MADRID REAL ESTATE MARKET ANALYSIS

HALYNA ABELCHAKOVA

TABLE OF CONTENTS

MADRID
REAL ESTATE

OBJECTIVE & CONTEXT

Objective

Analyze and predict property prices in the Madrid real estate market using data-driven approaches.

Context

With five years of real estate experience in Ukraine and a data analytics course, I analyzed the Madrid real estate market.

Inspired by a friend's relocation to Madrid, this project provides data-driven insights to support her decision-making.

DATA AND DATA SOURCES

MADRID REAL ESTATE

MADRID REAL ESTATE

DATA AND DATA SOURCES

- **58** columns
- **21,742** rows

Data Sources

The primary dataset, "Madrid Real Estate Market," was sourced from Kaggle and includes detailed information from online property ads. Additional data on population and area for each Madrid district was gathered from Wikipedia through web scraping.

KAGGLE DATASET PAGE

Search

MIRBEK TOKTOGARAEV AND 1 COLLABORATOR · UPDATED 4 YEARS AGO

40 | New Notebook | Download (1 MB)

Madrid real estate market

Real estate listings in Madrid crawled from popular internet portals

Data Card | Code (9) | Discussion (3) | Suggestions (0)

About Dataset

Context

The dataset consist listings from popular real estate portals of Madrid.

LOCATION

Madrid is one of the most visited cities in Europe both by tourists and businesspeople, and it's where many important local and multinational companies have their headquarters. Therefore Madrid enjoys both a large influx of tourists as well as people seeking to live and work in the city to give their professional careers a boost.

ATTRACTIVE PRICES

Usability
10.00

License
Data files © Original Authors

Expected update frequency
Never

Tags

Real Estate | Housing

WIKIPEDIA PAGE

USED FOR WEBSCRAPING

District	Population (1 Jan 2020)[127]	Area (ha)
Centro	140,991	522.82
Arganzuela	156,176	646.22
Retiro	120,873	546.62
Salamanca	148,405	539.24
Chamartín	148,039	917.55
Tetuán	161,991	537.47
Chamberí	141,397	467.92
Fuencarral-El Pardo	250,636	23,783.84
Moncloa-Aravaca	122,164	4,653.11

EDA

MADRID REAL ESTATE

INITIAL EXPLORATORY DATA ANALYSIS

The goal of EDA is to understand the dataset's structure, uncover distributions, identify patterns, and detect anomalies. This step guides further analysis and modeling. We used Matplotlib and Seaborn libraries for EDA.

```
1  housing_madrid.describe()
```

✓ 0.0s Python

	Unnamed: 0	id	sq_mt_built	sq_mt_useful	n_rooms	n_bathrooms	n_floors	sq_mt_allotment	latitude	longitude	portal	door	
count	21742.000000	21742.000000	21616.000000	8228.000000	21742.000000	21726.000000	1437.000000	1432.000000	0.0	0.0	0.0	0.0	2
mean	10870.500000	10871.500000	146.920892	103.458192	3.005749	2.091687	3.128740	241.692737	NaN	NaN	NaN	NaN	-5
std	6276.519112	6276.519112	134.181865	88.259192	1.510497	1.406992	0.907713	247.484853	NaN	NaN	NaN	NaN	
min	0.000000	1.000000	13.000000	1.000000	0.000000	1.000000	1.000000	1.000000	NaN	NaN	NaN	NaN	-3
25%	5435.250000	5436.250000	70.000000	59.000000	2.000000	1.000000	2.000000	2.000000	NaN	NaN	NaN	NaN	7
50%	10870.500000	10871.500000	100.000000	79.000000	3.000000	2.000000	3.000000	232.000000	NaN	NaN	NaN	NaN	1
75%	16305.750000	16306.750000	162.000000	113.000000	4.000000	2.000000	4.000000	354.000000	NaN	NaN	NaN	NaN	1
max	21741.000000	21742.000000	999.000000	998.000000	24.000000	16.000000	7.000000	997.000000	NaN	NaN	NaN	NaN	2

DATA CLEANING AND PREPROCESSING

MADRID REAL ESTATE

DATA CLEANING AND PREPROCESSING

To ensure data quality and suitability for analysis, a meticulous cleaning and preprocessing phase was undertaken:

- **Handling Missing Values:**

1. Dropping empty columns;
2. Dropping rows with missing values ('sq_mt_built', 'n_bathrooms');
3. Replacing missing values with 'Unknown' (categorical column 'house_type_id');
4. Replacing missing values with ‘0’ (numerical columns 'n_floors', 'parking_price' and 'sq_mt_allotment');
5. Replacing missing values with average (numerical column 'built_year');
6. Replacing missing values with mode (numerical column 'floor');
7. Replacing the outlier value with the new value (‘built_year’);

- **Standardization:**

1. Bringing all values of columns to a standard ('floor', ‘house_type_id’);
2. Extracting important data from columns with excessive information ('subtitle', ‘neighborhood_id’);
3. Renameing columns in the dataset;

- **Formatting:**

1. Convert Data Types to relevant;

After cleaning:

- **32** columns
- **21,600** rows

SQL

MADRID REAL ESTATE

SQL: DATABASE CREATION, ERD & SQL QUERIES

MADRID REAL ESTATE

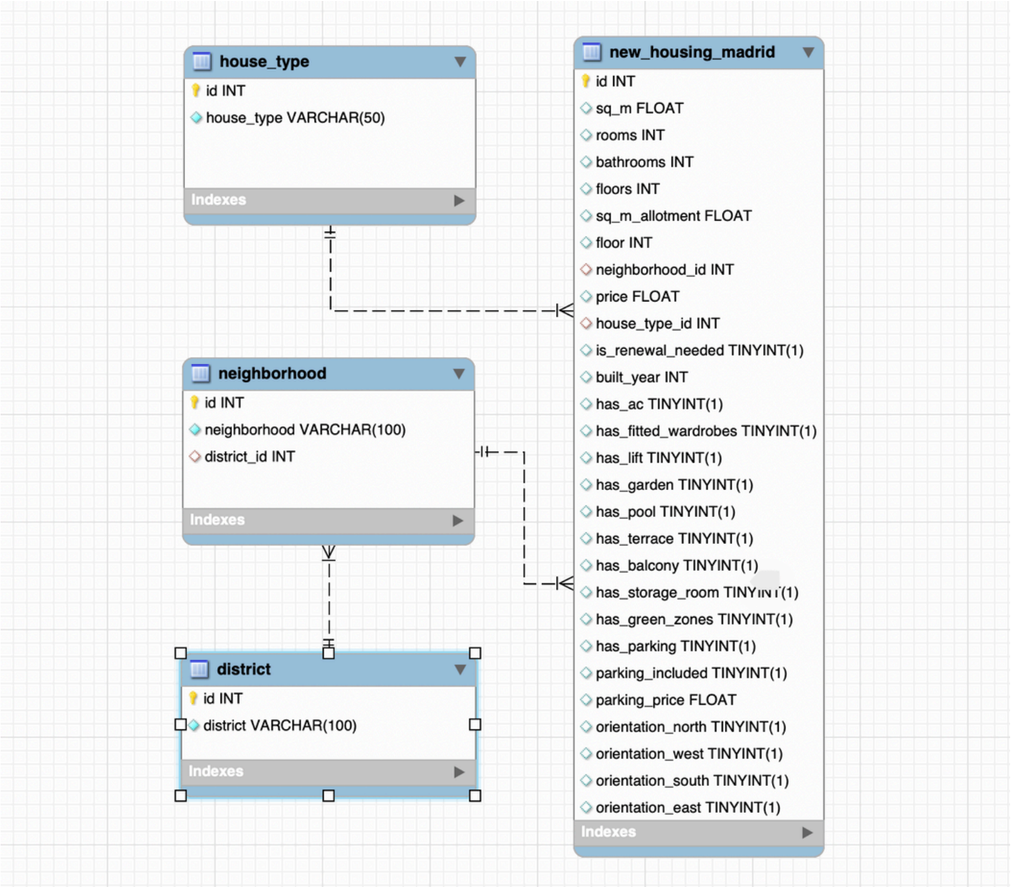
The process began with the creation of a relational database **schema** and **tables**.

I designed **Entity-Relationship Diagram** (ERD) to illustrate the relationships between the tables within the database.

With the schema and relationships defined, the next step involved **populating** the database tables with data.

Once the database was populated, SQL **queries** were employed to extract relevant information for analysis.

ERD



SCHEMA AND TABLES CREATION

```sql
-- Create Schema
CREATE SCHEMA IF NOT EXISTS real_estate;


-- Create neighborhood table
CREATE TABLE neighborhood (
    id INT PRIMARY KEY AUTO_INCREMENT,
    neighborhood VARCHAR(100) UNIQUE NOT NULL,
    district_id INT,
    FOREIGN KEY (district_id) REFERENCES district(id)
);

-- Create house_type table
CREATE TABLE house_type (
    id INT PRIMARY KEY AUTO_INCREMENT,
    house_type VARCHAR(50) UNIQUE NOT NULL
);
```


TABLES POPULATION

```sql
-- Populate district table
INSERT IGNORE INTO district (district)
SELECT DISTINCT district FROM housing_madrid;

-- Populate neighborhood table
INSERT IGNORE INTO neighborhood (neighborhood, district_id)
SELECT DISTINCT hm.neighborhood, d.id
FROM housing_madrid hm
JOIN district d ON hm.district = d.district;
```

SQL QUERIES

MADRID REAL ESTATE

These queries were designed to gather insights into the real estate market in Madrid

```sql
-- Calculate the Average Price per Square Meter by District
SELECT d.district, ROUND(AVG(nhm.price / nhm.sq_m), 2) AS avg_price_per_sqm
FROM new_housing_madrid nhm
JOIN neighborhood n ON nhm.neighborhood_id = n.id
JOIN district d ON n.district_id = d.id
GROUP BY d.district
ORDER BY avg_price_per_sqm DESC;
```

district	avg_price_per_sqm
Salamanca	6550.42
Chamberí	5777.22
Chamartín	5529.03
Centro	5488.09

```sql
-- Get Average Property Price and Count by House Type
SELECT ht.house_type,
       COUNT(*) AS num_properties,
       ROUND(AVG(nhm.price), 2) AS avg_price
FROM new_housing_madrid nhm
JOIN house_type ht ON nhm.house_type_id = ht.id
GROUP BY ht.house_type
ORDER BY avg_price DESC;
```

house_type	num_properties	avg_price
House or Chalet	1811	1675450.96
Penthouse	1031	836842.62
Duplex	675	811825.34
Apartment	18083	514107.45

```sql
-- Count the Number of Properties with Specific Features
SELECT
    COUNT(*) AS total_properties,
    SUM(has_garden) AS properties_with_garden,
    SUM(has_pool) AS properties_with_pool,
    SUM(has_terrace) AS properties_with_terrace
FROM new_housing_madrid;
```

total_properties	properties_with_garden	properties_with_pool	properties_with_terrace
21600	1445	5063	9455

```sql
-- Retrieve All Properties in a Specific District
SELECT nhm.id, nhm.sq_m, nhm.rooms, nhm.bathrooms, nhm.price, n.neighborhood, d.district
FROM new_housing_madrid nhm
JOIN neighborhood n ON nhm.neighborhood_id = n.id
JOIN district d ON n.district_id = d.id
WHERE d.district = 'Villaverde';
```

id	sq_m	rooms	bathrooms	price	neighborhood	district
1	64	2	1	85000	San Cristóbal	Villaverde
15	64	3	1	72000	San Cristóbal	Villaverde
89	65	3	1	94000	San Cristóbal	Villaverde
159	66	3	1	104900	San Cristóbal	Villaverde
161	74	3	1	100000	San Cristóbal	Villaverde
167	69	3	1	107000	San Cristóbal	Villaverde

API

MADRID REAL ESTATE

API DEVELOPMENT

A Flask API was developed to interact with the real estate database, enabling flexible and efficient **data retrieval** based on user criteria. The process included:

- Setting up the **Flask** application
- Creating **endpoints** for data retrieval
- **Deploying** the API

The API allows users to filter property listings based on criteria such as number of rooms, district, and house type.

FILTERING SEARCH BY NUMBER OF ROOMS:


```
{
  "district": "Villa de Vallecas",
  "h_id": 1837,
  "house_type": "Apartment",
  "neighborhood": "Ensanche de Vallecas - La Gavia",
  "rooms": 1,
  "sq_m": 61.0
},
{
  "district": "Usera",
  "h_id": 1862,
  "house_type": "Apartment",
  "neighborhood": "Pradolongo",
  "rooms": 1,
  "sq_m": 31.0
},
```

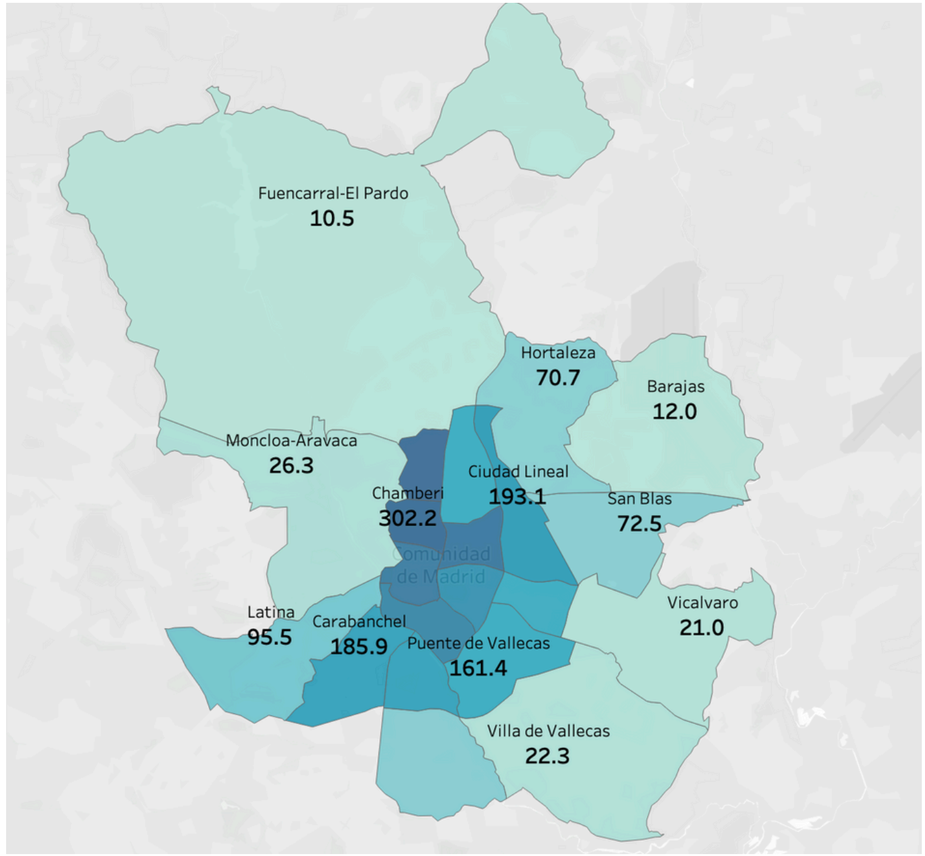
VISUALIZATION

MADRID REAL ESTATE

VISUALIZATIONS

NUMBER OF APARTMENTS FOR SALE BY DISTRICT

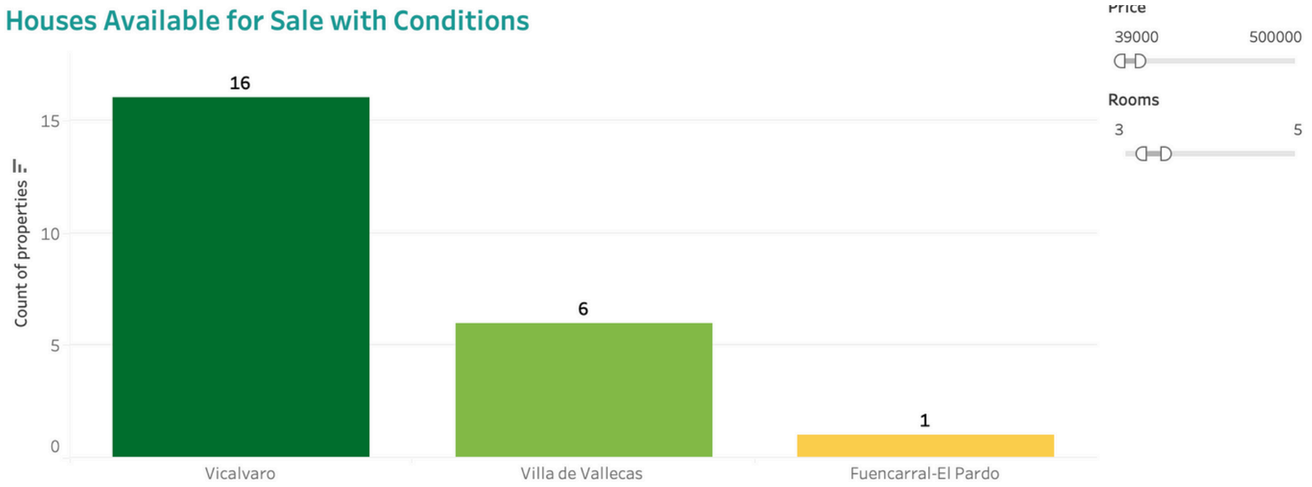
DENSITY OF POPULATION OF MADRID BY DISTRICT



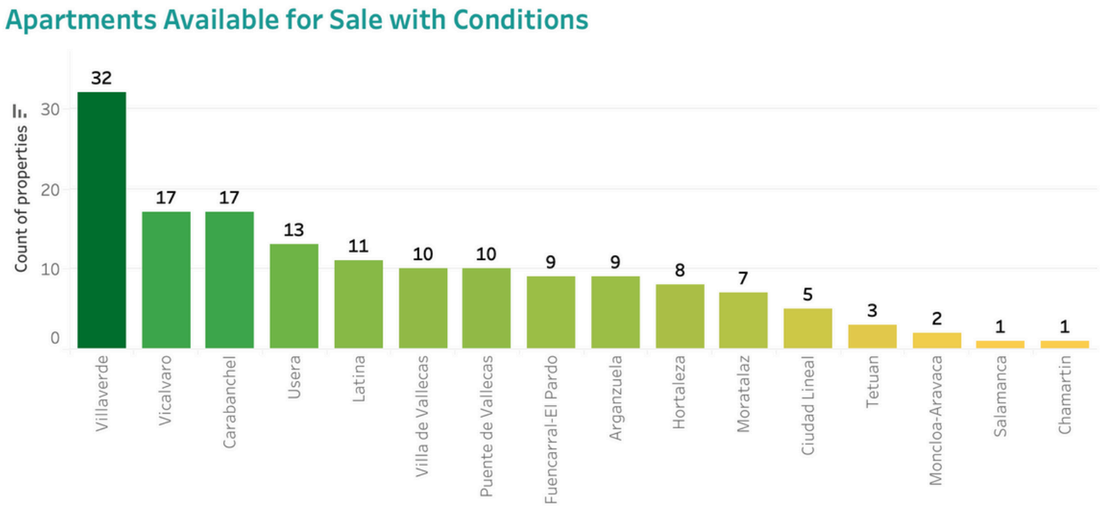
VISUALIZATIONS

MADRID REAL ESTATE

COUNT OF HOUSES WITH POOL AND GARDEN WITHIN 500,000 EUR BUDGET



COUNT OF APARTMENTS WITH AC, FITTED WARDROBE, STORAGE ROOM, PARKING, TERRACE, WITH 3-5 ROOMS AND WITHIN 500,000 EUR BUDGET



MACHINE LEARNING

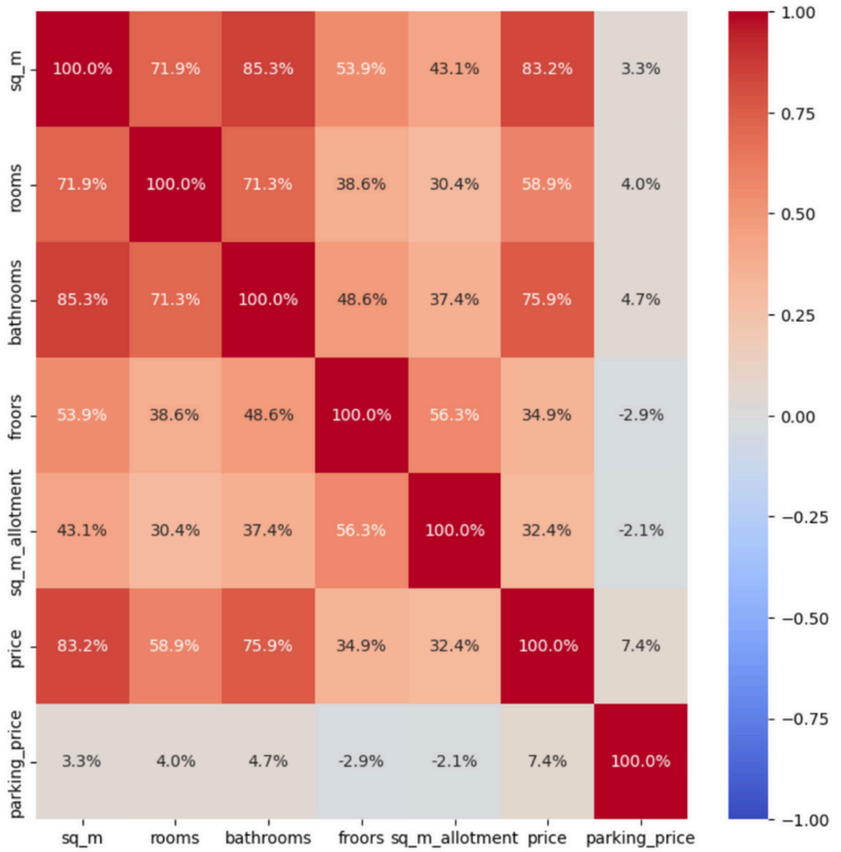
MADRID REAL ESTATE

MACHINE LEARNING

The **machine learning** component focused on **predicting property prices** using various features. This involved:

- feature engineering (encoding categorical variables, feature scaling);
- model training
- evaluation
- hyperparameter tuning.

CORRELATION OF NUMERICAL VALUES



MACHINE LEARNING

STEP BY STEP BREAKDOWN

1. **Feature engineering**: encoding categorical variables, feature scaling;
2. **Selecting features** (categorical encoded, numerical and boolean values) and **target** (price);
3. **Splitting** data on training and testing parts;
4. Selecting **models** and **parameters** for each of them:

- '**KNN**': {'n_neighbors': [**5**, 10]},
- '**Bagging**': {'n_estimators': [50, **100**], 'estimator__max_depth': [10, **20**]},
- '**Random Forest**': {'n_estimators': [50, **100**], 'max_depth': [10, **20**]},
- '**Gradient Boosting**': {'n_estimators': [50, **100**], 'max_depth': [**10**, 20]},
- '**AdaBoost**': {'n_estimators': [50, **100**], 'estimator__max_depth': [10, **20**]},
- '**XGBoost**': {'n_estimators': [**50**, 100], 'max_depth': [**10**, 20]}

The **AdaBoost model** was the best predictive model, and gave an accurate result of R2 Score: **0.886683,** with **100** estimators and max depth of **20.**

	MAE	RMSE	R2 Score
KNN	212645.142778	446221.786758	0.649880
Bagging	113202.226693	255086.862254	0.885583
Random Forest	113285.087235	254884.169352	0.885764
Gradient Boosting	113267.822344	262184.606438	0.879127
AdaBoost	109234.366710	253857.046533	0.886683
XGBoost	117279.152071	271279.143374	0.870596

CHALLENGES

MADRID REAL ESTATE

CHALLENGES

```python
import requests

def get_coordinates(address):
    api_key = '076ffe039a7f4d7ba5596db0d316ce87'  # Replace with your OpenCage API key
    url = f'https://api.opencagedata.com/geocode/v1/json?q={address}&key={api_key}'
    response = requests.get(url).json()
    if response['results']:
        return response['results'][0]['geometry']['lat'], response['results'][0]['geometry']['lng']
    else:
        return None, None

# Apply geocoding to the full addresses
housing_madrid['latitude'], housing_madrid['longitude'] = zip(*housing_madrid['full_address'].apply(get_coordinates))

# Display the updated DataFrame
housing_madrid[['full_address', 'latitude', 'longitude']].head()
```

[31] ✓ 65m 2.3s

	full_address	latitude	longitude
0	64, Calle de Godella, Calle de Godella, 64, Sa...	19.617250	-99.066010
1	nan, Calle de la del Manojo de Rosas, Calle de...	52.777770	9.075390
2	68, Calle del Talco, Calle del Talco, 68, San ...	16.332760	-96.595620
3	nan, Calle Pedro Jiménez, Calle Pedro Jiménez,...	40.342633	-3.707647
4	nan, Carretera de Villaverde a Vallecas, Carre...	38.500000	-0.500000

```python
# Fill missing latitude and longitude values
housing_madrid['latitude'].fillna(housing_madrid['latitude'], inplace=True)
housing_madrid['longitude'].fillna(housing_madrid['longitude'], inplace=True)
```

[32] ✓ 0.0s

My biggest challenge

Creating an interactive map in Tableau was the biggest challenge. Here's the process and obstacles:

1. Initial Plan with Geocoding:
   - Used property addresses to convert into coordinates.
   - Faced a limit of 2,500 requests and slow processing times.
2. Attempt with Zip Codes:
   - Created a new column with zip codes for each district as an alternative.
   - This approach did not yield accurate results.
3. Final Solution with Spatial File:
   - Found and used a spatial file of Madrid in Tableau.
   - Successfully created the interactive map using the spatial file.

CONCLUSION

MADRID REAL ESTATE

CONCLUSION

In this project, we analyzed the Madrid real estate market using data analytics to discover key insights for buyers and stakeholders.

We have found the best predictive model (**AdaBoost model)** which gave us accurate level of prediction (**0.886683**);

I would advise to start search from Vicalvaro district as the most relevant to the search criterias;

This project was both interesting and challenging and I hope it will help my friend to find a perfect house.

MADRID REAL ESTATE

THANK YOU

HALYNA ABELCHAKOVA