	Entire home/apt

Fig. 8. Neighbourhood top 6 table

a. Review Scores:

The table displays the average review scores for different aspects such as rating (general), location, and value, this data is available on http://insideairbnb.com/. The review scores range from 4.36 to 4.95, indicating generally positive feedback.

We can see that the top 2 neighborhoods, Longwood Medical Area and Leather District, have considerably lower numbers of reviews and listings so it is necessary to be careful to conclude that these neighborhoods are really the best in Boston. However, comparing the top 6 by the sentiment analysis and the top 6 by Airbnb data, the results are near, with only one change: West Roxbury does not appear, giving the place to North End which has the best location score (excluding the Leather District).

b. Superhost (%):

Indicates the percentage of listings that have a superhost on each neighborhood, the relation of this column with the sentiment analysis was not very clear on the table, but we can see the trend that neighborhoods with lower than 30% of its listings with a superhost are on the lower positions.

c. Sentiment Analysis:

The sentiment analysis column represents the mean of the sentiment score (compound), the range from 0.71 to 0.91, indicating a generally positive sentiment. This column was used to sort the table and show the top 6 neighborhoods.

The table also displays the sentiment class mode and median for each neighborhood, which are important central metrics to understand the distribution of the sentiment analysis at each neighborhood. Looking at the median that is always higher or equal to the class 4, these 6 neighborhoods have the reviews sentiment above the average.

d. Price:

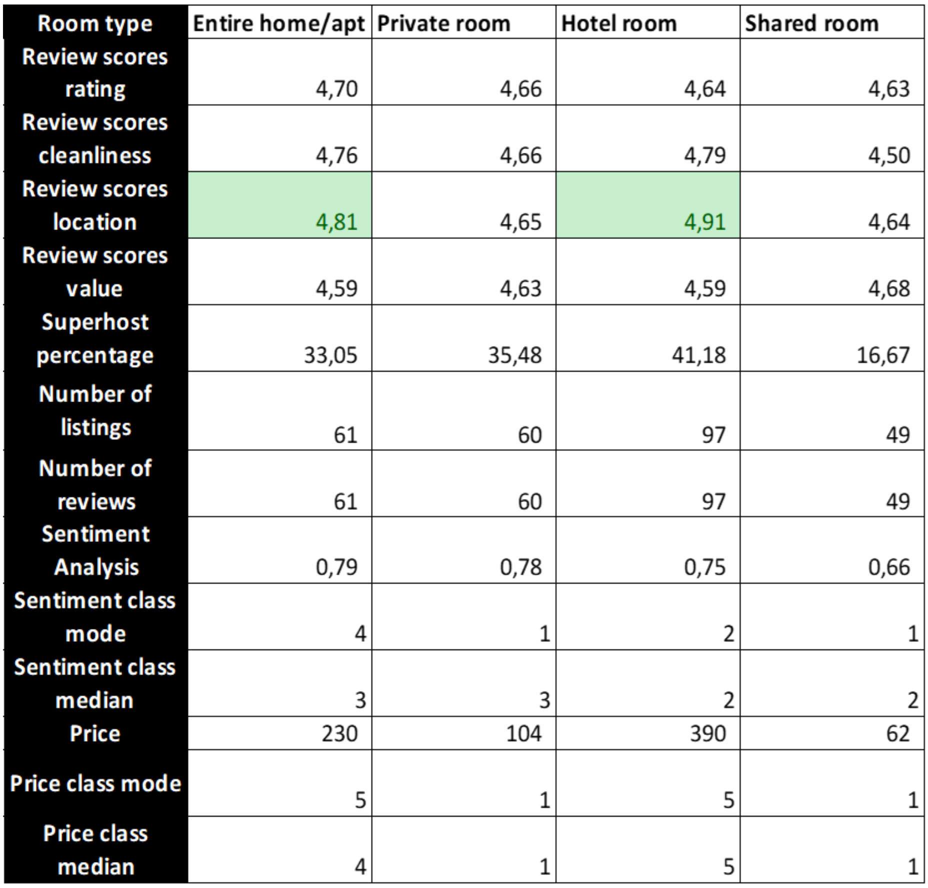
The table displays the average price of listings in each neighborhood, the range from $104 to $283. We can see the high variance of mean price in both, the best and the worst, showing that different customers have different preferences considering the relation between neighborhood and price. The same lack of pattern can be seen at the price class mode and median.

e. Room type mode:

This column represents the most frequent room type in each neighborhood, along the table it includes only "Entire home/apt" and "Private room".

II. Room type:

The table presented in Fig. 9 aims to correlate the room type with the sentimental score (amount) with other scores presented in the dataset intended to highlight the relationship between them.



Room type	Entire home/apt	Private room	Hotel room	Shared room
Review scores rating	4,70	4,66	4,64	4,63
Review scores cleanliness	4,76	4,66	4,79	4,50
Review scores location	4,81	4,65	4,91	4,64
Review scores value	4,59	4,63	4,59	4,68
Superhost percentage	33,05	35,48	41,18	16,67
Number of listings	61	60	97	49
Number of reviews	61	60	97	49
Sentiment Analysis	0,79	0,78	0,75	0,66
Sentiment class mode	4	1	2	1
Sentiment class median	3	3	2	2
Price	230	104	390	62
Price class mode	5	1	5	1
Price class median	4	1	5	1

Fig. 9. Room type scores

a. Review Scores:

The table in Fig. 9 displays the average review scores for different aspects such as rating, location, and value, this data is available on http://insideairbnb.com/. Again, the result is satisfactory since the sentiment analysis by room type presents the same order as the rating scores given by the guests and provided by airbnb.

The location score has the highest difference between the different room types, that the “Hotel rooms” and the "Entire home/apt" have considerably higher values than the other types.

b. Superhost Percentage:

The "Hotel room" type has a higher proportion of superhosts compared to other room types.

c. Sentiment Analysis:

The sentiment analysis score is highest for the "Entire home/apt" room type (0.79), followed very closely by "Private room" (0.78), then the "Hotel room" (0.75), and "Shared room" (0.66). The mode of the sentiment class is 4 for "Entire home/apt," 1 for "Private room" and "Shared room," and 2 for "Hotel room." The median of the sentiment class is 3 for "Entire home/apt" and "Private room," and 2 for "Hotel room" and "Shared room."

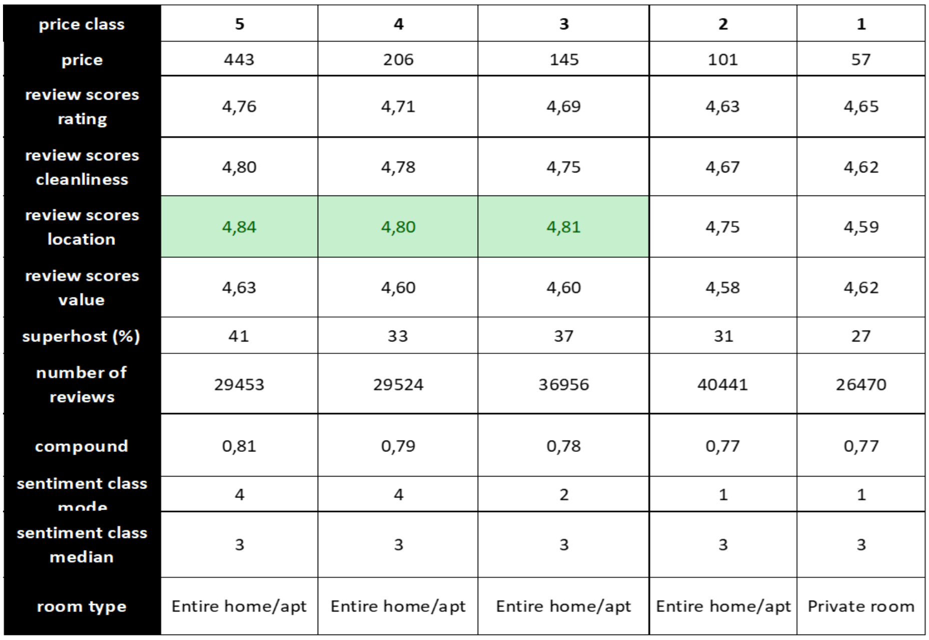
This indicates that listings categorized as "Entire home/apt" and "Private room" generally have more positive sentiment compared to "Hotel room" and "Shared room".

d. Price:

The average price is highest for the "Hotel room" type ($390), followed by "Entire home/apt" ($230), "Private room" ($104), and "Shared room" ($62). The mode and median price classes are generally higher for the "Hotel room" type compared to other room types. This indicates that listings categorized as "Hotel room" generally are more expensive than other room types.

III. Price class:

The table shows different price classes (3, 1, 0, 4, 2) and their corresponding attributes. This variable “price_class” was created by us, using the qbin function from the pandas package, just like the “sentiment_class” presented before.



price class	5	4	3	2	1
price	443	206	145	101	57
review scores rating	4,76	4,71	4,69	4,63	4,65
review scores cleanliness	4,80	4,78	4,75	4,67	4,62
review scores location	4,84	4,80	4,81	4,75	4,59
review scores value	4,63	4,60	4,60	4,58	4,62
superhost (%)	41	33	37	31	27
number of reviews	29453	29524	36956	40441	26470
compound	0,81	0,79	0,78	0,77	0,77
sentiment class mode	4	4	2	1	1
sentiment class median	3	3	3	3	3
room type	Entire home/apt	Entire home/apt	Entire home/apt	Entire home/apt	Private room

Fig. 10. Price class table

a. Review Scores:

The table displays the average review scores for different aspects such as rating, location, and value, this data is available on http://insideairbnb.com/. One more time the result is satisfactory since the sentiment analysis by each price class presents almost the same order as the rating scores given by the guests and provided by airbnb.

b. Superhost Percentage

The sentiment analysis score does not show a clear correlation with the superhost percentage in this table. The superhost percentage remains relatively consistent across both sentiment analysis scores and price classes.

c. Sentiment analysis:

This column indicates the sentiment class based on the sentiment analysis score. The sentiment classes range from 1 to 5, with higher values representing more positive sentiment. The sentiment class median in this table is constant (3) and the compound value is very similar between different price classes, it doesn't provide much variation for comparison. Leading us to conclude that there are happy customers in all of the price classes.

d. Room type

This column indicates the type of room most seen by each sentiment class, all classes "Entire home/apt" or "Private room".

3.2 Correlation analysis

The correlation Analysis as a statistical technique which allows to determine the strength and the direction of the relationship between two variables, taking us to the relationship identification between the accommodation average sentiment analysis as *compound_listings* and other characteristics of the accommodations such as the number of reviews, the quantity of beds and bedrooms and the price, among others.

The correlation coefficient ranges from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.It is important to note that correlation does not imply causation, but it provides a measure of the linear association between variables

An important characteristic for the correlation analysis is the data validation, for instance if our sentiment analysis is correct, each listing compound average should have a very high positive correlation with the different review scores for each accommodation available from Airbnb. Indeed, it validates our analysis once the “compound_listings” more related variables are: 'review_scores_rating', review_scores_accuracy', 'review_scores_cleanliness', 'review_scores_checkin', 'review_scores_communication', 'review_scores_location' and ‘'review_scores_value'.

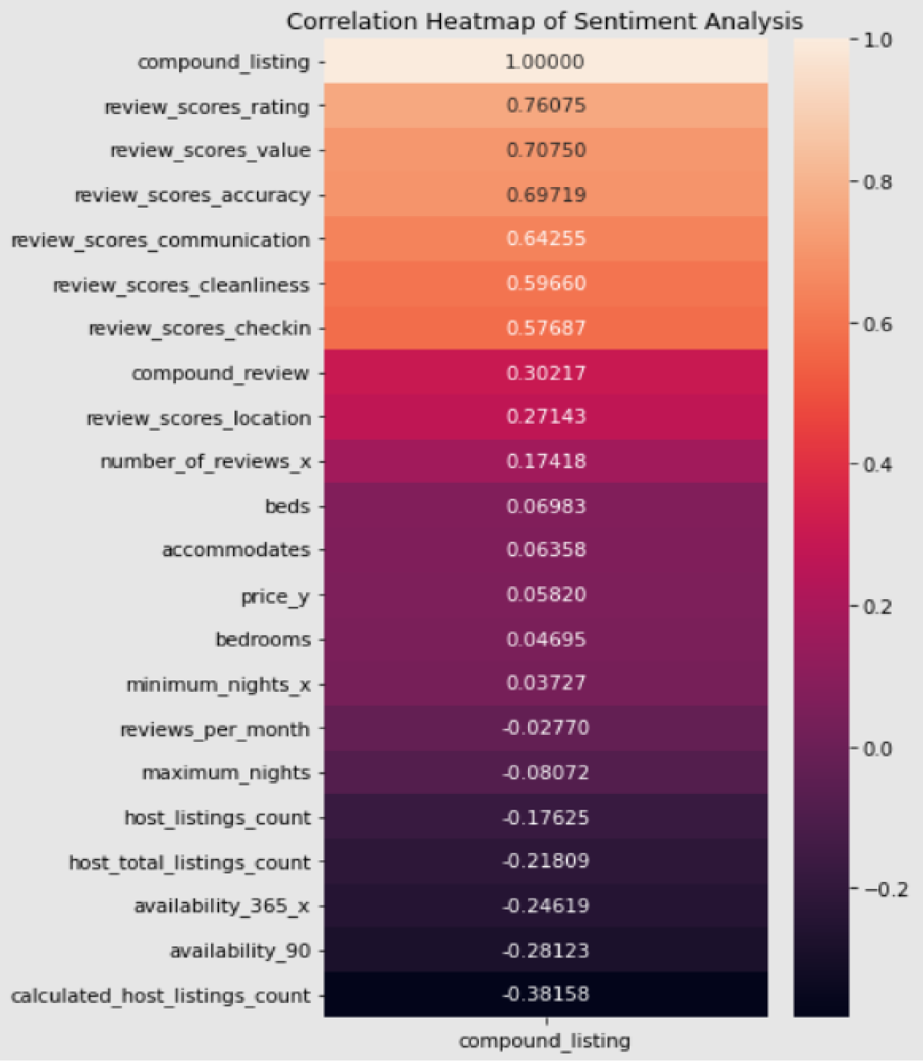


Fig. 12. Correlation heatmap

Correlation analysis also provides useful quantitative information for decision-making. It assists in identifying factors that should be considered when making decisions, for instance, deciding some characteristics when investing in an accommodation to rent at Airbnb.

Number of Reviews (0.174184): There is a positive correlation between "compound_listing" and the number of reviews. It suggests accommodations with a greater number of reviews tend to have higher sentiment analysis scores.

Beds (0.069833), Accommodates (0.063582), Price (0.058203) and Bedrooms (0.046954): These variables show weak positive correlations with "compound_listing." It implies that there may be a slight tendency for accommodations with more beds, which accommodate more guests, also higher prices and more bedrooms, tend to have better sentiment analysis scores.

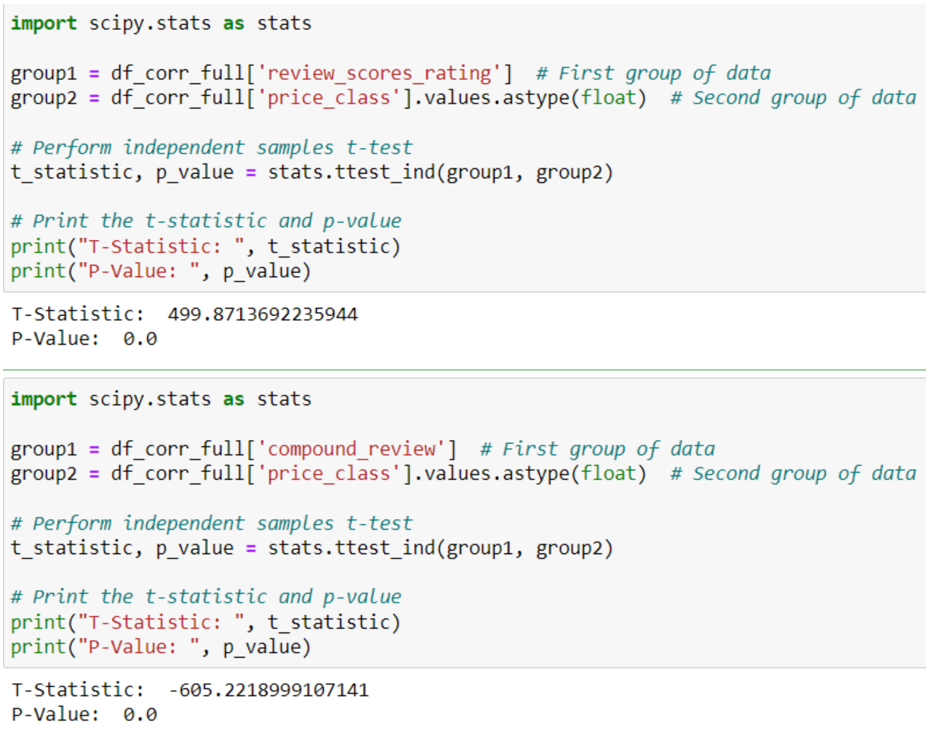
Calculated Host Listings Count (-0.381577): This variable exhibit moderate negative correlations with "compound_listing." It suggests that accommodations with higher sentiment analysis scores tend to have a lower number of host listings. In other words, hosts with a lot of listings at Airbnb tend to have a lower score.

Availability 365 (-0.246193) and Availability 90 (-0.281227): There is a moderate negative correlation with “compound_listing”. It makes sense, as the best accommodation has less availability once the demand for the best listings is higher than the worst ones.

3.3 Hypothesis testing

We perform the t-test as a statistical test in order to determine if there is a significant difference between the means of two groups (variables). It helps us compare whether the observed difference in sample means is likely to represent a true difference in the population means or if it is just due to random chance. To illustrate what can be done by hypothesis testing in context of this project, we performed the t test for two variables: the "review_scores_rating" and”compound_listings”, the result for both is similar what indeed confirm the power of our sentiment analysis and the result will be presented once for both variables.

To perform the statistical testing we used the stats package from scipy package, it generates two results, the T-statistics value and the P-value. The code used is presented below:



```
import scipy.stats as stats

group1 = df_corr_full['review_scores_rating']  # First group of data
group2 = df_corr_full['price_class'].values.astype(float)  # Second group of data

# Perform independent samples t-test
t_statistic, p_value = stats.ttest_ind(group1, group2)

# Print the t-statistic and p-value
print("T-Statistic: ", t_statistic)
print("P-Value: ", p_value)
```

```
T-Statistic:  499.8713692235944
P-Value:  0.0
```

```
import scipy.stats as stats

group1 = df_corr_full['compound_review']  # First group of data
group2 = df_corr_full['price_class'].values.astype(float)  # Second group of data

# Perform independent samples t-test
t_statistic, p_value = stats.ttest_ind(group1, group2)

# Print the t-statistic and p-value
print("T-Statistic: ", t_statistic)
print("P-Value: ", p_value)
```

```
T-Statistic:  -605.2218999107141
P-Value:  0.0
```

Fig. 13. Python code to perform t-test and the test result.

The t-statistic measures the difference between the means of the two groups relative to the variation within each group. It indicates the magnitude of the difference between the groups taking into account the sample size and variability. In both cases, the t-statistic value is very high, suggesting a substantial difference between the "price_class" and both "review_scores_rating" and "compound_listings".

The p-value represents the probability of observing such extreme or more significant differences between the groups if there were no true difference in the population. A smaller p-value suggests stronger evidence against the null hypothesis (the hypothesis that the two groups are not significantly different). In this case, the p-value is 0.0, which is below the conventional significance level of 0.05.

This indicates strong evidence to reject the null hypothesis and suggests that there is a significant difference. The t-statistic value indicates a large difference between the means of the groups, and the extremely low p-value suggests strong evidence against the null hypothesis.

4 Conclusion

The current work proposed to make a sentimental analysis in comment from Airbnb rent room dataset. Besides that, a correlation between some information like room type, review score or price was possible to make clear the room characteristics with client satisfaction. Considering the room type, although the review score for the hotel room had the highest value, using sentiment analysis was possible to identify to see that the entire home had a better score, followed by the private room.

In the correlation analysis, the strength and direction of the relationship between the sentiment analysis scores and other accommodation characteristics are examined.

Finally, hypothesis testing using the independent samples t-test is performed to determine if there is a significant difference between the means of “review_scores_rating" and "price_class" and the mean of the sentiment score, “compound”. The results of the t-test validate the power of the sentiment analysis.

Overall, the project demonstrates the application of text mining and sentiment analysis techniques to extract insights from unstructured text data and improve customer satisfaction in the Airbnb context.

References

1. E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," in IEEE Computational Intelligence Magazine, vol. 9, no. 2, pp. 48-57, May 2014, doi: 10.1109/MCI.2014.2307227. Retrieved from: https://ieeexplore.ieee.org/abstract/document/6786458

2.Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc. Retrieved from: https://surface.syr.edu/cgi/viewcontent.cgi?article=1043&context=istpub

3. Slides for the Text Mining chapter 1 of the course Data Mining 2. Retrieved from Moodle: https://moodle.up.pt/pluginfile.php/132044/mod_resource/content/1/1-TextMining.pdf

4. Slides for the Text Mining chapter 2 of the course Data Mining 2. Retrieved from Moodle: https://moodle.up.pt/pluginfile.php/132045/mod_resource/content/2/2-TextMining.pdf

5. Slides for the Text Mining chapter 3 of the course Data Mining 2. Retrieved from Moodle: https://moodle.up.pt/pluginfile.php/132046/mod_resource/content/1/3-TextMining.pdf

6. McKinney W, others. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. 2010. p. 51–6. Retrieved from: https://pandas.pydata.org/

7. Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc. Retrieved from: https://www.nltk.org/

8. Pipis, G. How To Run Sentiment Analysis In Python Using VADER. City, 2020. Retrieved from: https://predictivehacks.com/how-to-run-sentiment-analysis-in-python-using-vader/

9. Kirilenko, A. P., Stepchenkova, S. O., Kim, H., & Li, X. (Robert). (2018). Automated Sentiment Analysis in Tourism:

Comparison of Approaches. Journal of Travel Research, 57(8), 1012–1025. https://doi.org/10.1177/0047287517729757. Retrieved from: https://journals.sagepub.com/doi/full/10.1177/0047287517729757?journalCode=jtrb

10. Keita, Z. Social Media Sentiment Analysis In Python With VADER — No Training Required! , City, 2022. Retrieved from: https://towardsdatascience.com/social-media-sentiment-analysis-in-python-with-vader-no-training-required-4bc6a21e87b8

11. Pauli Virtanen, others, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272. Retrieved from: https://scipy.org/