Airbnb Listings in London, United Kingdom

March 17, 2022
Barcelona

Demi Vinke
Fabrizio Rocco
Vincent Standler
Vivien Laurent
Damla Yalçın

OUR TEAM MEMBERS

Demi Vinke

Fabrizio Rocco

Vincent Stadler

Vivien Laurent

Damla Yalçın

esade
RAMON LLULL UNIVERSITY

TABLE OF CONTENT

esade
RAMON LLULL UNIVERSITY

PROBLEM AND MOTIVATION

PROBLEM

There are a growing number of apartments on Airbnb, and it can be very difficult for a new landlord to understand:

- What does it take to stand out?
- How to price correctly?
- How to know what guests appreciate regarding service?

WHAT WE OFFER…

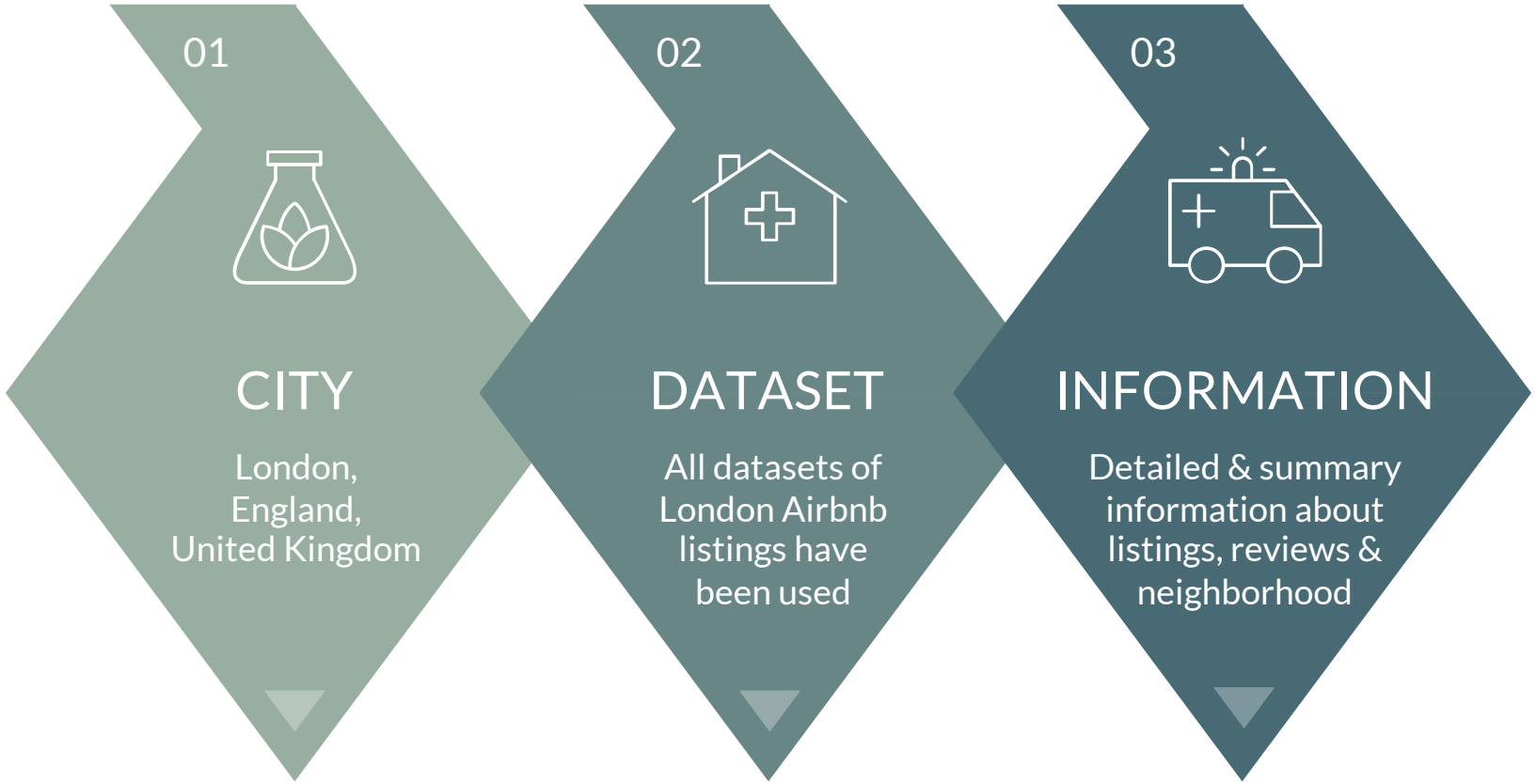
A platform that landlords can subscribe to for three main features:

1) Getting similar apartment suggestions
2) Getting a pricing range based on similar apartments
3) Getting an idea of what services are valued by customers

MOTIVATION

- Interesting to design a platform that can resolve these problems
- Creating a platform that Airbnb could offer as a paid service
- Opportunity to use multiple AI techniques

ABOUT THE DATA

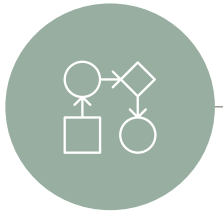


esade
RAMON LLULL UNIVERSITY

LISTINGS RECOMMENDATION

Understanding the Problem

- For a new landlord in Airbnb platform, it would be difficult to have an idea of similar apartments. Therefore, our aim is to provide a recommender system that can show the 5 most similar apartments to understand how the competitors are doing.



Understanding the Data

- Airbnb London listings data contains 66641 instances and 74 columns.
- 24 of features have 'float', 17 of features have 'int64', and 33 of them have 'object' type.
- 3 variables contain roughly 32000, and 8 variables contain approximately 18000 missing values

Preprocessing the Data

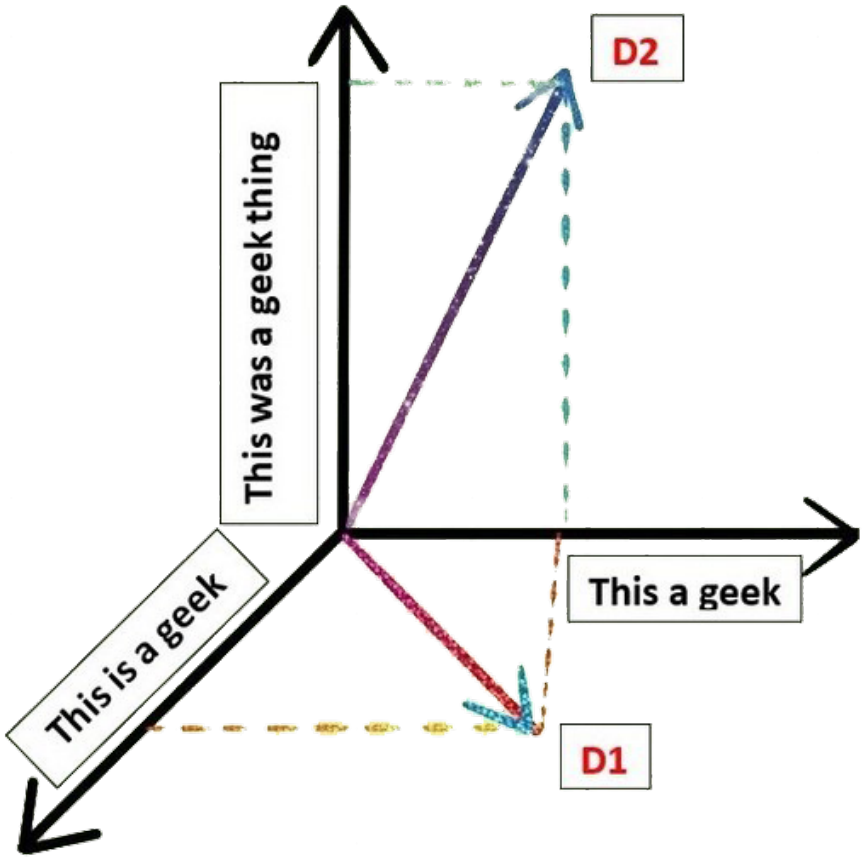
- Several regex functions (lowercase, alphanumerical, amenities)
- Only 6 columns are necessary for this use case

ALGORITHMS SPECIFICITIES: COSINE SIMILARITY

- Cosine Similarity measures the cosine angle between two vectors projected in a multi-dimensional space

- In the **sklearn** module, there is a built-in function called **cosine_similarity()** to calculate the cosine similarity.

- The smaller the angle the higher the cosine similarity
- When the vectors are pointing in the same direction it means that they are similar.



esade
RAMON LLULL UNIVERSITY

THE RECOMMENDER FUNCTION

```
def recommend (data, query, matrix, vect):
    user_query = vect.transform([query])
    similarity = cosine_similarity(user_query, matrix)
    top_5 = np.argsort(similarity[0])[-5:]
    best_res = np.argmax(similarity[0])
    return data.loc[[top_5[4], top_5[3], top_5[2], top_5[1], top_5[0]]]
```

Vectorize the query inputted by the user

Cosine similarity between user's vector and matrix

Sort the 5 most similar listings

Return the top 5 in the web app

RECOMMENDATION IN ACTION

Data downloaded ✅

Rows	Columns	City
66641	57	London, UK

Look for similar apartments

Apartment close to London Eye with 3 bedrooms and balcony

Recommend

Top 5 similar apartments

	listing_url	name	description	price
19605	https://www.airbnb.com/…	next london eye apartment	stunning apartment in th…	235.(
40250	https://www.airbnb.com/…	modern luxury chelsea a…	modern luxury bedrooms…	140.(
42708	https://www.airbnb.com/…	london eye apartment he…	beautiful brand new apar…	138.(
42977	https://www.airbnb.com/…	modern contemporary be…	modern luxury bedrooms…	160.(
61377	https://www.airbnb.com/…	london eye apartment he…	beautiful brand new apar…	106.(

Next London Eye Apartment

2 reviews · London, England, United Kingdom Share Save

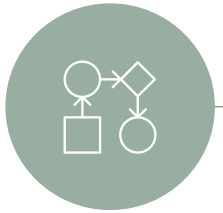
London Eye Apartment (Heart of Central London)

★ 4.71 · 17 reviews · Greater London, United Kingdom Share Save

PRICE PREDICTION

Understanding the Problem

- For a new landlord in Airbnb platform, it would be difficult to predict price correctly. Therefore, our aim is to provide a pricing range to a new landlord based on similar apartments.
- Target Variable: Price (Regression Problem)



Understanding the Data

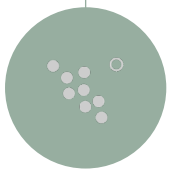
- Airbnb London listings data contains 66641 instances and 74 columns.
- 24 of features have 'float', 17 of features have 'int64', and 33 of them have 'object' type.
- 3 variables contain roughly 32000, and 8 variables contain approximately 18000 missing values

Preprocessing the Data

- Depending on the correlation, having missing values and data type:
- 41 columns have been removed.
- All categorical variables have been either one-hot or label-hot encoded manually.

PREPROCESSING STEPS

Removing Outliers

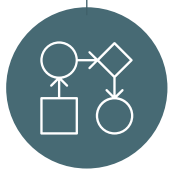


We removed the outliers from price (greater than or equal to 1195)

Filling missing values

We filled 8 variables with the median and 2 with ratios derived from other variables

Cleaning Numerical Features



We removed the symbols from the variables ($, %)

Encoding Categorical Features

Depending on unique values and mean of the features, we have used one-hot and label-encoding

Variance Threshold

We used a feature selector algorithm which only looks at the predictors (X), not the prediction target (y), and removes all low-variance features

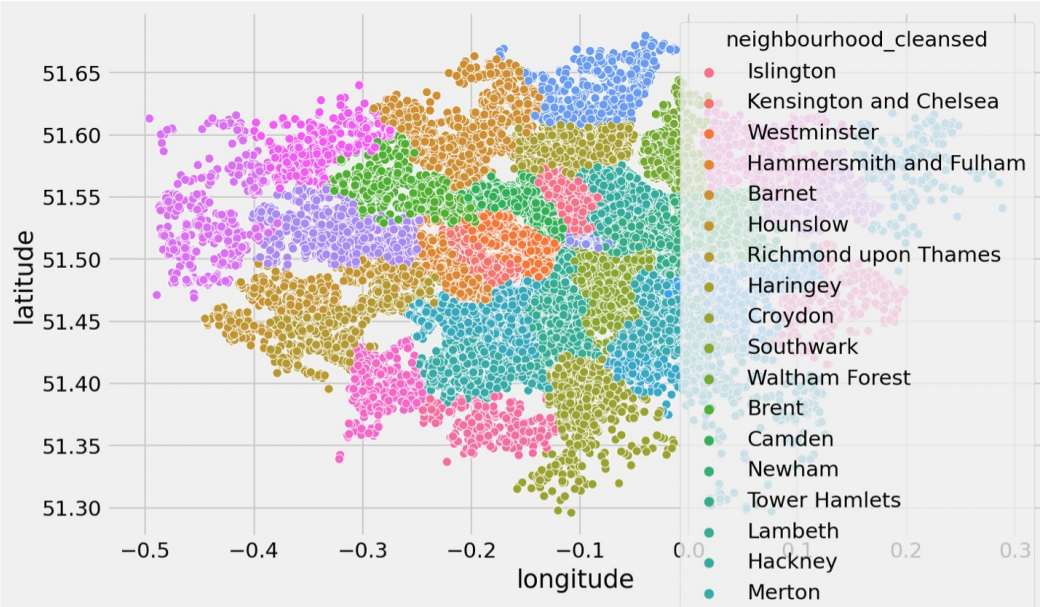
Variance Cutoff: 80% same value

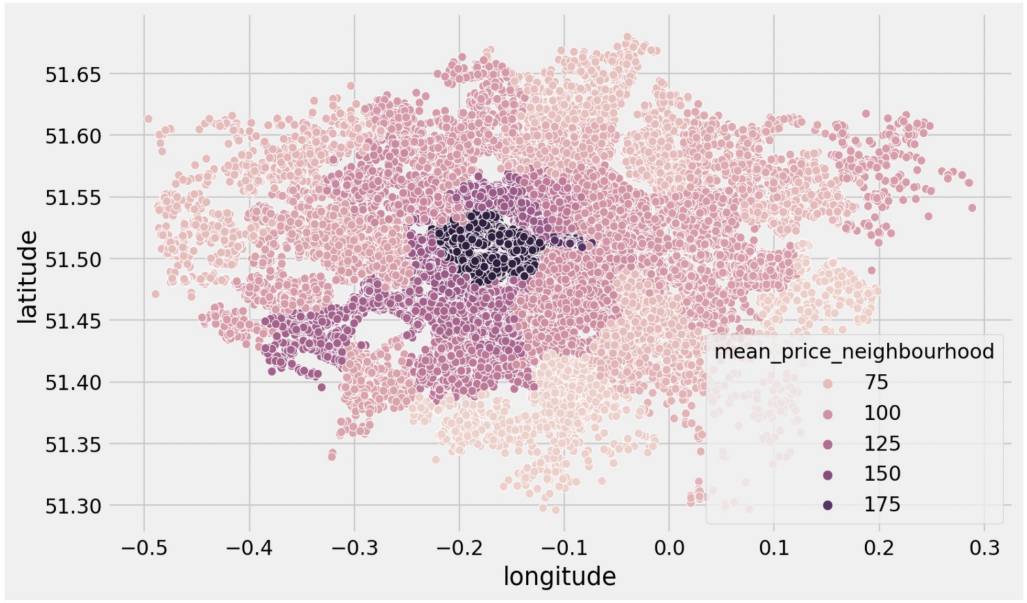
PREPROCESSING AMENITIES

Cleaning symbols from amenities and storing them in a new variable called **'amenities_list'**	→	Length of amenities_list is 65875
Dropping duplicates and storing them in a new variable called **'amenities'**	→	Length of amenities is 2516
Removing all columns where no more than 13000 houses have that amenity (20% of the data)	→	Number of remaining amenities is 28

PREPROCESSING NEIGHBORHOOD

- Neighborhood data of London was problematic in terms of the numbers and differences of neighborhoods.
- We have grouped the neighborhood data according to average price and label-encoded depending on the new price range we obtained.
- The most expensive neighborhood depending on the average price values are located at the center of London.



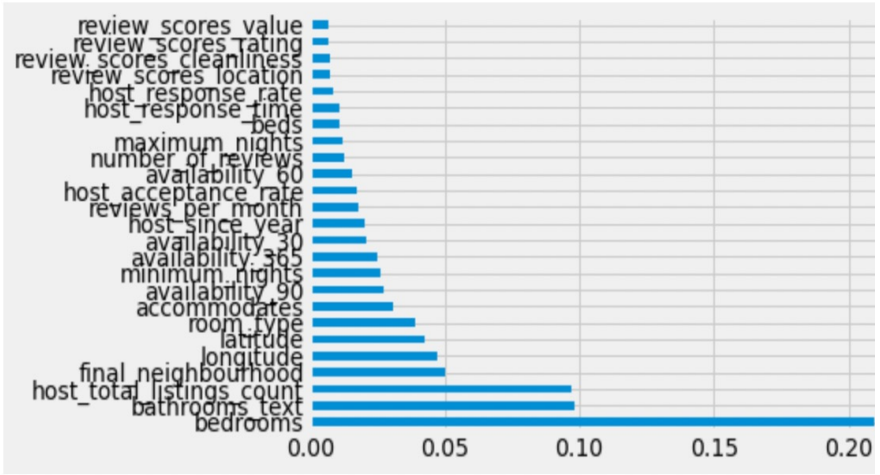


TRAINING MACHINE LEARNING MODELS

- We have split the data according to 80:10:10 ratios to obtain train, validation, and test splits.
- We trained a Linear Regression and a Decision Tree Classifier as initial models.
- These models could not explain much of the variance

	R Squared
Linear Regression	0.4524
Decision Tree Classifier	0.0519

- We have selected the top 25 features with the largest feature importance and trained a Random Forest Regressor and a Gradient Boosting Regressor.



esade RAMON LLULL UNIVERSITY

TRAINING MACHINE LEARNING MODELS

The five situations in which we trained models:

1. Running a base model
2. Finding the best parameters for this model
3. Creating two new features
4. Removing all but the 25 most important features (two features not included)
5. Using the created features, as well as removing features

	Base		Grid search		Feature engineering		Feature removal		Feature removal and engineering	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
Random Forest Regressor	0.704	34.245	0.644	37.540	0.632	38.488	0.665	37.260	-	-
Gradient Boosting Regressor	0.659	36.762	0.739	32.192	0.737	32.567	0.681	36.454	0.669	36.935

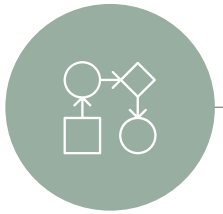
THE BEST MODEL

Best Model	Test set result		Validation result	
	R^2	RMSE	R^2	RMSE
GBR	0.716	34.141	0.724	34.678

UNSUPERVISED LEARNING ALGORITHM

Understanding the Problem

- For a new landlord in Airbnb platform, it would be difficult to understand the key patterns in hosts reviews and comments. Our aim is to use Sentiment Analysis to show which services are valued.
- Outcome: word cloud for negative and positive comments, examples of positive and negative reviews.



Understanding the Data

- We have used 'reviews' and 'listings' data as well as 'neighbourhood' to as input data.
- There are almost 2 million reviews in the data set.

Preprocessing the Data

- We have detected missing values and systematically non-useful comments and removed them.
- 14375 features contained 'The host canceled this reservation.'
- We have also dropped dashes, numeric, escape characters and comments that only contain '.'

PREPROCESSING STEPS

Removing missing values and non-interpretable ones

We dropped rows without comments and where host cancelled

Text Cleaning

We removed the commonly used ‘stop words’ using NLTK (Natural Language Toolkit)

Language Cleaning

We kept only English comments using langdetect (language detection library)

Visualization & Sentiment Analysis

We calculated the polarity of comments and visualized the analysis on an interactive map: Word Cloud

SENTIMENT ANALYSIS – ESTIMATING POLARITY WITH *VADER*

To analyze the sentiment of comments and extract relevant information from them we needed a measure to score how positive or negative they are.

To achieve this, we used the lexicon and rule-based sentiment analysis tool *VADER* (Valence Aware Dictionary and Sentiment Reasoner).

VADER determines a so-called polarity score for every comment, which ranges from -1 to 1 (negative values correspond to negative comments, 0 to neutral ones and positive values to positive scores).

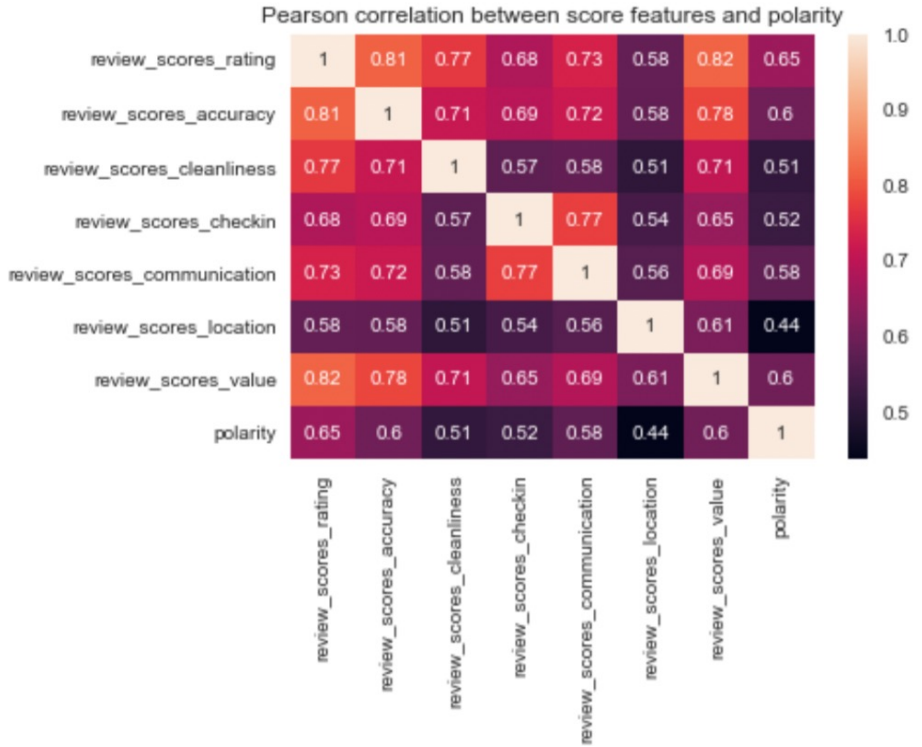
VADER adds valence to words capitalized, with punctuation (compound score increases with number of exclamation points!), with degree modifiers (e.g. very good vs good), emojis, and handles conjunctions.

SENTIMENT ANALYSIS – CORRELATION BETWEEN FEATURES AND POLARITY

- We have checked the correlation between polarity feature and other rating scores.
- We observed that there is a significant relation between all of them.

- **The highest correlation** was observed with the overall rating (review_scores_rating).

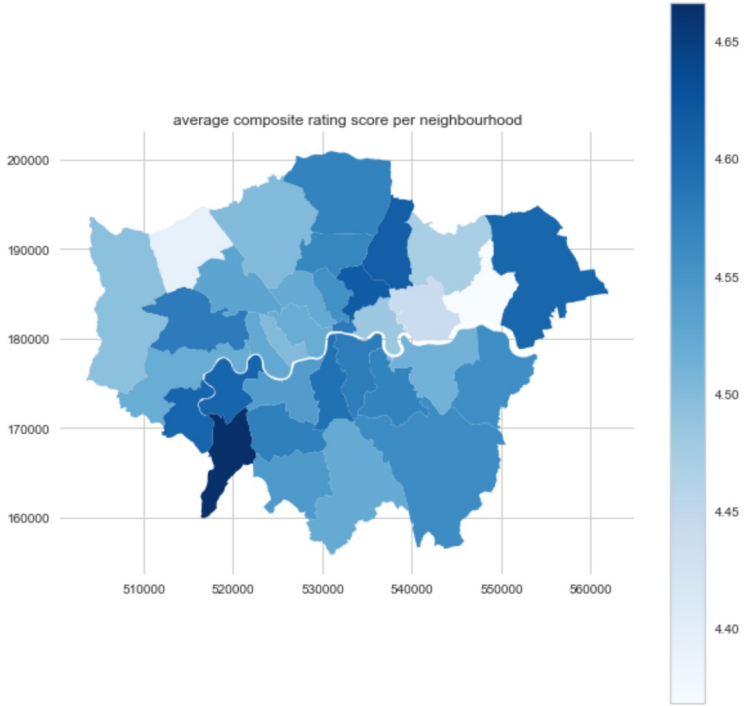
- **The lowest correlation** was between polarity and reviews_score_location.
- This means that location does not play a huge role in terms of guest reviews during their stay in an Airbnb apartment in London.

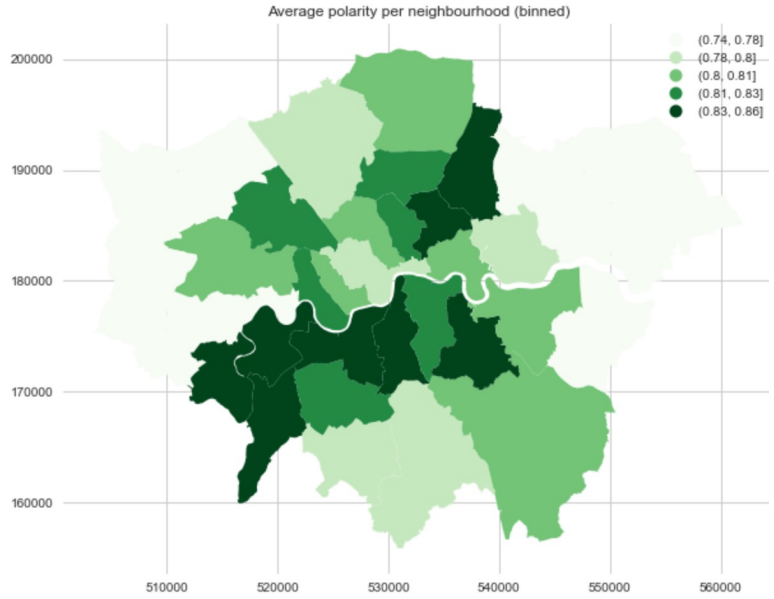


esade RAMON LLULL UNIVERSITY

EXPLORING NEIGHBORHOODS

- We have looked at is the **distribution of ratings and polarity scores across different neighborhoods** in London by considering the **average composite rating score** as well as the **average polarity per neighborhood:**





- The **average composite rating score and polarity scores are quite high.** Some neighborhoods tend to **perform better on all metrics** (**South-Western part of London**).

INVESTIGATING POSITIVE AND NEGATIVE COMMENTS

- We have used '**WordClouds**' to discover what visitors applaud or complain about:

POSITIVE COMMENTS

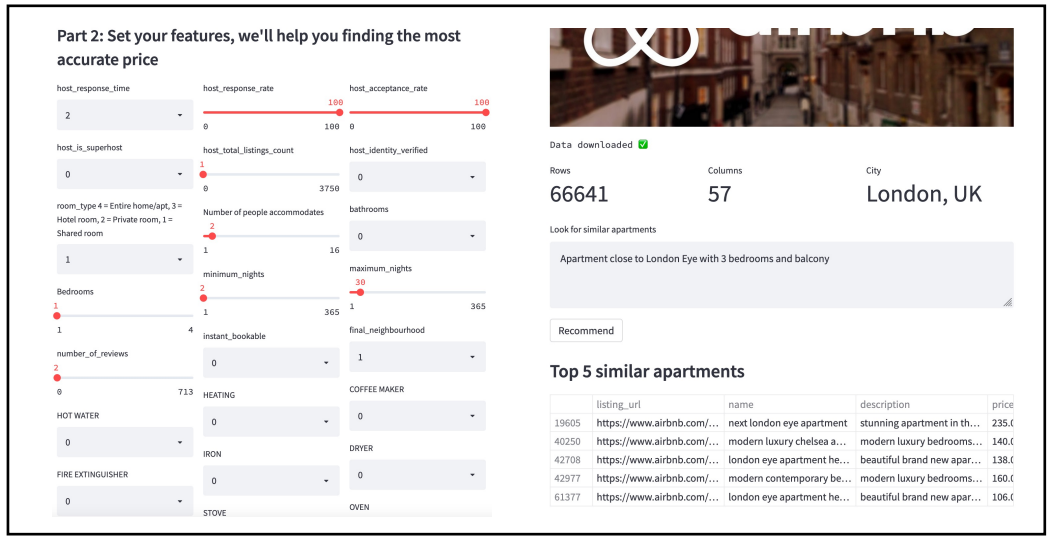
NEGATIVE COMMENTS

esade
RAMON LLULL UNIVERSITY

IMPACT ON AIRBNB'S BUSINESS MODEL AND REVENUE

'Airbnb Premium'

- Subscription for landlords
- Recommendation on pricing, advertisement and description
- Insights into what is valued by guests



esade
RAMON LLULL UNIVERSITY

THANK YOU!

Questions?

APPENDIX

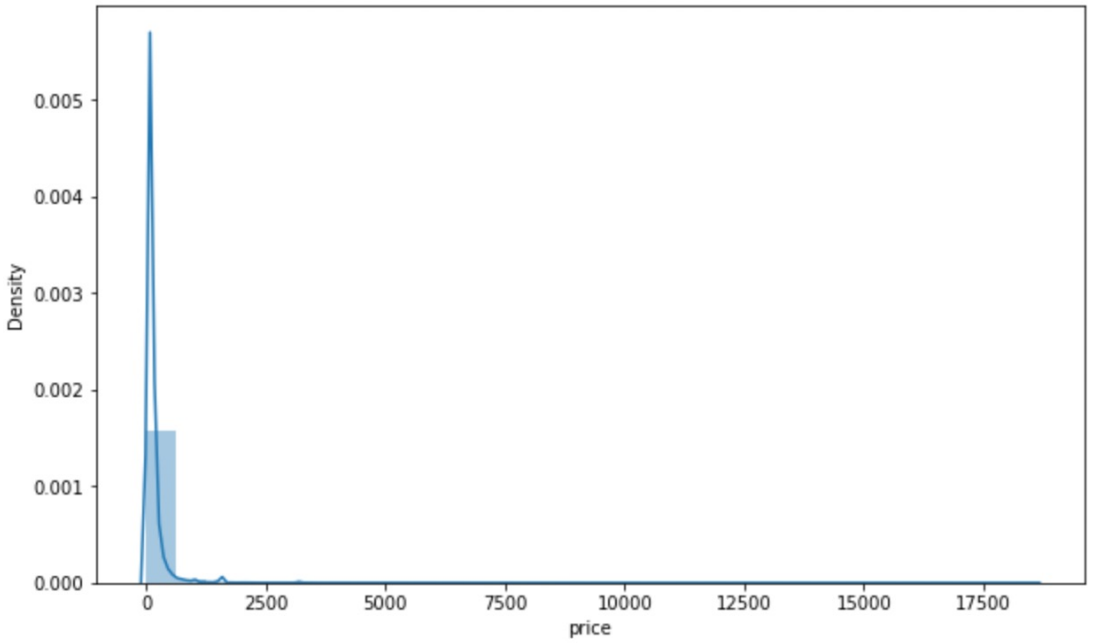
Codes, analysis, and further explanations…

GITHUB REPOSITORY

https://github.com/fabriziorocco/Airbnb-Project

REMOVING OUTLIERS

- Price range of Airbnb London Data contains extreme values.
- Therefore, we have removed the instances with more than or equal to 1195.



Removing outliers in price

As can be seen in the graph above, the graph is very skewed. This is because of outliers and therefore we will remove every val from the mean. This is a general way of removing outliers.

In [38]:
```
#checking the distribution of price in order to remove outliers
mean_price = airbnb['price'].mean()
std_price = airbnb['price'].std()
min_z = mean_price - 3 * std_price
max_z = mean_price + 3 * std_price

print(mean_price)
print(std_price)
print(min_z)
print(max_z)
```

```
145.20722978346663
349.93237346485677
-904.5898906111038
1195.004350178037
```

In [39]:
```
airbnb.drop(airbnb[airbnb['price'] >= 1195].index, inplace = True)
```

PREPROCESSING DATA

Label-Encoded	One-hot Encoded	Filled with Medians
host_response_time	host_is_superhost	host_response_time
room_type	host_identity_verified	host_response_rate
host_since_year	instant_bookable	host_acceptance_rate
final_neighborhood		review_scores_rating
		review_scores_checkin
		review_scores_accuracy
		review_scores_cleanliness
		review_scores_communication
		review_scores_location
		review_scores_value
		reviews_per_month

FILLING MISSING VALUES

We have created two new columns: 'beds' and 'bedrooms'

1. We computed the average number of beds and bedrooms per accommodate per apartment
2. We added these all up, to divide by the length of the dataset to generate the average
3. We use the average number of beds per person, to multiply with the number of accommodates
4. Then, all missing values in the original beds and bedrooms are filled with the value created at step 3.

```
#Calculating average number of beds and bedrooms per accommodate

airbnb['ratio_beds'] = airbnb['beds']/airbnb['accommodates']
airbnb['ratio_bedrooms'] = airbnb['bedrooms']/airbnb['accommodates']

#Taking the average of all for beds
sum_beds = airbnb['ratio_beds'].sum()
length = len(airbnb)
avg_beds_A = sum_beds / length

#Taking the average of all for bedrooms
sum_bedrooms = airbnb['ratio_bedrooms'].sum()
avg_bedrooms_A = sum_bedrooms / length

#Creating a new column based on the average for beds
airbnb['avg_beds'] = airbnb['accommodates'] * avg_beds_A

#Creating a new column based on the average for bedrooms
airbnb['avg_bedrooms'] = airbnb['accommodates'] * avg_bedrooms_A

#In case a value is NA, it takes from these columns
airbnb["beds"].fillna(airbnb['avg_beds'], inplace = True)
airbnb["bedrooms"].fillna(airbnb['avg_bedrooms'], inplace = True)
```

FEATURE ENGINEERING

- We computed the average bedrooms and bathrooms per person to include in as a feature. The reason for this is that both bedrooms, bathrooms and accommodates have a lot of predictive power in the model.

```
airbnb['bedrooms_p_a'] = airbnb['bedrooms'] / airbnb['accommodates']
airbnb['bathrooms_p_a'] = airbnb['bathrooms_text'] / airbnb['accommodates']
```

NEURAL NETWORK WITH KERAS

Parameter	Value
Epochs	200
Batch Size	128
Callbacks	Early stopping monitoring
Optimizer	Adam
Loss	Mean Absolute Error

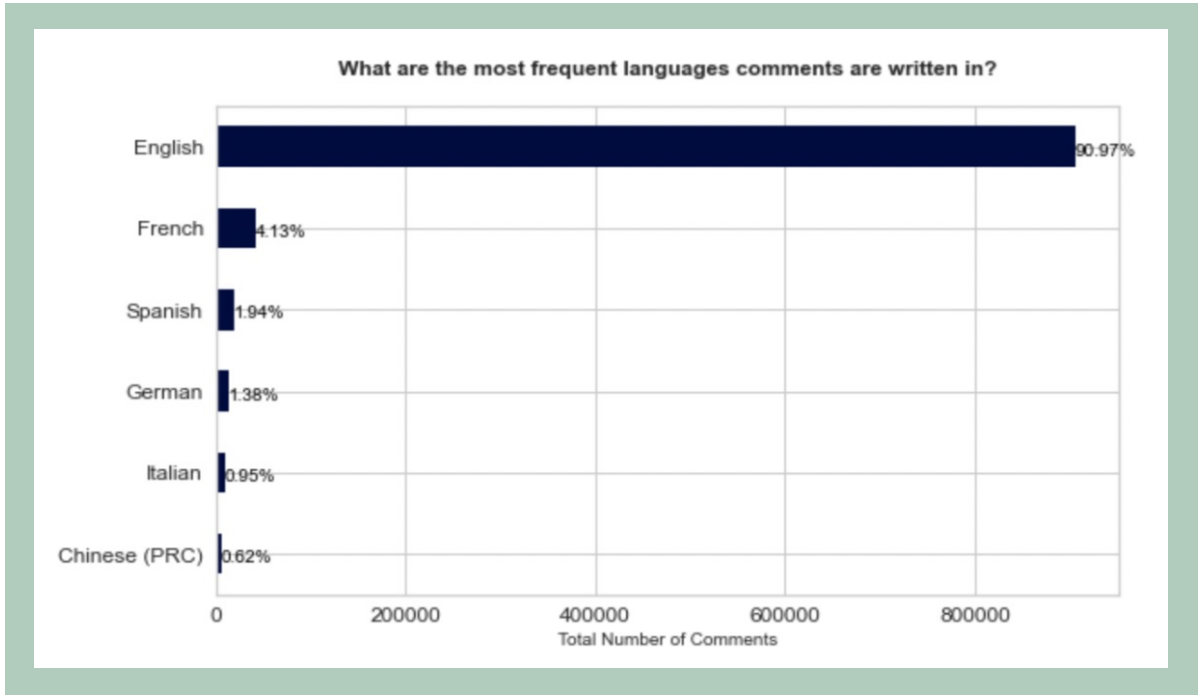
```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_22 (Dense)            (None, 80)                4640

 dropout_14 (Dropout)        (None, 80)                0

 dense_23 (Dense)            (None, 120)               9720

 dropout_15 (Dropout)        (None, 120)               0

 dense_24 (Dense)            (None, 20)                2420

 dropout_16 (Dropout)        (None, 20)                0

 dense_25 (Dense)            (None, 10)                210

 dropout_17 (Dropout)        (None, 10)                0

 dense_26 (Dense)            (None, 21256)             233816

=================================================================
Total params: 250,806
Trainable params: 250,806
Non-trainable params: 0
_________________________________________________________________
```

PREPROCESSING DATA – DETECTING LANGUAGES

- In a touristic Airbnb location such as London, it is common that some reviews are not in English.
- Thus, we have looked at the occurrence and frequency of other languages using the 'langdetect' library.

- The 'langdetect' function has given an error due to:
  - emojis (e.g. "⭐") that were used as comments,
  - expressive punctuation that was given as the only comment (e.g. "!!!!!!!!"),
  - other non-interpretable combinations of full stops, brackets and special characters.

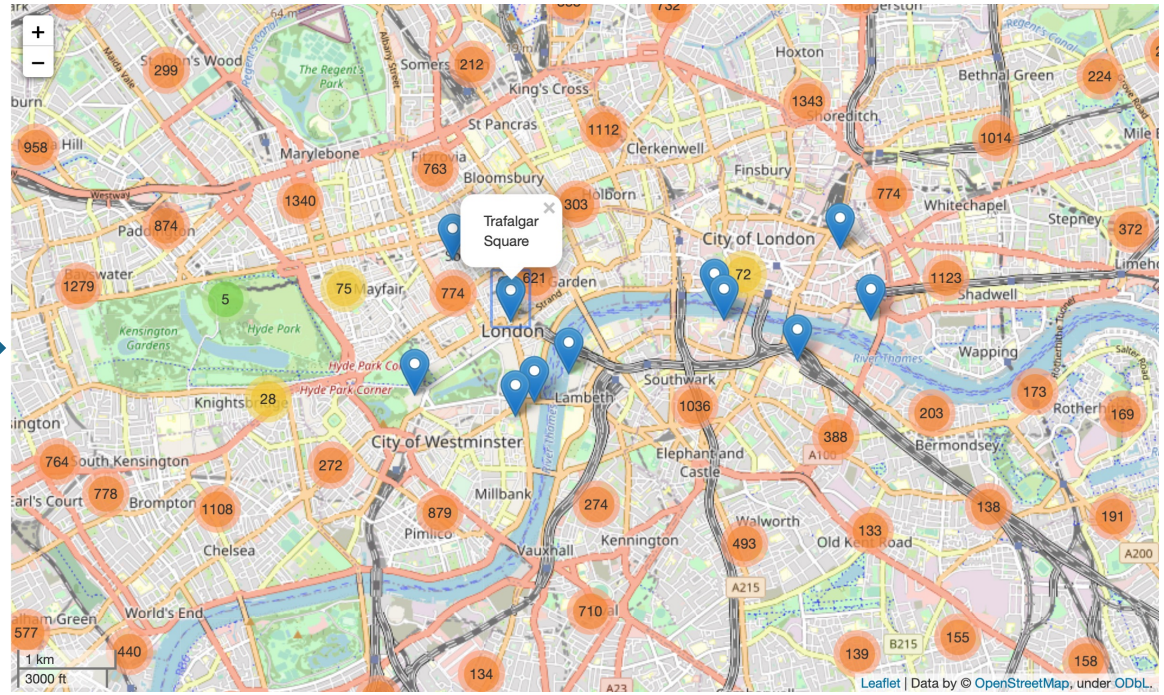
- We have dropped these rows from the 'reviews' dataset to increase interpretability.



esade RAMON LLULL UNIVERSITY

DISTRIBUTION OF LISTINGS ON MAP USING FOLIUM

```
# Neighborhood Distribution according to count) of listings
latitude  = listings['latitude'].tolist()
longitude = listings['longitude'].tolist()

locations = list(zip(latitude,longitude))

neighbourhood_map = folium.Map(
    location =[listings["latitude"].mean(),0],
    zoom_start = 11,
    control_scale=True,
    tiles = 'OpenStreetMap')

attractive_spots_df.apply(lambda row:folium.Marker(
    location=[row["lat"],row["long"]],
    radius=10,
    popup=row['name']).add_to(neighbourhood_map),axis=1)

FastMarkerCluster(data=locations).add_to(neighbourhood_map)
neighbourhood_map
```



esade RAMON LLULL UNIVERSITY

TEXT CLEANING USING REGEX AND NLTK

```
# Cleaning the text data

reviews['comments'].str.replace('\d+','')# remove numbers

reviews['comments'].str.lower()# lowercase

reviews['comments'].str.replace('\r\n',"")# remove windows new line

reviews['comments'].str.replace('\r<br/>',"")# remove html new line

reviews['comments'].apply(
    lambda x: " ".join([i for i in x.split()
    if i not in (union_set)])) # remove all the stopwords with nltk library

reviews['comments'].str.replace('[^\w\s]'," ")# remove all punctuation

reviews['comments'].str.replace('\s+', ' ') #replace x spaces by one space

enriched_reviews['comments'].values[5]
```

```
'i m happy alina s guest we ve great time enjoyed stay alina great host
felt welcomed her alina s house location convenient min walk finsbury p
ark tube station also direct picadilly line heathrow airport in case yu
early departure use opportunity sleep bit train flat nice clean comfort
able especially double bed new mattress slept like newborn also red sof
a small roof terrace great enjoyed last night sky to going visit highly
reccomend alina beautiful house stay in alina thank much hope see one a
gain '
```

CREATING A WORDCLOUD FROM WORD COUNTS

```python
#Create the word cloud
cvec_dict = dict(
    zip(clean_cvec_df.words,
        clean_cvec_df.counts))

wordcloud = WordCloud(width=800,
                      height=400,
                      mode = 'RGBA',
                      max_words = 200,
                      background_color=None)

wordcloud.generate_from_frequencies(frequencies=cvec_dict)
plt.figure( figsize=(20,10) )
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

esade
RAMON LLULL UNIVERSITY

THE WEB APP: MODULE 1

Data downloaded ✅

Rows	Columns	City
66641	57	London, UK

Look for similar apartments

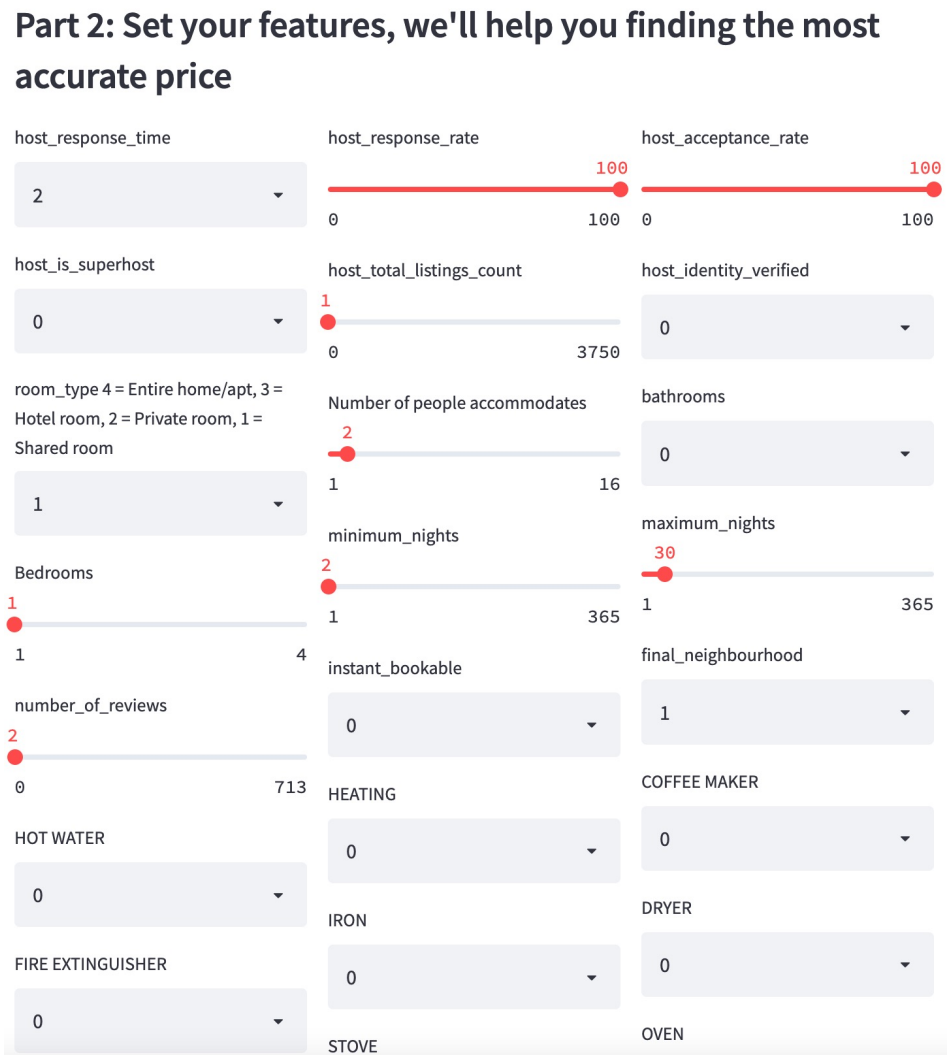
Apartment close to London Eye with 3 bedrooms and balcony

Recommend

Top 5 similar apartments

	listing_url	name	description	price
19605	https://www.airbnb.com/…	next london eye apartment	stunning apartment in th…	235.(
40250	https://www.airbnb.com/…	modern luxury chelsea a…	modern luxury bedrooms…	140.(
42708	https://www.airbnb.com/…	london eye apartment he…	beautiful brand new apar…	138.(
42977	https://www.airbnb.com/…	modern contemporary be…	modern luxury bedrooms…	160.(
61377	https://www.airbnb.com/…	london eye apartment he…	beautiful brand new apar…	106.(

THE WEB APP: MODULE 2



Predict

The price of your house is [50.16297389]€

esade
RAMON LLULL UNIVERSITY

Part 3: Sentiment analysis of reviews in your neighbourhood

Choose your neighbourhood

Westminster

Positive neighbourhood data

Positively Tuned Comments

Negative neighbourhood data

Negatively Tuned Comments