IUM 25L - Projekt

Data analysis

In [1]:
```
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

from nocarz.config import RAW_DATA_DIR, PROCESSED_DATA_DIR, ID_COLUMNS, CATEGORI
```

Users

In [2]:
```
users = pd.read_csv(RAW_DATA_DIR / "users.csv")
users
```

Out[2]:

	id	name	surname	city	street	street_numbe
0	449065179	Benedykta	Białas	Radomsko	Rybaki	25
1	29806310	Hipolit	Majewski	Nałęczów	Lubuska	43
2	176082216	Franciszka	Turowska	Mogielnica	Ajschylosa	50
3	225052416	Zbyszek	Waglewski	Konstancin-Jeziorna	Siarczanogórska	102
4	583625490	Ola	Słowakiewicz	Łódź	Sytkowska	17
...	...	...	...	...	...	
63673	22109770	Halina	Mackaewicz	Kietrz	Bobrownicka	131
63674	400576776	Iliana	Wieczorkowska	Sulechów	Łady	113
63675	187632320	Leokadiusz	Gogolewski	Izbica Kujawska	Rawicz-Mysłowskiego Mieczysława	11
63676	122330593	Celestyn	Glinka	Tuszyn	Młyńska Boczna	87
63677	3419287	Lech	Więcek	Lewin Brzeski	Mrzeżyńska	4

63678 rows × 7 columns

Basic information about the dataset

In [3]:
```
print(f"Number of records: {users.shape[0]}")
print(f"Number of attributes: {users.shape[1]}")
print("\nColumn information:")
users.info()
```

```
Number of records: 63678
Number of attributes: 7

Column information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63678 entries, 0 to 63677
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             63678 non-null  int64
 1   name           63678 non-null  object
 2   surname        63678 non-null  object
 3   city           63678 non-null  object
 4   street         63678 non-null  object
 5   street_number  63678 non-null  int64
 6   postal_code    63678 non-null  object
dtypes: int64(2), object(5)
memory usage: 3.4+ MB
```

Statistics for numerical columns

In [4]:
```
print("\nDescriptive statistics:")
users.describe().T
```

```
Descriptive statistics:
```

Out[4]:

	count	mean	std	min	25%	50%
id	63678.0	1.653834e+08	1.654927e+08	5633.0	31802792.25	102999084.5
street_number	63678.0	6.990593e+02	4.052032e+02	1.0	346.00	699.0

Amount and percentage of rows dropped

In [5]:
```
processed_users = users.dropna(subset=['id']).reset_index(drop=True)
processed_users
```

Out[5]:

	id	name	surname	city	street	street_numbe
0	449065179	Benedykta	Białas	Radomsko	Rybaki	25
1	29806310	Hipolit	Majewski	Nałęczów	Lubuska	43
2	176082216	Franciszka	Turowska	Mogielnica	Ajschylosa	50
3	225052416	Zbyszek	Waglewski	Konstancin-Jeziorna	Siarczanogórska	102
4	583625490	Ola	Słowakiewicz	Łódź	Sytkowska	17
...	...	...	...	...	...	
63673	22109770	Halina	Mackaewicz	Kietrz	Bobrownicka	131
63674	400576776	Iliana	Wieczorkowska	Sulechów	Łady	113
63675	187632320	Leokadiusz	Gogolewski	Izbica Kujawska	Rawicz-Mysłowskiego Mieczysława	11
63676	122330593	Celestyn	Glinka	Tuszyn	Młyńska Boczna	87
63677	3419287	Lech	Więcek	Lewin Brzeski	Mrzeżyńska	4

63678 rows × 7 columns

In [6]:
```
total_rows = len(users)
dropped_rows = total_rows - len(processed_users)
drop_percentage = (dropped_rows / total_rows) * 100 if total_rows > 0 else 0

print(f"Total rows in original dataset: {total_rows}")
print(f"Rows dropped due to missing 'id': {dropped_rows}")
print(f"Percentage of rows dropped: {drop_percentage:.2f}%")
```

```
Total rows in original dataset: 63678
Rows dropped due to missing 'id': 0
Percentage of rows dropped: 0.00%
```

Save the processed data

In [7]:
```
processed_users.to_csv(PROCESSED_DATA_DIR / "users.csv", index=False)
```

Reviews

In [8]:
```
reviews = pd.read_csv(RAW_DATA_DIR / "reviews.csv")
reviews
```

Out[8]:

	listing_id	id	date	reviewer_id	reviewer_name
0	42515	563807	2011-09-24	997025	Dounia
1	42515	1296837	2012-05-17	2348546	D Corinne
2	42515	1358497	2012-05-27	2346980	Natalia
3	42515	2365282	2012-09-21	3503874	Ela
4	42515	3580013	2013-02-19	4185464	Nitin
...	...	...	...	...	...
72307	1313808360361659351	1317398851155668561	2024-12-22	593879932	Pamela Caroline
72308	1313808360361659351	1318040082994961354	2024-12-23	321427694	Raphael
72309	1313808360361659351	1318848921058080173	2024-12-24	321427694	Raphael
72310	1313808360361659351	1319517684620627355	2024-12-25	659980121	Paulo Jorge
72311	1314416675763134591	1317343860305443284	2024-12-22	481981207	Kosovare

listing_id	id	date	reviewer_id	reviewer_name

72312 rows × 6 columns

Basic information about the dataset

In [9]:
```
print(f"Number of records: {reviews.shape[0]}")
print(f"Number of attributes: {reviews.shape[1]}")
print("\nColumn information:")
reviews.info()
```

```
Number of records: 72312
Number of attributes: 6

Column information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 72312 entries, 0 to 72311
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   listing_id     72312 non-null  int64 
 1   id             72312 non-null  int64 
 2   date           72312 non-null  object
 3   reviewer_id    72312 non-null  int64 
 4   reviewer_name  72312 non-null  object
 5   comments       72305 non-null  object
dtypes: int64(3), object(3)
memory usage: 3.3+ MB
```

Statistics for numerical columns

In [10]:
```
print("\nDescriptive statistics:")
reviews.describe().T
```

```
Descriptive statistics:
```

Out[10]:

	count	mean	std	min	25%	50%
listing_id	72312.0	2.074621e+17	3.837994e+17	42515.0	1.173646e+07	2.609328e+07
id	72312.0	5.885498e+17	5.007632e+17	563807.0	4.574197e+08	6.918819e+17
reviewer_id	72312.0	1.624679e+08	1.648932e+08	5633.0	3.036651e+07	9.950773e+07

Amount and percentage of rows dropped

In [11]:
```
processed_reviews = reviews.dropna(subset=['listing_id', 'id']).reset_index(drop
processed_reviews
```

Out[11]:

	listing_id	id	date	reviewer_id	reviewer_name
0	42515	563807	2011-09-24	997025	Dounia
1	42515	1296837	2012-05-17	2348546	D Corinne
2	42515	1358497	2012-05-27	2346980	Natalia
3	42515	2365282	2012-09-21	3503874	Ela
4	42515	3580013	2013-02-19	4185464	Nitin
...	...	...	...	...	...
72307	1313808360361659351	1317398851155668561	2024-12-22	593879932	Pamela Caroline
72308	1313808360361659351	1318040082994961354	2024-12-23	321427694	Raphael
72309	1313808360361659351	1318848921058080173	2024-12-24	321427694	Raphael
72310	1313808360361659351	1319517684620627355	2024-12-25	659980121	Paulo Jorge
72311	1314416675763134591	1317343860305443284	2024-12-22	481981207	Kosovare

listing_id	id	date	reviewer_id	reviewer_name

72312 rows × 6 columns

In [12]:
```
total_rows = len(reviews)
dropped_rows = total_rows - len(processed_reviews)
drop_percentage = (dropped_rows / total_rows) * 100 if total_rows > 0 else 0

print(f"Total rows in original dataset: {total_rows}")
print(f"Rows dropped due to missing 'listing_id' or 'id': {dropped_rows}")
print(f"Percentage of rows dropped: {drop_percentage:.2f}%")
```

```
Total rows in original dataset: 72312
Rows dropped due to missing 'listing_id' or 'id': 0
Percentage of rows dropped: 0.00%
```

Listings

In [13]:
```
listings = pd.read_csv(RAW_DATA_DIR / "listings.csv")
listings
```

Out[13]:

	id	listing_url	scra
0	42515	https://www.nocarz.pl/rooms/42515	202412290
1	203997	https://www.nocarz.pl/rooms/203997	202412290
2	276025	https://www.nocarz.pl/rooms/276025	202412290
3	338682	https://www.nocarz.pl/rooms/338682	202412290
4	399388	https://www.nocarz.pl/rooms/399388	202412290
...	...	...	
2752	1318859808229991842	https://www.nocarz.pl/rooms/1318859808229991842	202412290
2753	1319353272672215826	https://www.nocarz.pl/rooms/1319353272672215826	202412290
2754	1319753554977771528	https://www.nocarz.pl/rooms/1319753554977771528	202412290
2755	1319753624300180036	https://www.nocarz.pl/rooms/1319753624300180036	202412290
2756	1320187730227740458	https://www.nocarz.pl/rooms/1320187730227740458	202412290

2757 rows × 75 columns

Basic information about the dataset

In [14]:
```
print(f"Number of records: {listings.shape[0]}")
print(f"Number of attributes: {listings.shape[1]}")
print("\nColumn information:")
listings.info()
```

```
Number of records: 2757
Number of attributes: 75

Column information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2757 entries, 0 to 2756
Data columns (total 75 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   id                                            2757 non-null   int64
 1   listing_url                                   2757 non-null   object
 2   scrape_id                                     2757 non-null   int64
 3   last_scraped                                  2757 non-null   object
 4   source                                        2757 non-null   object
 5   name                                          2757 non-null   object
 6   description                                   2659 non-null   object
 7   neighborhood_overview                         1052 non-null   object
 8   picture_url                                   2757 non-null   object
 9   host_id                                       2757 non-null   int64
 10  host_url                                      2757 non-null   object
 11  host_name                                     2757 non-null   object
 12  host_since                                    2757 non-null   object
 13  host_location                                 2240 non-null   object
 14  host_about                                    1332 non-null   object
 15  host_response_time                            2015 non-null   object
 16  host_response_rate                            2015 non-null   object
 17  host_acceptance_rate                          2309 non-null   object
 18  host_is_superhost                             2726 non-null   object
 19  host_thumbnail_url                            2757 non-null   object
 20  host_picture_url                              2757 non-null   object
 21  host_neighbourhood                            27 non-null     object
 22  host_listings_count                           2757 non-null   int64
 23  host_total_listings_count                     2757 non-null   int64
 24  host_verifications                            2757 non-null   object
 25  host_has_profile_pic                          2757 non-null   object
 26  host_identity_verified                        2757 non-null   object
 27  neighbourhood                                 1052 non-null   object
 28  neighbourhood_cleansed                        2757 non-null   object
 29  neighbourhood_group_cleansed                  0 non-null      float64
 30  latitude                                      2757 non-null   float64
 31  longitude                                     2757 non-null   float64
 32  property_type                                 2757 non-null   object
 33  room_type                                     2757 non-null   object
 34  accommodates                                  2757 non-null   int64
 35  bathrooms                                     2067 non-null   float64
 36  bathrooms_text                                2756 non-null   object
 37  bedrooms                                      2550 non-null   float64
 38  beds                                          2063 non-null   float64
 39  amenities                                     2757 non-null   object
 40  price                                         2068 non-null   object
 41  minimum_nights                                2757 non-null   int64
 42  maximum_nights                                2757 non-null   int64
 43  minimum_minimum_nights                        2757 non-null   int64
 44  maximum_minimum_nights                        2757 non-null   int64
 45  minimum_maximum_nights                        2757 non-null   int64
 46  maximum_maximum_nights                        2757 non-null   int64
 47  minimum_nights_avg_ntm                        2757 non-null   float64
 48  maximum_nights_avg_ntm                        2757 non-null   float64
 49  calendar_updated                              0 non-null      float64
 50  has_availability                              2683 non-null   object
```

```
 51  availability_30                                2757 non-null   int64
 52  availability_60                                2757 non-null   int64
 53  availability_90                                2757 non-null   int64
 54  availability_365                               2757 non-null   int64
 55  calendar_last_scraped                          2757 non-null   object
 56  number_of_reviews                              2757 non-null   int64
 57  number_of_reviews_ltm                          2757 non-null   int64
 58  number_of_reviews_l30d                         2757 non-null   int64
 59  first_review                                   2081 non-null   object
 60  last_review                                    2081 non-null   object
 61  review_scores_rating                           2081 non-null   float64
 62  review_scores_accuracy                         2081 non-null   float64
 63  review_scores_cleanliness                      2081 non-null   float64
 64  review_scores_checkin                          2081 non-null   float64
 65  review_scores_communication                    2081 non-null   float64
 66  review_scores_location                         2081 non-null   float64
 67  review_scores_value                            2081 non-null   float64
 68  license                                        1 non-null      object
 69  instant_bookable                               2757 non-null   object
 70  calculated_host_listings_count                 2757 non-null   int64
 71  calculated_host_listings_count_entire_homes    2757 non-null   int64
 72  calculated_host_listings_count_private_rooms   2757 non-null   int64
 73  calculated_host_listings_count_shared_rooms    2757 non-null   int64
 74  reviews_per_month                              2081 non-null   float64
dtypes: float64(17), int64(23), object(35)
memory usage: 1.6+ MB
```

Statistics for numerical columns

In [15]:
```
print("\nDescriptive statistics:")
listings.describe().T
```

```
Descriptive statistics:
```

Out[15]:

	count	mean	std	
id	2757.0	5.713685e+17	5.283351e+17	4.2515(
scrape_id	2757.0	2.024123e+13	0.000000e+00	2.02412
host_id	2757.0	1.794862e+08	1.919876e+08	6.7837(
host_listings_count	2757.0	3.521545e+01	9.560836e+01	1.0000(
host_total_listings_count	2757.0	4.677983e+01	1.207214e+02	1.0000(
neighbourhood_group_cleansed	0.0	NaN	NaN	
latitude	2757.0	4.620714e+01	1.972549e-02	4.61424
longitude	2757.0	6.144963e+00	2.480122e-02	5.97391
accommodates	2757.0	2.642728e+00	1.503393e+00	1.0000(
bathrooms	2067.0	1.179245e+00	4.908848e-01	0.0000(
bedrooms	2550.0	1.243137e+00	8.274552e-01	0.0000(
beds	2063.0	1.554048e+00	1.071572e+00	0.0000(
minimum_nights	2757.0	8.474791e+00	4.299344e+01	1.0000(
maximum_nights	2757.0	4.261581e+02	4.067376e+02	1.0000(
minimum_minimum_nights	2757.0	7.759521e+00	4.170338e+01	1.0000(
maximum_minimum_nights	2757.0	8.636924e+00	4.213541e+01	1.0000(
minimum_maximum_nights	2757.0	5.557548e+02	4.530387e+02	1.0000(
maximum_maximum_nights	2757.0	5.774628e+02	4.515392e+02	1.0000(
minimum_nights_avg_ntm	2757.0	8.317991e+00	4.192604e+01	1.0000(
maximum_nights_avg_ntm	2757.0	5.706972e+02	4.490267e+02	1.0000(
calendar_updated	0.0	NaN	NaN	
availability_30	2757.0	1.281320e+01	1.229672e+01	0.0000(
availability_60	2757.0	2.740624e+01	2.482561e+01	0.0000(
availability_90	2757.0	4.308487e+01	3.766778e+01	0.0000(
availability_365	2757.0	1.440025e+02	1.373085e+02	0.0000(
number_of_reviews	2757.0	2.622851e+01	5.769636e+01	0.0000(
number_of_reviews_ltm	2757.0	6.544432e+00	1.424067e+01	0.0000(
number_of_reviews_l30d	2757.0	3.935437e-01	1.091264e+00	0.0000(
review_scores_rating	2081.0	4.735214e+00	3.845453e-01	1.0000(
review_scores_accuracy	2081.0	4.766535e+00	3.839779e-01	1.0000(
review_scores_cleanliness	2081.0	4.717967e+00	4.043822e-01	1.0000(
review_scores_checkin	2081.0	4.816338e+00	3.438892e-01	1.0000(
review_scores_communication	2081.0	4.807871e+00	3.626918e-01	1.0000(

	count	mean	std	
review_scores_location	2081.0	4.792946e+00	3.261770e-01	1.00000
review_scores_value	2081.0	4.618933e+00	4.308944e-01	1.00000
calculated_host_listings_count	2757.0	1.804897e+01	4.518974e+01	1.00000
calculated_host_listings_count_entire_homes	2757.0	1.663076e+01	4.479012e+01	0.00000
calculated_host_listings_count_private_rooms	2757.0	1.340225e+00	3.341848e+00	0.00000
calculated_host_listings_count_shared_rooms	2757.0	4.715270e-03	7.839718e-02	0.00000
reviews_per_month	2081.0	1.046814e+00	1.493456e+00	1.0000

Amount and percentage of rows dropped

In [16]:
```
processed_listings = listings.dropna(subset=ID_COLUMNS).reset_index(drop=True)
processed_listings
```

Out[16]:

	id	listing_url	scra
0	42515	https://www.nocarz.pl/rooms/42515	202412290
1	203997	https://www.nocarz.pl/rooms/203997	202412290
2	276025	https://www.nocarz.pl/rooms/276025	202412290
3	338682	https://www.nocarz.pl/rooms/338682	202412290
4	399388	https://www.nocarz.pl/rooms/399388	202412290
...	...	...	
2752	1318859808229991842	https://www.nocarz.pl/rooms/1318859808229991842	202412290
2753	1319353272672215826	https://www.nocarz.pl/rooms/1319353272672215826	202412290
2754	1319753554977771528	https://www.nocarz.pl/rooms/1319753554977771528	202412290
2755	1319753624300180036	https://www.nocarz.pl/rooms/1319753624300180036	202412290
2756	1320187730227740458	https://www.nocarz.pl/rooms/1320187730227740458	202412290

2757 rows × 75 columns

In [17]:
```
total_rows = len(listings)
dropped_rows = total_rows - len(processed_listings)
drop_percentage = (dropped_rows / total_rows) * 100 if total_rows > 0 else 0

print(f"Total rows in original dataset: {total_rows}")
print(f"Rows dropped due to missing values: {dropped_rows}")
print(f"Percentage of rows dropped: {drop_percentage:.2f}%")
```

```
Total rows in original dataset: 2757
Rows dropped due to missing values: 0
Percentage of rows dropped: 0.00%
```

Property types

In [18]:
```
processed_listings['property_type'].describe()
```

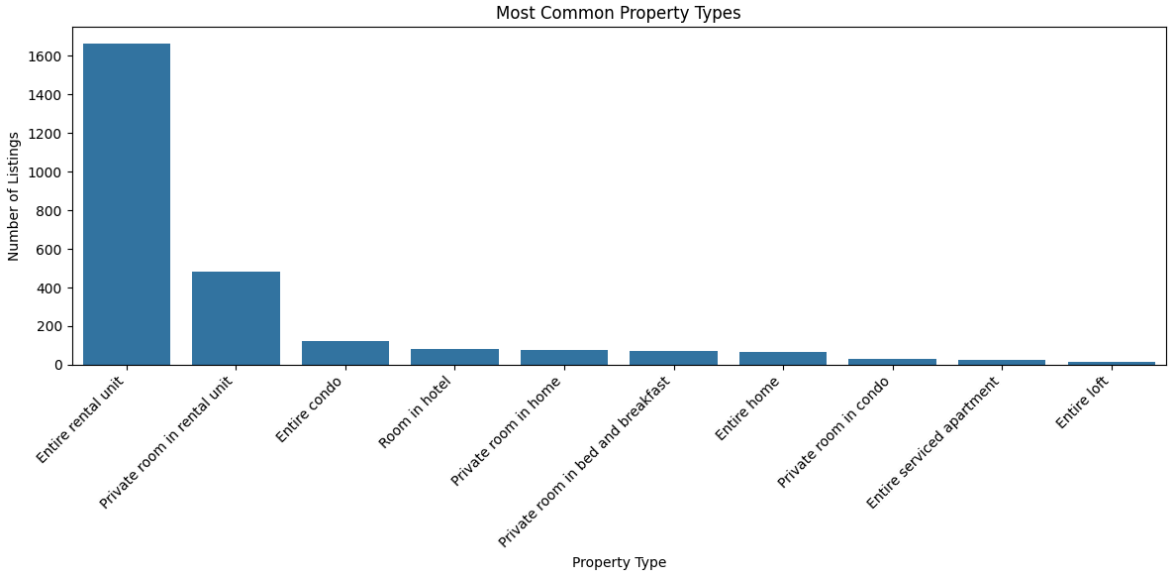
Out[18]:
```
count                   2757
unique                    37
top       Entire rental unit
freq                    1666
Name: property_type, dtype: object
```

In [19]:
```
property_type_counts = processed_listings['property_type'].value_counts()

print("Most common property types:")
print(property_type_counts.head(10))
```

```
Most common property types:
property_type
Entire rental unit                   1666
Private room in rental unit           484
Entire condo                          123
Room in hotel                          81
Private room in home                   74
Private room in bed and breakfast      69
Entire home                            66
Private room in condo                  31
Entire serviced apartment              27
Entire loft                            16
Name: count, dtype: int64
```

In [20]:
```
plt.figure(figsize=(12, 6))
sns.barplot(x=property_type_counts.head(10).index, y=property_type_counts.head(1
plt.title("Most Common Property Types")
plt.xlabel("Property Type")
plt.ylabel("Number of Listings")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Prediction columns

In [21]:
```
processed_listings['price'] = processed_listings['price'].replace('[\$,]', '', r

print("Numerical Columns Statistics")
numerical_stats = processed_listings[NUMERICAL_TARGETS].describe().T
numerical_stats['missing'] = processed_listings[NUMERICAL_TARGETS].isna().sum()
numerical_stats['missing_percent'] = (processed_listings[NUMERICAL_TARGETS].isna
display(numerical_stats)

print("\nCategorical Columns")
categorical_stats = []

for col in CATEGORICAL_TARGETS:
    unique_count = processed_listings[col].nunique()
    missing_count = processed_listings[col].isna().sum()
    missing_percent = (missing_count / len(processed_listings) * 100).round(2)

    value_counts = processed_listings[col].value_counts().head(5)
    top_values = ", ".join([f"{val} ({count})" for val, count in value_counts.it
    if len(top_values) > 100:
        top_values = top_values[:100] + "..."

    categorical_stats.append({
        'Column': col,
        'Unique Values': unique_count,
        'Missing Values': f"{missing_count} ({missing_percent}%)",
        'Top 5 Values (count)': top_values
    })

display(pd.DataFrame(categorical_stats))
```

```
Numerical Columns Statistics
```

	count	mean	std	min	25%	50%	75%	max	missing
accommodates	2757.0	2.642728	1.503393	1.0	2.0	2.0	4.0	15.0	0
bathrooms	2067.0	1.179245	0.490885	0.0	1.0	1.0	1.0	6.5	690
bedrooms	2550.0	1.243137	0.827455	0.0	1.0	1.0	1.0	9.0	207
beds	2063.0	1.554048	1.071572	0.0	1.0	1.0	2.0	12.0	694
price	2068.0	153.799323	271.245144	18.0	85.0	115.0	162.0	9726.0	689

```
Categorical Columns
```

	Column	Unique Values	Missing Values	Top 5 Values (count)
0	property_type	37	0 (0.0%)	Entire rental unit (1666), Private room in ren...
1	room_type	4	0 (0.0%)	Entire home/apt (1960), Private room (791), Sh...
2	bathrooms_text	22	1 (0.04%)	1 bath (1626), 1 shared bath (310), 1.5 baths ...
3	neighbourhood	78	1705 (61.84%)	Genève, Switzerland (657), Geneva, Switzerland...
4	name	2688	0 (0.0%)	Résidence Le Montbrillant (15), Double Busines...
5	description	2413	98 (3.55%)	Make life easier at this peaceful, centrally l...

Save the processed data

In [22]:
```
processed_listings = processed_listings[ID_COLUMNS + CATEGORICAL_TARGETS + NUMER
processed_listings.to_csv(PROCESSED_DATA_DIR / "listings.csv", index=False)
```

Split the dataset into train and test sets

In [23]:
```
train_data, test_data = train_test_split(processed_listings, test_size=0.2, rand
train_data.to_csv(PROCESSED_DATA_DIR / "train.csv", index=False)
test_data.to_csv(PROCESSED_DATA_DIR / "test.csv", index=False)
```