MASTER

DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK

PROJECT

FLAT OUT FACTS: PREDICTING APARTMENT PRICES USING PROPERTY ATTRIBUTES AND TEXT DESCRIPTIONS

ANASTASIOS KONDO

SUPERVISION:
PROF. CARLOS J. COSTA

JUNE 2025

JURY:
PRESIDENT: PROF. MÁRIO ROMÃO
RAPPORTEUR: PROF. FLÁVIO ROMÃO
SUPERVISOR: PROF. CARLOS J. COSTA

GLOSSARY

ANN – Artificial Neural Network

BERT – Bidirectional Encoder Representations from Transformers

CRISP-DM – Cross-Industry Standard Process for Data Mining

DNN – Deep Neural Network

GB – Gradient Boosting

KNN – K-Nearest Neighbors

MAPE – Mean Absolute Percentage Error

ML – Machine Learning

MLP – Multilayer Perceptron

MLR – Multiple Linear Regression

NLP – Natural Language Processing

OLS – Ordinary Least Squares

PCA – Principal Component Analysis

RF – Random Forest

RMSE – Root Mean Squared Error

SVR – Support Vector Regression

TF-IDF – Term Frequency–Inverse Document Frequency

ABSTRACT

This study investigates how accurately machine learning can predict apartment prices in Limassol, Cyprus, and whether adding textual data from listing descriptions improves performance beyond standard property features. Over 4,000 listings were scraped between November 2024 and February 2025, each containing structured numerical attributes (e.g., area, property age, coordinates) and free-text descriptions written by sellers. The text was preprocessed and vectorized using TF-IDF. Two input sets were tested: one with only structured features, and another combining those with textual data. Five regression algorithms were evaluated using grid search and cross-validation. All machine learning models outperformed the hedonic linear regression benchmark, highlighting their ability to capture more complex pricing patterns. Gradient Boosting performed best, achieving $R^2 = 0.84$ and MAPE = 16.5% without text. Adding descriptions led to a modest improvement ($R^2 = 0.86$, MAPE = 15.6%), suggesting that text captures some qualitative signals not fully reflected in the numeric data. However, the gain was limited, likely due to overlapping content or the shallow representation of TF-IDF. Overall, while listing descriptions offer incremental value, most predictive power stems from the core property features. Future work could explore more advanced embedding techniques to better capture meaning and nuance.

KEYWORDS: House price prediction; Real estate; Machine learning; Text mining

LIST OF TABLES

LIST OF FIGURES

Table of Contents

ACKNOWLEDGEMENTS

This project started with a simple question: what would be worth studying, even when it gets challenging? The goal was to work on something practical, something with purpose that could be applied in real life.

With the housing market being a relevant topic for many, especially future homeowners, the idea was to analyze my city’s market and create a tool to evaluate individual apartment listings using data.

The reality is that real estate websites show the asking price, not necessarily the fair value of the property being advertised. Emotion plays a huge role in our purchases. Why would it be any different when it comes to something as big and symbolic as buying a property? If anything, the emotional weight is even greater. It is hard to imagine making such a major decision without having some numbers to back it. That was the rationale behind this study.

Thank you to Professor Carlos J. Costa for his guidance and support. Special thanks as well to Mr. Loizou from AskWire. His input on feature engineering was especially helpful and much appreciated.

1. Introduction

Predicting house prices is valuable for individuals, businesses, and institutions engaged in the real estate market. Buyers use predictions to gauge whether a listing reflects fair value, helping them avoid overpaying or missing opportunities. Sellers rely on estimates to price their properties competitively. For banks, insurers, and investors, accurate price models inform lending, risk assessment, and investment strategies. These models also support decision-making by highlighting market trends and outliers.

Real estate is a cornerstone of global economic activity. In the European Union, it plays a central role in regional development and household wealth, with residential property often being the largest asset owned by individuals. In Q2 2022, EU housing prices peaked with a 10.5% year-on-year rise before slowing in 2023. By mid-2024, the market had begun recovering, with a 3.0% annual increase. Despite this volatility, some regions bucked the trend, most notably Cyprus, where property sales surged by 31.0% year-on-year. These figures prove the highly uneven and dynamic nature of housing markets (Eurostat, 2024).

The pace of change is particularly pronounced in rapidly growing urban centers, where demand is shaped by foreign direct investment and shifting demographics. Limassol, a coastal city on the southern edge of Cyprus with a population of around 240,000, exemplifies this dynamic. As the island's commercial, and shipping hub, Limassol draws strong interest from foreign investors attracted by its low taxes and warm climate. Cyprus's residency-by-investment scheme, which grants permanent residency to non-EU nationals investing €300,000 or more in real estate, further amplifies demand. Tourism adds another layer of pressure, sustaining year-round interest in short-term rentals and seafront properties. The result is skyrocketing demand for apartments, which dominate the urban market. In Q1 2023, Limassol accounted for approximately one-third of all property sales and 45% of transaction value in Cyprus, with apartment prices rising 10.4% year-on-year, well ahead of other major cities like Nicosia and Paphos (Deloitte, 2023).

This surge in demand reveals the deeper complexity of real estate markets, where prices reflect a mix of context-specific, interrelated factors. Traditional models like hedonic pricing, with their linear assumptions, often struggle to capture these nuances. As real estate markets grow more competitive, the need to identify opportunities and detect inefficiencies has led to the adoption of more advanced tools. Simultaneously, the data itself has evolved, now encompassing not just structured features but also unstructured inputs like images and textual data. In turn, machine learning has gained popularity for its ability to model nonlinear relationships and integrate diverse data sources.

Yet despite this momentum in both the evolution of valuation methods and the growth of markets like Limassol, smaller markets remain largely overlooked in academic research on property price prediction. Many studies have focused on highly liquid, mature markets, leaving smaller, fast-growing cities under-analyzed and full of inefficiencies that create opportunities for those able to detect them. With sales in Limassol jumping by 30 percent year-on-year, it is clear that some actors are already capitalizing on this growth. However, in the broader European context, Cyprus remains a small and underrepresented market as it accounts for less than 1% of the EU's House Price Index, compared to 21% for Germany. This contrast highlights the mismatch between the country's growth momentum and its visibility in real estate research. The profit potential in such markets fuels demand for advanced tools that can uncover inefficiencies and unlock hidden value. In cities like Limassol, where investment is driven by lifestyle appeal, residency incentives, and rental yields, this dynamic is especially pronounced. At the same time, features like sea views, beach proximity, or vague terms like "luxury finishes" introduce subjectivity that complicates pricing even among similar properties. While institutional actors may rely on sophisticated models to exploit these inefficiencies, individual buyers and sellers typically lack such resources, leaving much of this potential untapped. As Zhang et al. (2024) argue, future research should explore whether predictive models generalize across different markets.

Given this gap in the literature, this study asks: *"To what extent can apartment prices in Limassol be accurately predicted using machine learning techniques, and does the inclusion of listing descriptions improve predictive performance compared to structured features alone?"*

To guide the analysis, the study sets out *three main objectives*:

1. To evaluate how accurately apartment prices can be predicted using structured property features
2. To assess whether listing descriptions add meaningful predictive value
3. To compare the performance of different machine learning algorithms

The analysis follows the CRISP-DM framework (Shearer, 2000), commonly used in data mining projects, and aligns with the POST DS methodology (Costa and Aparicio, 2020), which extends CRISP DM by emphasizing structured project management, an approach that suits the iterative model development and evaluation process in this study.

From a theoretical perspective, this study examines whether machine learning models can accurately predict apartment prices in smaller, less mature markets, such as Limassol, and whether incorporating textual data enhances predictions based solely on structured property features. From an empirical perspective, it compares the predictive performance of models trained on structured features alone versus those that also include TF-IDF-transformed listing descriptions. It also examines how prediction accuracy varies across different price segments, offering a detailed view of model performance across the market.

This paper is organized as follows. Section 2 presents a literature review on the primary topics relevant to our work. Section 3 presents the methodology. Section 4 outlines the results of our experiments. Section 5 presents a discussion of the results and how it relates to the literature. Lastly, Section 6 concludes the paper and suggests directions for future research.

2. Literature review

This section covers the three core areas relevant to our study: hedonic price models based on statistical methods, machine learning approaches, and models that apply text mining and natural language processing techniques to house price prediction.

2.1 Hedonic approaches

Real estate price prediction has long attracted researchers' interest due to the financial incentives tied to identifying mispriced properties. These opportunities persist because real estate markets are often inefficient: information is fragmented, disclosure is inconsistent, and valuations rely heavily on subjective judgment (Glaeser et al., 2008). As Herath and Maier (2015) note, high transaction costs, infrequent sales, and the uniqueness of each asset further hinder price discovery. In such settings, even approximate estimates of fair value can offer a strategic advantage.

The aforementioned inefficiencies highlight the value of predictive models. They help not by perfectly detecting mispricing, but by flagging listings that merit closer review (Vargas-Calderón & Camargo, 2020). They are also useful when comparable sales are scarce or pricing is unclear (Khani Dehnavi et al., 2025). Moreover, since house prices tend to be negotiable, data-based estimates can strengthen a buyer's position and help sellers set more realistic asking prices. In this way, these models help reduce information assymetries which are prevalent in real estate markets (Jung et al., 2022; Akerlof, 1970).

The foundation of real estate price modeling was established by Court (1939), who proposed that a product's value could be decomposed into the value of its individual characteristics. This concept was later formalized by Rosen (1974) through the hedonic pricing model, which treats properties as bundles of attributes. He argued that buyers evaluate each structural attribute, such as size, layout, or number of rooms, to determine what they are willing to pay. Empirical research has shown that basic structural features like property size, room count, building age, floor level, and amenities consistently account for a substantial share of price variation (Sirmans et al., 2005).

While Rosen's model established the role of structural property features in price formation, it did not account for spatial relationships between properties. Later research made it clear that location also plays a critical role, not just in terms of neighborhood but in how a property is positioned relative to others. Studies have shown that proximity to factors such as schools, parks, or the coast influences buyer preferences and price patterns (Goodman and Thibodeau, 1998; Pace and Gilley, 1997; Monson, 2009). Frew and Wilson (2002) went further by showing that adding location variables significantly improves the predictive accuracy of hedonic models, even those originally focused only on structural features. This highlights that regardless of model complexity, spatial context is essential. To capture this, modern models often include geographic coordinates and/or engineered distance-based features to account for spatial variation and improve predictive performance (Wei et al., 2022; Rey-Blanco et al., 2024).

Even when location is properly accounted for, traditional hedonic models still face other important limitations. First, they require the analyst to predefine variables, transformations, and interactions. While this can capture simple nonlinearities, it often misses more complex relationships, especially when interactions are not explicitly defined (Chin and Chau, 2003). This limits accuracy and scalability, particularly in high-dimensional datasets (Limsombunchai, 2004). In contrast, machine learning models learn such patterns automatically, making them more suitable for mass appraisal tasks (Peterson & Flanagan, 2009). This was further validated by Ho, Tang, & Wong (2020), who showed that Random Forest and Extra Trees cut test set MAPE by as much as 63.6% relative to the linear regression model.

2.2 Machine learning approaches

A variety of machine learning algorithms have been used for house price prediction. Random Forest (RF), Gradient Boosting (GB) and Support Vector Regression (SVR) are common choices, while many studies have also tested artificial neural networks (ANN). A smaller group of studies takes a time series approach, focusing on forecasting broader market trends. Most existing solutions rely exclusively on numerical housing features, while a smaller subset incorporates unstructured inputs extracted through text mining or images from property descriptions. While there is no clear winner among algorithms, the consensus is that machine learning methods tend to outperform traditional linear or hedonic models.

SVR has proven to be a reliable method for house price prediction, consistently delivering strong results across various datasets and setups. Ho, Tang, and Wong (2020) emphasized that SVR remains a strong option when quick predictions are needed, as it maintains solid accuracy even with limited computation time. Vasquez and Chellamuthu (2021) further showed that kernel choice matters, with nonlinear kernels clearly outperforming linear ones.

When predictive accuracy is prioritized over speed, ensemble methods offer clear advantages. Ho et al. (2020) recommend using RF or GB in such cases, a view echoed across the literature. RF has consistently delivered strong results across different market contexts. In Saint Petersburg, it outperformed linear regression in a mass appraisal of two-room apartments (Antipov & Pokryshevskaya, 2012). In Spain, Baldominos et al. (2018) identified it as the top-performing model for high-end listings. Tchuente and Nyawa (2022) applied it across multiple French cities and reported generally robust performance, despite some variation by location. In South Korea, Hong, Choi, and Kim (2020) found that it cut prediction error down to 5.5%. Even in the typical Boston Housing dataset, Adetunji et al. (2022) found it produced stable and accurate results. Building on the success of RF, GB methods have also gained traction for their strong predictive performance. Zaki et al. (2022), for instance, applied XGBoost to a basic structured dataset and found it delivered strong results. Zhang, Li & Branco (2024) tested GB against RF, SVR, and deep neural networks (DNNs) using two input setups: structured data only and structured plus text, and found that it outperformed all other algorithms in both cases.

Artificial neural network (ANN) solutions, including deep learning, have also been widely explored. Early evidence from Limsombunchai et al. (2004), who used New Zealand housing data, and Peterson and Flanagan (2009), working in a U.S. context, demonstrated that ANNs produced more accurate and stable results than traditional hedonic regression, especially in mass appraisal settings. More recent studies show ANNs remain competitive or superior in diverse applications. For instance, Rampini and Re Cecconi (2022), using real estate data from Italy, found that ANN models outperformed not only traditional methods but also XGBoost, suggesting their suitability for more complex valuation tasks. Moreover, Mostofi et al. (2022) implemented a deep learning architecture enhanced with PCA and showed that DNNs maintained high prediction accuracy even with relatively small samples. Similarly, Kalliola et al. (2021) confirmed the effectiveness of multi-layer perceptrons (MLP) in limited data environments, reinforcing the potential of ANNs when traditional models might struggle. In contrast, Root, Strader, & Huang (2023) argue that ANNs often underperform compared to regression trees and SVMs, as they require large data and careful tuning to deliver consistent results.

An alternative research direction explores hybrid models that integrate multiple algorithms or data types to enhance predictive accuracy. These approaches aim to combine the strengths of different methods such as the interpretability of linear models, the robustness of ensemble trees, and the flexibility of ANNs into a single framework. For example, Varma et al. (2018) fed outputs from linear regression and RF into a neural network to improve performance on Mumbai housing data. Zhao et al. (2019) extracted visual features from property images using a CNN and used XGBoost for the final prediction, outperforming both DNNs and KNN. Zhao and Wang (2023) designed a stacking ensemble combining SVR, RF, GBM, and ridge regression into a meta model, which consistently outperformed individual models across datasets. Similarly, Akyüz et al. (2023) proposed a three stage hybrid system combining residual based clustering, KNN, and SVR, which significantly outperformed standalone models on both real world and benchmark datasets.

While much of the literature treats house price prediction as a regression problem at the individual property level, some studies adopt alternative formulations. For example, time series approaches aim to forecast broader market trends, as in Samadani and Costa (2021), who combined time series models with machine learning to predict price evolution in Portugal. Others frame the task as a classification problem. For instance, Park and Bae (2015) used ML algorithms to predict whether the final sale price would be higher or lower than the listing price, helping sellers make more informed pricing decisions.

To consolidate this review, Table 1 summarizes key studies applying machine learning to real estate price prediction, detailing the markets analyzed, algorithms tested, best-performing models, key predictive variables, and reported performance metrics.

2.3 Text mining approaches

Text mining refers to the process of extracting meaningful information from unstructured textual data by transforming it into a structured, analyzable format (Feldman & Sanger, 2007). In real estate, it is used to extract information from property descriptions, capturing qualitative aspects such as renovation quality, scenic views, or interior finishes that structured variables may overlook. These elements can influence buyer perception and affect price expectations (Baldominos, 2018). Language that reflects exclusivity, comfort, or outdoor appeal has been shown to carry predictive value (Alfano & Guarino, 2022). However, to incorporate textual data into most predictive models, listing descriptions must be transformed into numerical representations using vectorization techniques. These techniques are generally grouped into two categories: frequency-based and context-based methods.

Frequency-based methods transform text into numerical features by capturing how often terms appear, either within a document or across the entire corpus. A basic example is the Bag-of-Words model, which represents text as binary or count-based indicators of word presence. Nowak and Smith (2015) used this method to encode unigrams and bigrams from listing descriptions as dummy variables in a hedonic regression. Their baseline model included only bedroom and bathroom counts and time dummies, omitting key variables like square footage or location, so adding even simple text features led to a 25% reduction in pricing error. While effective, this approach treats all words equally, ignoring their informativeness. This limitation motivates more refined weighting schemes like TF-IDF (Term Frequency–Inverse Document Frequency; Salton & Buckley, 1988), which emphasizes words that are common in a specific listing but rare across others. In real estate, TF-IDF helps surface unique or high-impact phrases that correlate with price. Stevens (2014) applied TF-IDF-transformed features in machine learning models such as SVM and GB, showing improved predictive performance over Bag-of-Words. Similarly, Bushuyev et al. (2024) combined TF-IDF with structured features in a LightGBM model, reporting a 13.4% drop in mean squared error. These studies show that even simple frequency-based representations can significantly complement structured features in predictive performance.

Context-based methods move beyond word counts by learning meaning from how words are used in sentences. Unlike frequency-based approaches that treat each word separately, these models generate embeddings: numerical representations shaped by context, so that words used in similar ways receive similar values. There are two main types: neural embeddings, which focus on nearby words, and transformer-based models like BERT, which consider the entire sentence.

Early neural embedding models such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and Doc2Vec (Le & Mikolov, 2014) create compact numerical representations of text. Word2Vec and Doc2Vec are simple ANNs that learn from nearby words in a sentence, while GloVe relies on word co-occurrence counts across the entire dataset. Word2Vec and GloVe produce fixed vectors for each word while Doc2Vec extends this to full documents. For example, Vargas-Calderón and Camargo (2019) used Doc2Vec to turn property descriptions into vectors and combined them with structured data in an XGBoost model to predict whether a home was priced above or below average. While their goal was classification, the study showed that these text-based features added useful information.

More recently, transformer-based models like BERT (Devlin et al., 2018) generate word embeddings that adapt to their surrounding context. These models capture more subtle meanings and have been applied to real estate price prediction by combining description embeddings with structured features, often leading to improved results. Baur et al. (2023) used BERT to encode listing descriptions and combined the output with a GB model. Using listings from Berlin, Hamburg, Munich, and Los Angeles, they found that adding BERT-based text features reduced pricing errors by up to 17% across all markets, with especially strong gains for higher-priced listings.

One of the most comprehensive evaluations of text representation techniques for house price prediction was conducted by Zhang et al. (2023). Unlike other studies that typically rely on a single technique, they tested TF-IDF, self-trained Word2Vec, several pretrained embeddings (GloVe, FastText, Google News), and BERT across both traditional and deep learning models. Their experimental design included three setups: structured-only, text-only, and structured-plus-text. In the combined setup (the one relevant here), TF-IDF ranked first in predictive performance, followed by Word2Vec and then BERT. Although Word2Vec performed best in the text-only setup, that scenario falls outside this study’s scope. Notably, the performance gap between TF-IDF and the neural embeddings was narrow, suggesting that the added complexity of pretrained models may not justify the trade-off. The authors attribute this to domain mismatch, as models like BERT and pretrained Word2Vec were not trained on real estate–specific language. In contrast, TF-IDF delivered strong results out of the box while also reducing the feature space from over 15,000 to just 412 terms. Given its low complexity, interpretability, and computational efficiency, we argue that TF-IDF offers the best trade-off and will therefore be used in this study.

Building on this prior work, we apply text mining to house price prediction in a smaller, less liquid, and largely unexplored real estate market. While existing studies clearly show that incorporating textual data improves valuation, they almost exclusively focus on large, data-rich markets. These are typically major metropolitan areas with high transaction volumes and abundant online listings, making them ideal for developing and testing models. However, this raises the question of whether such text mining techniques can generalize to smaller, less liquid markets like Limassol, where data availability and market dynamics differ. Zhang et al. (2023) also noted this, emphasizing the need to test the generalizability of text-based models across regions and data sources. We compare models trained on structured features alone to those that combine structured and textual data to isolate their added value. Finally, we evaluate standard machine learning, tree-based ensembles, and dee to assess their effectiveness in this underexplored setting. Given the unique dynamics of Cyprus's housing market, where sales trends have diverged sharply from broader European patterns, it remains unclear whether findings from larger, more liquid markets will hold.

To consolidate this review, Table 1 summarizes key studies applying machine learning to real estate price prediction, including input features, algorithms used, market studied, best-performing models, and reported performance.

Table 1. Summary of studies predicting property price with machine learning

Author(s)	Target Feature	Key Features	Algorithms Used	Market Studied	Best Model(s)	Performance
Ho, Tang & Wong (2020)	Log price	Area, age, floor, travel time	SVM, RF, GBM	Hong Kong	GB	MAPE = 0.32%; R^2 = 0.90
Baldominos et al. (2018)	Price	Total area, internal area, year, rooms, baths, amenities, location	Linear Regr., kNN, MLP, SVR, Tree Ensembles	Madrid, Spain	Ensemble (Regression Trees)	MAPE = 16.8%; R^2 = 0.46
Hong et al. (2020)	Price	Area, rooms, construction year, floor, building type, distance to subway	OLS, RF	Gangnam, South Korea	RF	MAPE = 5.5%; OLS MAPE = 20%
Zaki et al. (2022)	Price	13 structural features	Hedonic regression, XGBoost	Not specified	GB	R^2 = 0.841; Hedonic R^2 = 0.42
Zhang, Li & Branco (2024)	Price (normalized)	80+ features incl. area, bedrooms, pool, parking, location	Linear regression, SVR, RF, GB, DNN	Canada	GB	RMSE = 0.0189; R^2 = 0.841
Limsombunchai et al. (2004)	Price (NZD)	Age, type, bedrooms, baths, garages, amenities	Hedonic regression, ANN	New Zealand	ANN	ANN RMSE = 449,111; R^2 = 0.90Hedonic RMSE = 876,215; R^2 = 0.62
Rampini & Re Cecconi (2021)	Price	Size, location, floor, rooms, condition, energy class	ElasticNet, XGBoost, ANN	Italy	ANN	MAE = €15,360
Mostofi et al. (2022)	Price	Area, room count, age, floor, longitude, latitude	DNN (with PCA), RF, SVR, KNN	Iran	DNN + PCA	MAPE = 7.18%; R^2 = 0.94
Kalliola et al. (2021)	Apartment price (debt-free)	Living area, number of flats, construction year, income, services	MLP (Bayesian-tuned)	Helsinki, Finland	Optimized MLP	RMSE = €33,232; MAE = €23,321; R^2 = 0.95; RME = 8.3%
Zhao & Wang (2023)	Median value of owned homes	13 Boston housing features (e.g. RM, LSTAT, TAX)	LR, SVR, Ridge, Lasso, MLP, RFR, GBR, Stacking Ensemble	Boston, USA	Stacking (SVR, RFR, GBR, LR → Ridge)	MSE = 8.75; R^2 = 0.8841; EVS = 0.8839

3. Methodology

This study follows the CRISP-DM framework (Shearer, 2000), with added structure from the POST-DS methodology (Costa and Aparicio, 2020), which integrates project management into the process. The steps below cover data exploration, preparation, modeling, and evaluation. The full code used in this study is available in a dedicated GitHub repository.

3.1 Data understanding

The dataset combines two types of real estate data: structured features (i.e number of bedrooms, area) and unstructured textual descriptions written by sellers or agents. While structured data captures objective, quantifiable attributes, textual descriptions often include subjective elements commonly found in real estate ads, such as mentions of design, renovation quality, or views. These qualitative aspects can influence perceived value but are not always reflected in fixed fields.

Data was collected via web scraping from Bazaraki.com, the leading property listings platform in Cyprus, using Python and BeautifulSoup. The focus is exclusively on apartments in the Limassol district, a coastal area in the south of Cyprus. Figures 1 and 2 show the geographical context of Cyprus and the Limassol district.

Figure 1. Location of Cyprus within Europe

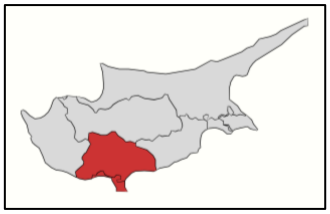
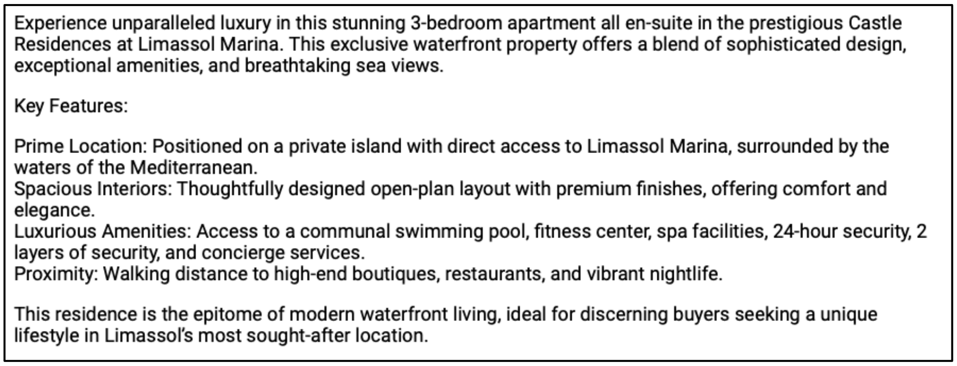


Figure 2. Location of the Limassol District in Cyprus

The scraping process yielded 8,176 apartment listings, each containing both structured features and a corresponding property description, although with some variation. Figure 3 presents an example of a typical listing description. Our initial target was the listed price (later log-transformed for modeling purposes). To ensure uniformity during extraction, any missing fields were left blank. Data was collected between October 2024 and February 2025.



Experience unparalleled luxury in this stunning 3-bedroom apartment all en-suite in the prestigious Castle Residences at Limassol Marina. This exclusive waterfront property offers a blend of sophisticated design, exceptional amenities, and breathtaking sea views.

Key Features:

Prime Location: Positioned on a private island with direct access to Limassol Marina, surrounded by the waters of the Mediterranean.
Spacious Interiors: Thoughtfully designed open-plan layout with premium finishes, offering comfort and elegance.
Luxurious Amenities: Access to a communal swimming pool, fitness center, spa facilities, 24-hour security, 2 layers of security, and concierge services.
Proximity: Walking distance to high-end boutiques, restaurants, and vibrant nightlife.

This residence is the epitome of modern waterfront living, ideal for discerning buyers seeking a unique lifestyle in Limassol's most sought-after location.

Figure 3. Example of a house listing's description text extracted from real estate listings.

3.2 Data preparation

The raw dataset consists of structured property features and unprocessed textual descriptions. The cleaning and preprocessing steps for each set are outlined separately in the following subsections.

3.2.1 Structured features

The extracted data included numerous structured fields, but only those with reliable and consistent coverage were retained. These align well with key variables identified in the literature. In addition, given our study's focus on evaluating the contribution of textual descriptions, the analysis was based on a basic set of structured variables. The selected features and their descriptions are presented in Table 2.

Table 2. Selected structured property features and their descriptions

Feature	Description	Type
area	Total internal area of the apartment in square meters	Continuous
bedrooms	Number of bedrooms in the apartment	Discrete
bathrooms	Number of bathrooms, including en-suite or guest WCs	Discrete
floor	The floor level on which the apartment is located	Discrete
age	Age of the property, calculated from the construction year	Continuous
lat	Latitude coordinate of the apartment's location	Continuous
long	Longitude coordinate of the apartment's location	Continuous
is_penthouse	Boolean indicator of whether the apartment is a penthouse	Binary
has_pool	Boolean indicator of whether the apartment has access to a pool (luxury proxy)	Binary

The following four steps were applied to clean and preprocess the features, resulting in the final dataset used for modeling. They are presented in the order in which they were carried out: Feature removal, value mapping, missing values, and outlier removal.

Features such as detailed amenities (e.g., fireplace, garden, alarm system), parking types, air conditioning levels, energy ratings, and furnishing status were removed due to inconsistent presence across listings and excessive categorical granularity. Preliminary tests using the models in this study showed only negligible gains in predictive performance, not enough to justify the added complexity.

Next, value mapping was applied to standardize bedroom and bathroom counts. Non-numeric entries were converted to integers: studios were mapped to 0 bedrooms, and "6 and more" was set to 6. For bathrooms, "5+" was treated as 5. While this simplification is not ideal, it reflects how the information was presented in the listings. These cases made up a small portion of the data and were unlikely to materially affect the results.

The features with the most extensive missingness were the floor number and the construction year, which were used to calculate the property age. Each required dedicated handling.

Floor levels ranged from "ground floor" to "8th and above", without finer granularity. To model this as a continuous variable, the "ground floor" was mapped to 0, and all upper floors were set to 8. Missing values were first filled by extracting numbers from listing descriptions using regex, which still left 41% of entries missing. These were then imputed using a KNN regressor trained on area, price, bedrooms, and location. The model achieved a mean absolute error of approximately ±1 floor, an acceptable trade-off to retain observations without introducing substantial noise. While grouping upper floors may reduce precision, the chosen models can still capture such effects.

Each listing included a condition label (new, resale, or under construction) and a construction year, which was either a valid year, the string "older" (pre-1994), or missing. "Older" entries were flagged and cleared. After applying regex to extract years from descriptions, 82% of the dataset still had missing values. These were composed of 24% from new listings, 9% from resale, 47% from under construction, and 1% from older listings. Assigning 2025 to under-construction properties (resulting in age = 0) reduced the overall missing rate to 35%. The remaining values were imputed using the median construction year within each group. The final variable used was property age, calculated as 2025 minus the construction year.

Other missing values were observed in the geographical coordinates and the number of bathrooms. These observations were dropped. For binary features, missingness was assumed to indicate non-presence, which is a reasonable assumption in the context of real estate listings where sellers typically specify what a property includes, not what it lacks. These were set to 0.

Finally, to ensure data quality, outliers in price and area were identified using three complementary statistical methods: the percentile method (e.g., 1st and 99th percentiles), the interquartile range (IQR) method (values beyond 1.5 times the IQR), and the standard deviation method (values exceeding three standard deviations). A data point was removed only if all three methods agreed, ensuring true outliers were eliminated while minimizing unnecessary data loss. Additionally, manual filtering was applied to remove listings with incorrect geographical coordinates, including those placed outside Cyprus or in unrealistic locations. A visual inspection was also conducted by plotting the listings on a map, allowing for the removal of isolated listings that were significantly distant from property clusters.

3.2.2 Textual features

The property descriptions are texts found in each listing, written to inform and persuade potential buyers. They often include context like “quiet residential area near the city center” or highlight features such as “brand new kitchen”. An example of this is shown in Figure 3. We aim to extract valuable information from this unstructured text to enhance price prediction. To do so, we apply the TF-IDF embedding technique.

We used Scikit-learn’s TfidfVectorizer, removing words that appeared in fewer than 3% or more than 95% of listings. This helped eliminate both noise from rare terms and dilution from overly frequent ones. As a result, the number of features dropped from 3,812 to 279, improving both computational efficiency and model performance. The resulting TF-IDF matrix was then used as input in the regression models. A full list of terms is available in the GitHub repository.

Since TF-IDF works best with clean and consistent text, we applied several preprocessing steps to prepare the listing descriptions. The rationale behind each step was to help the model detect patterns that structured features might miss, without repeating information already present in those features. We also wanted to avoid anything that could accidentally leak the target variable into the model. The steps were the following:

- We began by removing Greek text, emojis, URLs, and listing-specific identifiers, such as registration numbers or internal codes. Since fewer than 3% of listings were written in Greek, this step allowed us to focus on English descriptions only without losing valuable content. The administrative codes added no real value for price prediction, so they were taken out to reduce noise.
- Next, we cleaned up symbols and numbers. We replaced characters like "+", "&", and "%" with their word equivalents: "plus", "and", "percent", to keep everything consistent. All numeric content was removed, including digits, written numbers like "one" to "ten", ordinals like "1st" and "second", and anything representing currency, such as "€" or "eur". These elements typically correspond to structured features already included in the model like area, or bedroom count so we removed them from the text to avoid duplication and ensure our models learn from descriptive content rather than repeating known inputs.
- We manually standardized domain-specific terms to make sure different forms of the same idea were treated as one. For instance, we converted “sqm”, “m²”, and “sq.m” into a single version, “square meters”. We also grouped different spellings of “air

conditioning", such as "ac", "a/c", and "air-condition". This was done by reviewing the dataset manually, identifying inconsistent wording, and mapping all variations to a unified format.

- We replaced all punctuation and special characters with spaces to avoid breaking words into meaningless fragments. Excess whitespace was removed to ensure the text was clean and ready for tokenization.
- We used spaCy to tokenize the text, splitting each cleaned description into individual words. This allowed us to work with the text at the word level for the next steps in the pipeline.
- After tokenization, we removed all tokens that were not valid English words using WordNet. This helped eliminate typos, misspellings, foreign terms, and broken pieces of text.
- With only meaningful tokens remaining, we applied lemmatization using spaCy. This step reduced each word to its base form, for example, changing "running" to "run" and "houses" to "house", to make sure our models recognize different forms of the same concept as one. We also removed any single-character tokens since they do not carry useful information.
- Finally, we removed common English stop words like "the", "is", and "and". These words show up frequently in almost every description but do no contribute anything meaningful to the prediction task, so removing them helped reduce noise and improve model focus.

With the core cleaning steps complete, we shifted focus to refining the vocabulary itself. While the remaining text was standardized and free of obvious noise, many high-frequency terms still carried little predictive value. To address this, we followed a simple iterative process aimed at filtering out generic language and highlighting more meaningful content:

1. **Extract N-grams**
   With clean lemmas, we extracted unigrams, bigrams, and trigrams: single words, two-word phrases, and three-word phrases that frequently appeared across the corpus. N-grams help identify recurring language patterns and offer a quick overview of the dominant terms in the dataset. This step helped us assess whether the remaining text contained signals that could actually help the model predict price.
2. **Visualize**
   We visualized the most frequent n-grams to evaluate what kinds of terms were dominating the corpus. This allowed us to ask a critical question: Can this word or phrase help differentiate one property from another in terms of price?
3. **Remove Real Estate-Specific Stop Words**
   Using that lens, we removed common domain-specific words like "area," "bedrooms," and "apartment." These terms either duplicated information already captured in structured

features (for example, number of bedrooms, property type), or added no value because they appeared in almost every listing. If a word does not help the model distinguish between listings, it does not belong in the text.

4. **Repeat**
   We repeated the entire cycle, extracting n-grams, visualizing them, and removing any additional high-frequency, low-value terms that offered no real differentiation between properties. This iterative loop continued until the n-gram plots reflected a more focused and informative set of terms aligned with our goal of helping the model predict price.

Figure 4 presents the most frequent unigrams before and after removing real estate-specific stopwords. Initially, words like “area,” “apartment,” and “bedroom” dominate. These are clear examples of terms that add little value, as they either duplicate information already captured in structured features or appear in nearly every listing. After removing the real estate stopwords we identified, the resulting text reveals more descriptive and potentially differentiating terms such as “view,” “terrace,” “private,” and “quiet,” which are better suited to help the model distinguish between listings and predict price.

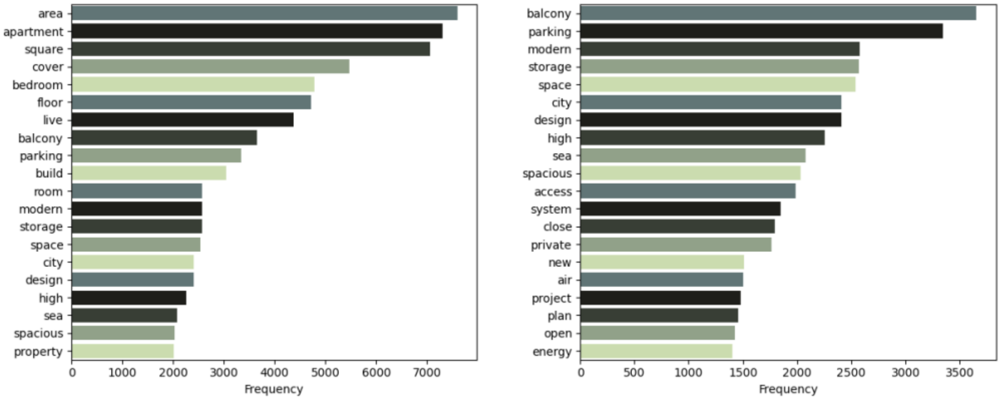
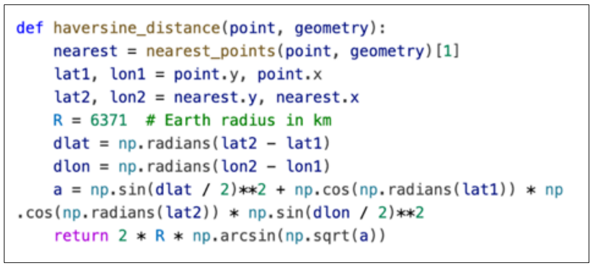


Figure 4. Most Frequent Unigrams Before and After Real Estate Stopword Removal

3.2.3 Feature Engineering

Feature engineering can improve model performance by embedding domain knowledge into the data. The goal was to replicate the reasoning a local expert might use when valuing a property, considering how location, landmarks, and neighborhood context affect price. To capture Limassol’s market dynamics, we introduced four spatial features: distance to the coast, City of Dreams Casino, and Four Seasons Hotel, along with a binary indicator for whether the apartment lies above or below the main highway.

All distances were calculated in kilometers using the Haversine formula, which accounts for the Earth's curvature and provides a more accurate estimate than straight-line methods. The implementation is shown in Figure 5.



```
def haversine_distance(point, geometry):
    nearest = nearest_points(point, geometry)[1]
    lat1, lon1 = point.y, point.x
    lat2, lon2 = nearest.y, nearest.x
    R = 6371  # Earth radius in km
    dlat = np.radians(lat2 - lat1)
    dlon = np.radians(lon2 - lon1)
    a = np.sin(dlat / 2)**2 + np.cos(np.radians(lat1)) * np
.cos(np.radians(lat2)) * np.sin(dlon / 2)**2
    return 2 * R * np.arcsin(np.sqrt(a))
```

Figure 5. Haversine distance function for proximity calculation

To construct these features, we required various geographical coordinates. While the apartment coordinates were obtained directly from the listings, all other spatial data were retrieved using the OpenStreetMap Overpass Turbo API.

Prior to feature creation, we excluded listings in Akrotiri, the UK-administered territory southwest of Limassol. Although technically within the Limassol district, the region is not part of the Republic of Cyprus and is, therefore, irrelevant to the local housing market. Listings from that area were less than 100 and so were removed. Figure 6 illustrates the spatial distribution of the apartment listings retained for our analysis. Black dots represent the retained listings; the red-shaded area marks excluded Akrotiri.

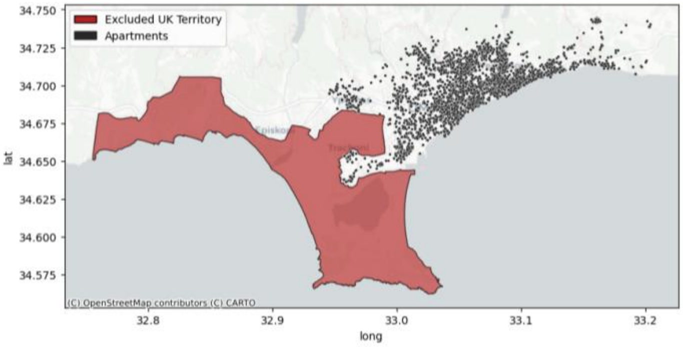


Figure 6. Geospatial distribution of listings

3.2.3.1 Proximity to Coast

Coastal living is a major draw in Cyprus, especially in Limassol, where the seafront is long, fully developed, and lined with sandy beaches, and the island's first skyscrapers. This area has become a status symbol, attracting both wealthy locals and foreign buyers. Based on this, we created a proximity-to-coast feature, assuming that closer distance to the coast drives up prices.

To measure this accurately, we focused on Limassol's urban seafront, shown in Figure 7. The black line represents the coastline segment used in the calculation. It begins at Marina Beach, just east of the Limassol Port, and continues eastward along the developed coastline. Although the coast technically extends further, we used the easternmost listing as a cutoff, since listing density drops off sharply beyond that point.

The western boundary starts at Marina Beach because the adjacent Limassol Port is industrial and offers no residential value. Further southwest, the coastline wraps around Akrotiri, which lies outside the Republic of Cyprus and is disconnected from the urban housing market. Including these segments would distort proximity values by making remote, irrelevant areas appear desirable.

The coastline was modeled as a dense sequence of GPS coordinates, and each apartment's proximity was calculated as the minimum distance to this line using the Haversine function presented in Figure 5. This approach captures practical, not just theoretical, access to the urban beachfront.

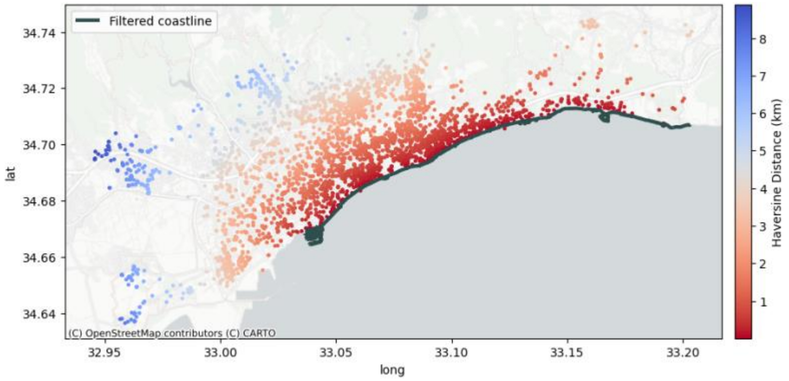


Figure 7. Proximity of listed apartments to the urban seafront

3.2.3.2 Proximity to Casino

The City of Dreams Mediterranean, shown in Figure 8, is Europe's largest integrated casino resort and one of Cyprus's most significant recent developments. Opened in 2023 in western Limassol after a €600 million investment, it includes a luxury hotel, entertainment district, and high-end retail. Its presence is accelerating real estate development in the area.

We include proximity to the casino to capture this emerging dynamic. As a new western anchor point and investment hotspot, it may drive capital appreciation in nearby neighborhoods. Listing density around the casino is high and fades eastward, possibly reflecting early-stage demand clustering.

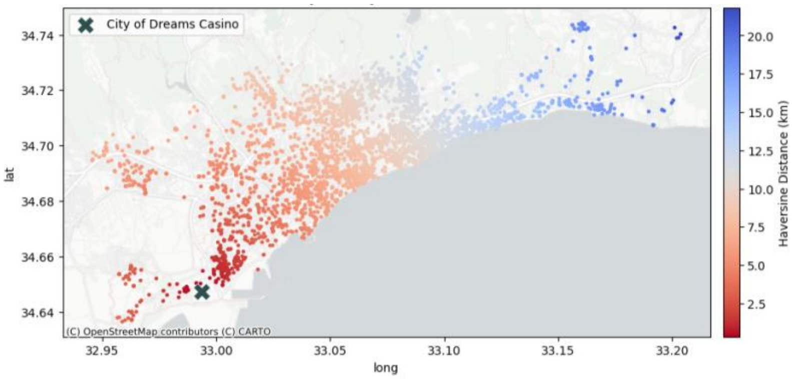


Figure 8. Proximity of listed apartments to City of Dreams casino

3.2.3.3 Proximity to Four Seasons Hotel

The Four Seasons Hotel, shown in Figure 9, is located in East Limassol, within Agios Tychonas, a long-established hub for luxury housing, five-star resorts, and upscale beachfront developments. Unlike the City of Dreams Casino, which reflects an emerging investment zone in the west, this area represents a mature and fully developed luxury cluster. We created a proximity feature based on the hotel's location to capture this concentration of high-end real estate. Together, the casino and the Four Seasons help the model capture price variation along Limassol's east-west luxury axis.

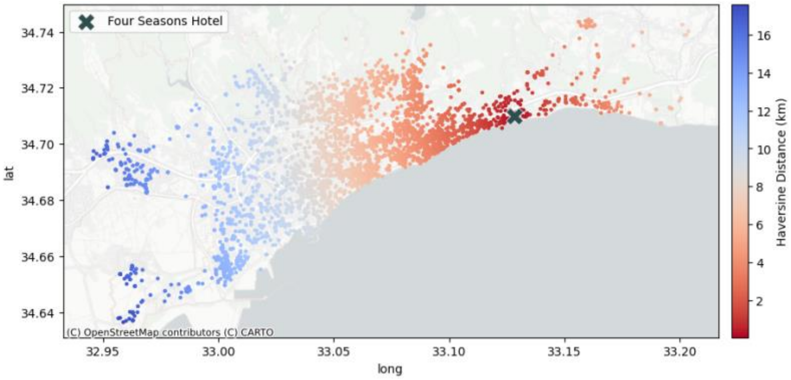


Figure 9. Proximity of listed apartments to Four Seasons Hotel

3.2.3.4 Position Relative to Highway

This feature is a binary variable indicating the property's location relative to the Limassol highway (A1/A6), where a value of 1 denotes properties situated north (above) the highway and 0 denotes those located south (below) it.

We hypothesize that two apartments with similar characteristics can have different prices depending on location. Areas above the highway are typically suburban, with more space and usually houses rather than apartments. In contrast, areas below the highway are denser and more urban, where proximity to amenities often drives higher demand and prices.

As shown in Figure 10, the black dotted line represents the highway, clearly separating the suburban areas above from the denser urban areas below. Listings appear more concentrated below the highway, suggesting that the urban-suburban divide may indeed exist. However, this pattern could reflect the sample rather than a true underlying division. Capturing this distinction could be important, as it may influence property values.

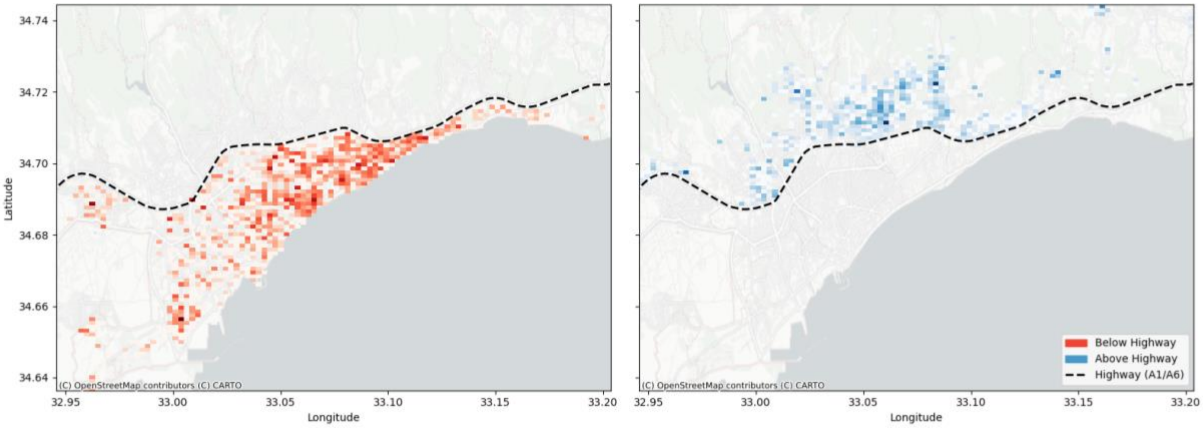


Figure 10. The density of Listings Above or Below the Highway

3.3 Exploratory Data Analysis

Prior to modeling, we explore the cleaned dataset to understand how the features behave and relate to the target variable. We start with the structured features, followed by a separate analysis of the textual descriptions.

3.3.1 Structured features

We begin the analysis with descriptive statistics to summarize the main characteristics of the data. Table 2 presents key summary values such as the mean, standard deviation, quartiles, and range for all structured features, along with counts for binary variables. The average listing price is €550,445, ranging from €128,000 to nearly €4 million, reflecting substantial variation across properties. Most apartments have two bedrooms, two bathrooms, and a median size of 89 square meters. Among the binary features, only a minority of listings are penthouses, including a pool, or are located above the highway.

Table 3. Summary of descriptive statistics

Feature	Count	Mean	SD	Min	Q1	Median	Q3	Max
Price	-	550,445	441,648	1280,00	299,000	420,000	630,000	3,950,000
Area	-	98	39	30	77	89	115	308
Floor	-	2	1	0	1	2	3	8

Bedrooms	-	2	1	0	2	2	3	6
Bathrooms	-	2	1	1	1	2	2	5
Latititude	-	34.698	0.018	34.636	34.689	34.7	34.711	34.744
Longitude	-	33.054	0.043	32.946	33.026	33.057	33.082	33.204
Age	-	3	7	0	0	0	2	45
Proximity to coast	-	2.36	1.85	0	0.78	2.1	3.32	8.9
Proximity to casino	-	8.41	3.5	0.33	6.02	8.42	10.54	21.82
Proximity to Four Seasons	-	7.31	3.69	0.02	4.46	6.7	9.6	17.62
Is above highway	1868							
Is penthouse	860							
Has pool	1476							

Number of observations: 5319

Next, we visualize the distributions of all continuous features to understand their behavior. Figure 11 shows kernel density plots for each variable in both their original form on the left and after applying a log transformation on the right. Many of these features, especially price and area, are heavily right-skewed, meaning most values are low with a few very high ones. This skewness can reduce modeling accuracy and make predictions more sensitive to outliers. We apply a log transformation to all continuous variables with positive values to address this. The result is more balanced, bell-shaped distributions better suited for modeling. Accordingly, we use the log-transformed versions of these variables in all subsequent analyses and models.

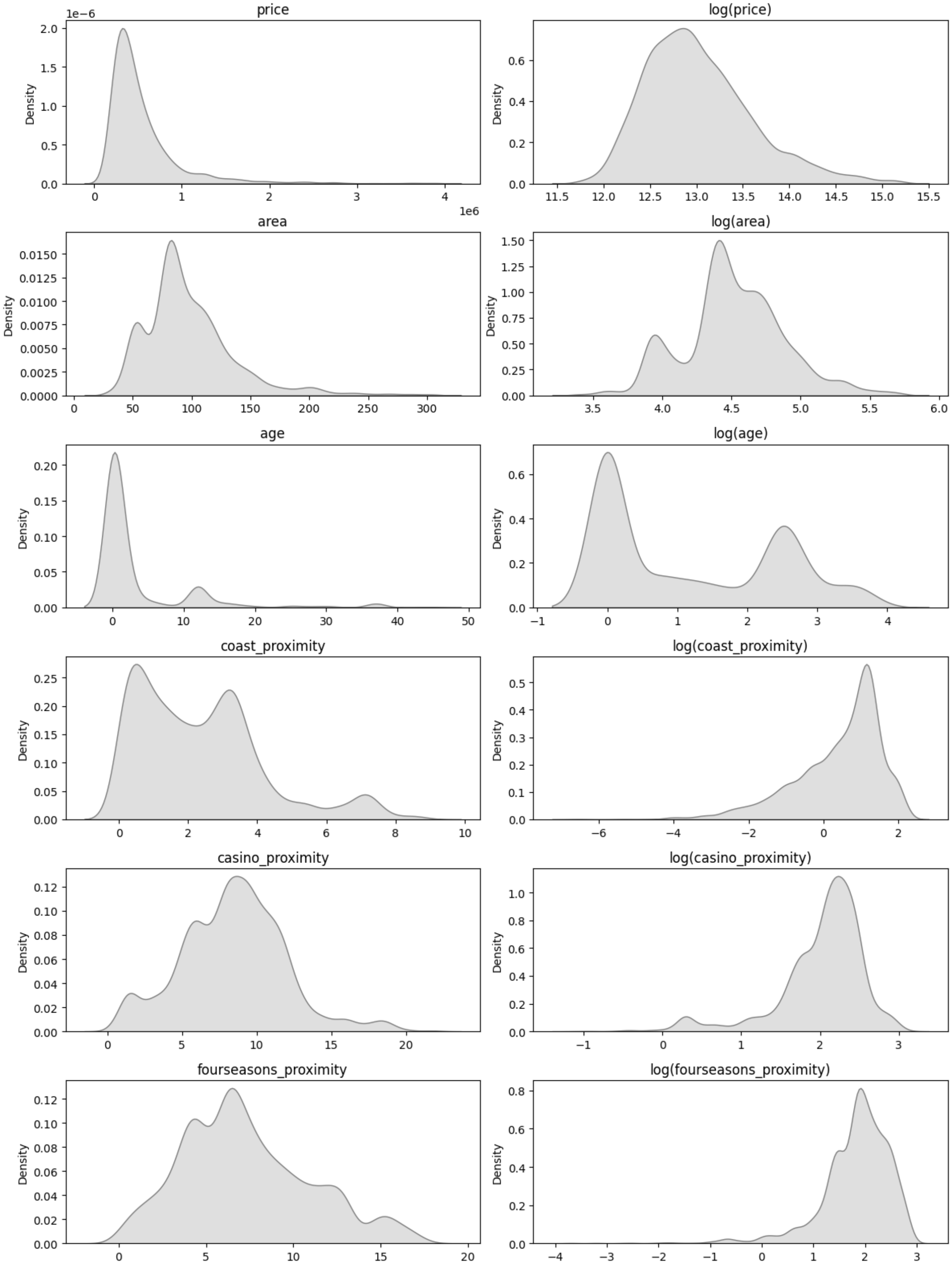


Figure 11. Kernel Density Plots of Continuous Features (Original vs Log-Transformed)

To examine the relationship between each feature and the target variable, we plot the natural logarithm of price (the target) against all structured features, using scatter plots for continuous variables and box plots for discrete and binary variables, as shown in Figure 13. Among these, one relationship stands out: the association between the natural logarithm of area and the natural logarithm of price. We highlight it separately in Figure 12 with a fitted linear regression line. While the pattern suggests a strong association, the R² value of 0.42 indicates that area alone explains only part of the variation in price.

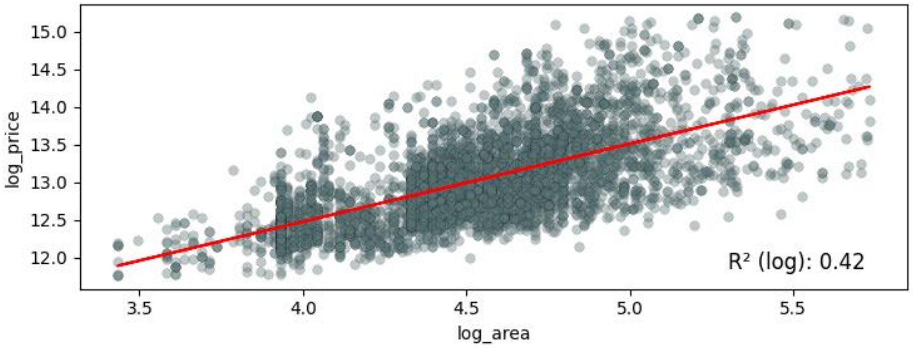


Figure 12. Relationship of log(price) vs. log(area) with linear fit

The remaining features show more complex patterns. Prices tend to decrease with greater distance from key landmarks and increase with structural characteristics like the number of bedrooms, bathrooms, and floor level. Binary features such as pool and penthouse status are associated with higher median prices, suggesting their potential role as indicators of luxury or premium properties. However, these relationships are often noisy, nonlinear, and overlapping. Even where trends are visible, they are not always strong or consistent.

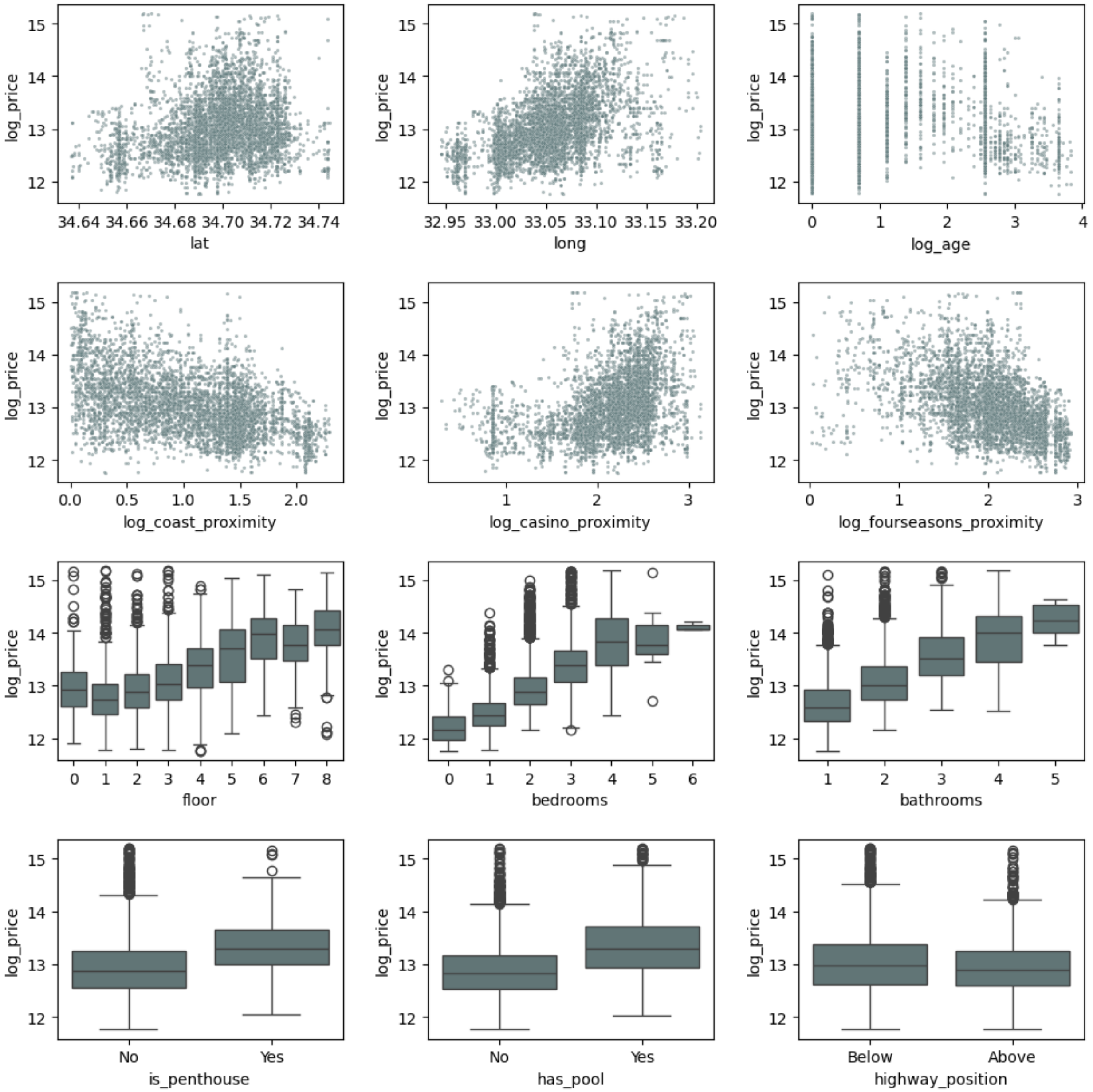


Figure 13. Relationships of structured features vs. log(price)

Moreover, we examine how the structured features relate to one another by calculating correlations, using Pearson correlation for continuous variables and point-biserial correlation for binary variables. Figure 14 presents the resulting correlation matrix. As expected, apartment area, number of bedrooms, and number of bathrooms show strong positive correlations with price and each other. Proximity to landmarks, such as the coast and the Four Seasons hotel, displays a negative correlation with price, indicating that properties closer to key attractions tend to be more expensive. Binary features such as pool access also show positive associations with price, while position relative to the highway shows little correlation.

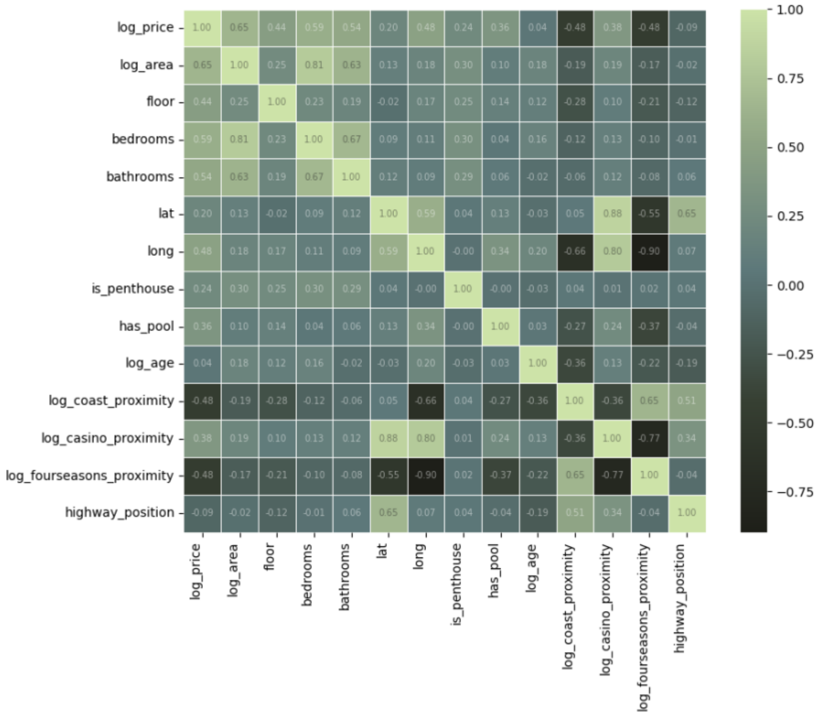


Figure 14. Correlation matrix of structured features

We observe multicollinearity among some predictors, particularly area, bedrooms, and bathrooms. While this can be problematic in interpretive models, our goal is to minimize prediction error which makes multicollinearity irrelevant in this context. In fact, the correlation analysis highlights meaningful associations with price, reinforcing the relevance of these features. Their interdependence supports the use of flexible machine learning models capable of capturing nonlinear interactions.

Overall, the exploratory analysis reveals two key insights. First, although area has the strongest relationship with price, it alone is not sufficient. That is, multiple features are needed to adequately capture price variation. Second, the relationships among features and with the target are often nonlinear and interdependent. These patterns align with the literature and further justify the use of machine learning over traditional linear approaches.

3.3.2 Textual features

This subsection provides a brief overview of the listing descriptions using a visual analysis. The goal is to assess whether text-based features can improve price prediction by capturing information not already reflected in the structured data. If properties across different price levels are described in similar ways, the text is unlikely to add value. After all, if both low-end and high-end apartments use the same language, there is no new signal for the model to learn from. On the other hand, if the descriptions reveal meaningful differences, such as sellers of higher-priced properties emphasizing premium features, it suggests that the text contains useful information. In that case, including it in the model becomes both logical and necessary to improve predictive accuracy.

To explore this hypothesis, we created word clouds for three price groups: the 5% cheapest listings, the 5% most expensive, and those priced near the median. If the descriptions reflect price differences, we should observe some noticeable variation in the language used. Figures 15 through 17 confirm that this is the case. In the cheapest listings (Figure 15), words like “balcony,” “project,” “modern,” and “close” are common, pointing to practical features and basic amenities. The most expensive listings (Figure 16) utilize terms such as “luxury,” “private,” “design,” and “sea,” which suggest a focus on lifestyle and exclusivity. Listings near the median (Figure 17) include more neutral words such as “parking,” “space,” and “modern.” This pattern supports our hypothesis: as price changes, so does the language used to describe properties. That suggests the text may indeed carry useful information as it pertains to price prediction.

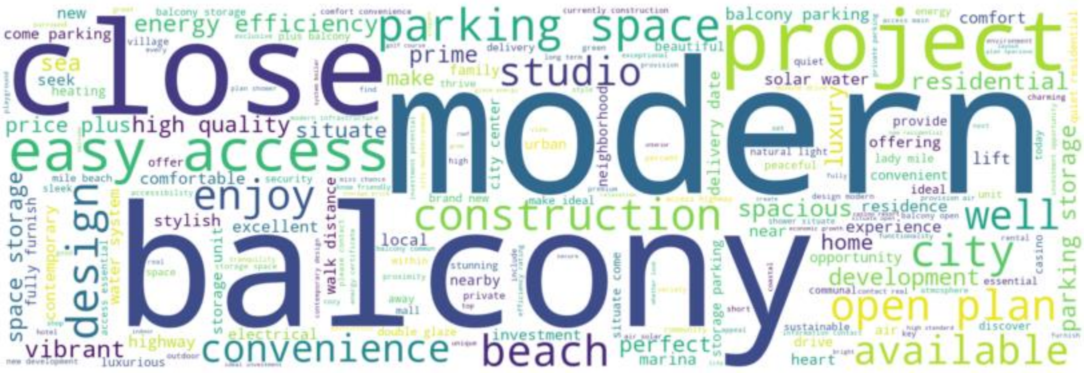


Figure 15. Wordcloud for 5% cheapest apartments

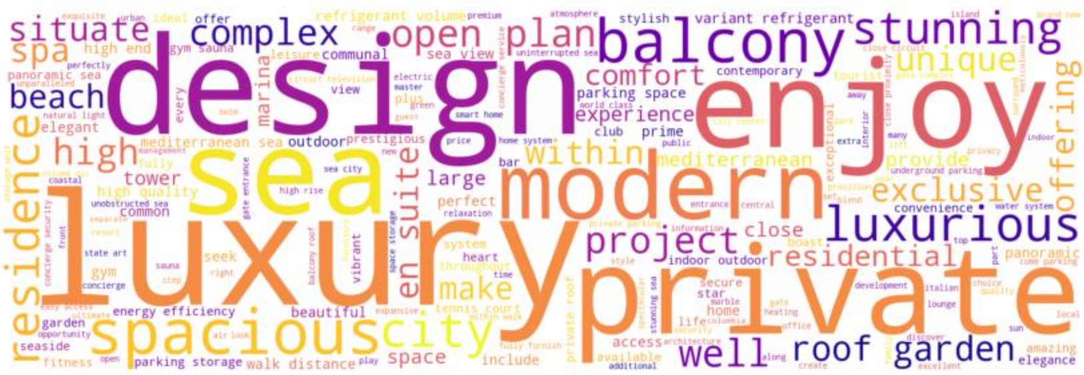


Figure 16. Wordcloud for 5% most expensive apartments

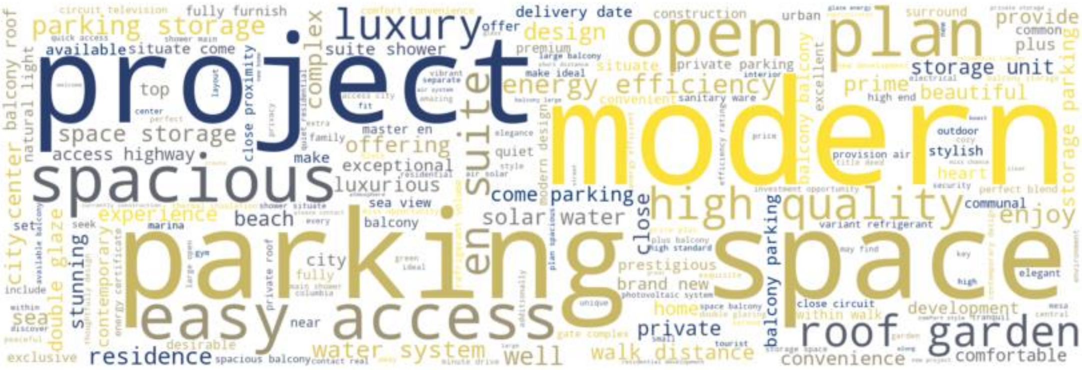


Figure 17. Wordcloud for median-priced apartments

Nonetheless, it is important to acknowledge a potential limitation. Real estate professionals write these descriptions intending to promote the property, which means the language used is likely not entirely objective. Certain features may be selectively emphasized or presented to enhance the property's appeal. This marketing-driven bias can influence the linguistic patterns our model learns from. While this issue is not examined in this study, its potential impact on the results is duly recognized.

3.4 Modelling

This subsection describes the modelling strategy used to predict apartment prices, including the selected algorithms and experimental setup (Aparicio et al, 2022). Two input configurations are evaluated: one using only structured features, and another combining structured features with text-based features extracted from listing descriptions.

The prediction task is framed as a regression problem, with the target variable being the natural logarithm of the apartment's listed price. Six algorithms are tested, selected for their demonstrated effectiveness in related studies and their ability to model different types of relationships. The chosen algorithms are the following:

- **Gradient Boosting (GB)** is an ensemble method that builds decision trees sequentially, with each tree correcting the errors of the previous ones. It is well-suited to structured data with complex, nonlinear interactions.
- **Random Forest (RF)** constructs multiple decision trees on bootstrapped samples and averages their outputs. It reduces variance and performs reliably with minimal tuning.
- **Support Vector Regression (SVR)** fits a function within a specified margin of tolerance and uses kernel functions to capture nonlinear relationships. It is particularly effective for smaller datasets.

- **Multilayer Perceptron (MLP)** is a neural network composed of one or more hidden layers, allowing it to approximate nonlinear functions. It is useful for learning moderately complex patterns in tabular data.
- **Deep Neural Network (DNN)** extends the MLP architecture by adding more layers, enabling it to learn hierarchical feature representations. It is capable of modelling high-dimensional, nonlinear relationships but requires careful tuning.
- **Multiple Linear Regression (MLR)** serves as the baseline hedonic model, predicting price as a linear function of the input variables. While limited in flexibility, it offers interpretability and serves as a reference point for evaluating more complex models.

All models were implemented in Python using Scikit-learn (Pedregosa et al., 2011), with the exception of DNN, which was implemented in PyTorch (Paszke et al., 2019) using the Skorch wrapper (Tschannen et al., 2019) to ensure compatibility with the Scikit-learn pipeline.

The dataset was split into 80% training and 20% testing, using a fixed random seed to ensure reproducibility. Hyperparameter tuning was performed using GridSearchCV with five repetitions of five-fold cross-validation, optimizing for mean squared error as the loss function (implemented as negative MSE for scoring purposes). Once the optimal hyperparameters were identified, each model was retrained on the full training set. The best-performing model was then selected for each input configuration.

In total, ten configurations were evaluated by applying five algorithms to two input types. In the combined setup, listing descriptions were converted into numerical vectors using word embeddings and concatenated with the structured features. The hyperparameter grid used for tuning is provided in Table 4. Parameters not listed were left at default values. Results from the tuning process are presented in the subsequent section.

Table 4. Grid search parameters and values

Algorithm	Parameter	Values	Description
GB	n_estimators	100, 300, 500, 1000	Number of boosting rounds. Higher values can improve performance but increase training time.
	max_depth	3, 4, 5, 6	Maximum depth of each tree. Controls model complexity and risk of overfitting.
	learning_rate	0.01, 0.1, 0.2, 0.3	Shrinks each tree’s impact. Lower values need more trees, but often generalize better.
	subsample	1, 0.8	Fraction of training data used for each tree. Adds randomness to reduce overfitting.
RF	bootstrap	True, False	Enables bootstrap sampling for building trees. Adds diversity to the ensemble.
	max_depth	5, 7, 10, 15, 30, None	Limits how deep trees can grow. None allows full growth until pure splits.
	max_features	10, 50, 100, 500, 1000	Sets how many features are considered at each split. Affects randomness and performance.

	n_estimators	'sqrt', 'log2'	Number of trees in the forest. More trees improve stability but increase training time.
MLP	hidden_layer_sizes	(64,), (128,), (64, 64), (128, 64)	Defines the size and number of hidden layers.
	activation	'ReLU'	Activation function applied between layers. ReLU helps with nonlinearity and efficiency.
	solver	'adam'	Training optimizer. 'adam' adapts the learning rate during training.
	learning_rate	0.001, 0.01	Starting learning rate. Affects how fast the model updates weights.
	alpha	0.001, 0.01, 0.1	L2 regularization term to prevent overfitting.
	batch_size	32, 64	Number of samples used per training step. Impacts speed and stability.
	max_iter	500	Max number of iterations. Training stops earlier if convergence is reached.
SVR	C	1, 10, 50, 100, 500	Regularization strength. Balances margin size and error tolerance.
	epsilon	0.005, 0.01, 0.05, 0.1, 0.2	The width of the margin where no penalty is given.
	kernel	rbf	Kernel type used to map data into higher dimensions.
	gamma	scale', 'auto'	Kernel coefficient. Controls influence of individual points.
DNN	batch_size	10, 20, 32, 64	Number of training samples used in each forward/backward pass.
	max_epochs	50, 75, 100, 150	Maximum number of full passes through the training data.
	learning_rate	0.0001, 0.001, 0.01	Initial learning rate for the optimizer.
	activation	'ReLU'	The activation function is applied to hidden layers (ReLU introduces nonlinearity).
	dropout_rate	0.1, 0.2	Fraction of neurons randomly dropped during training to reduce overfitting.
	neurons	50, 100, 200	Number of units per hidden layer, controlling model capacity.

The total number of experiments conducted in this study is 31,700. This includes 2,500 experiments for SVR, 2,400 for MLP, 6,000 for RF, 6,400 for GB, and 14,400 for the DNN. Each algorithm was evaluated across both input configurations and every unique hyperparameter setup was assessed using 5-fold cross-validation repeated five times, resulting in 25 runs per configuration.

3.5 Evaluation

We evaluated all models using three commonly used metrics in real estate prediction: Root Mean Squared Error (RMSE), the coefficient of determination (R^2), and Mean Absolute Percentage Error (MAPE). These metrics provide complementary insights into model performance.

Since the target variable was transformed using the natural logarithm of price, RMSE and R^2 were calculated in log space:

$$\mathrm{RMSE}_{\log} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_\imath})^2}$$

$$R_{log}^2 = 1 - \frac{\sum(y_i - \widehat{y_\imath})^2}{\sum(y_i - \bar{y})^2}$$

RMSE is useful because it expresses average prediction error in the same units as the target variable and penalizes large errors more heavily. However, when calculated in log space, RMSE is no longer in euros and cannot be directly interpreted. Exponentiating it to convert back to the original scale introduces bias due to the curvature of the exponential function, which typically results in an underestimation of the true error. The same limitation applies to R², which remains helpful for understanding the proportion of variance explained but does not reflect accuracy in monetary terms.

To overcome these issues, we also used MAPE, which was calculated after reversing the log transformation:

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\widetilde{y_\imath} - \widetilde{\widehat{y_\imath}}}{\widetilde{y_\imath}}\right|$$

MAPE expresses error as a percentage of the actual price, making it easier to interpret. For example, a MAPE of 10 percent means the model’s predictions are off by 10 percent on average, regardless of whether the apartment costs €200,000 or €2 million. This makes MAPE especially useful in real estate, where prices vary widely.

Finally, to examine whether prediction accuracy is consistent across the price spectrum, we divided the test set into ten price-based buckets and computed MAPE separately for each decile. This allowed us to assess whether the models perform equally well for lower and higher-priced properties.

4. Results

This section presents the experimental results. We begin by reporting the outcomes of the grid search on the training data, which were used to determine the optimal hyperparameters for each learning algorithm. Next, we evaluate the models based on their RMSE scores to identify the best-performing model for each input type: structured features only and those that combine structured features with textual features. The best-performing models for each input type are then applied to the test set to assess their performance.

4.3.1 Grid search results

The best validation RMSE and R² scores for all algorithm–input set combinations are reported in Figure 18. These scores reflect the performance of each algorithm using its optimal hyperparameters, identified through grid search. The corresponding hyperparameter values for each algorithm–input set combination are provided in Table 7 in the Appendix.

When using only structured features, GB, followed by RF, achieved the highest performance in terms of RMSE and R². MLP, DNN, and SVR performed moderately and did not match the tree-based models, even with extensive hyperparameter tuning, possibly due to their sensitivity to architecture and training data size.

With the addition of textual features, GB remained the best-performing model, with a slight improvement in RMSE and R². SVR also showed a more noticeable gain, while RF, MLP, and DNN performed worse compared to their structured-only versions.

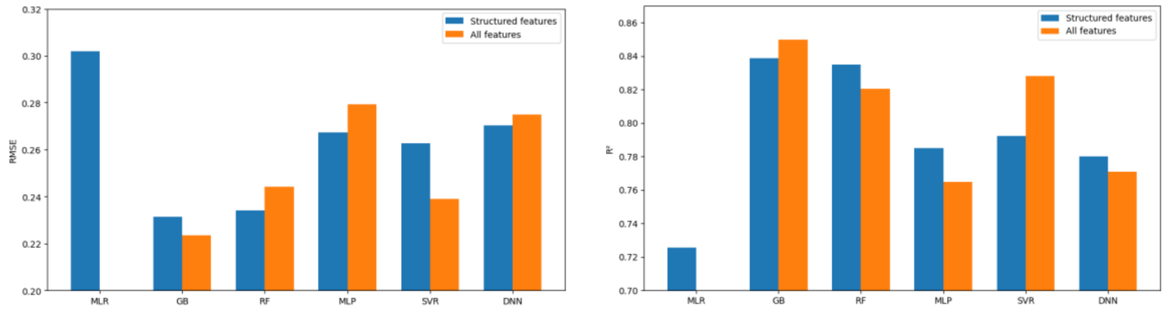


Figure 18. Validation RMSE and R² by Input Type and Algorithm

4.3.2 Final models' results

Based on the grid search results, GB performed better than all other algorithms across both input sets. We therefore proceed with two GB models that were tuned using the best hyperparameters: one trained only on structured features and one trained on both structured and textual data. Each model is evaluated on the test set to check how well it generalizes to new data. The selected setups are shown in Table 5.

Table 5. Top-performing models and their optimal parameters for each input set

Input set	Algorithm	Best parameters

Structured features	GB	learning_rate = 0.1,
		max_depth = 6,
		n_estimators = 500,
		subsample = 0.8

All features	GB	learning_rate = 0.1,
		max_depth = 5,
		n_estimators = 1000,
		subsample = 0.8

Table 6 shows the test results of the final models. The model using only structured features performed well, with a test R² of 0.836, RMSE of 0.229, and MAPE of 16.52 percent. When textual features were added, performance improved slightly: R² increased to 0.857, RMSE dropped to 0.214, and MAPE decreased to 15.59 percent. This means that, on average, predictions are off by about 15.6 percent. For example, for a property worth €1,000,000, the predicted price would usually be within ±€156,000. Both models fit the training data very well, and the small difference between training and test results suggests good generalization without excessive overfitting.

Table 6. Final models' results

Input set	Algorithm	train R²	test R²	train RMSE	test RMSE	train MAPE	test MAPE
Structured features	GB	0.992	0.836	0.054	0.229	3.99	16.52
All features	GB	0.995	0.857	0.041	0.214	3.15	15.59

We also examined how the model performs across different price segments. The test set was divided into ten deciles based on actual property prices, and MAPE was calculated for each, as shown in Figure 19.

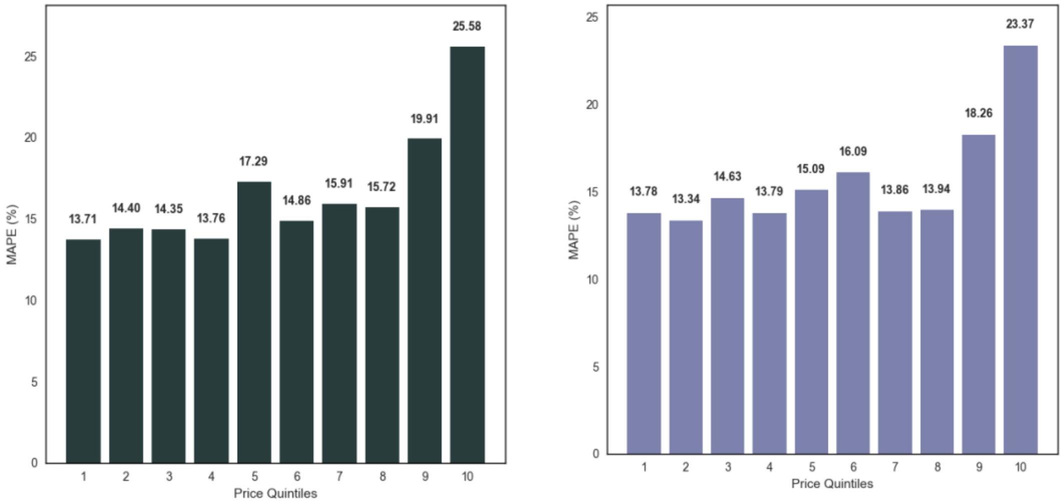


Figure 19. MAPE per price decile – Structured features vs. All features

The left side of the figure presents results for the model trained on structured features only. MAPE ranged from 13.71% to 25.58%, with relatively consistent performance across the lower and middle deciles (13.71% to 17.29%). However, error increased in the higher segments, rising to 19.91% in the ninth decile and peaking at 25.58% in the tenth. The right side of the figure illustrates the impact of adding textual features. While the overall pattern remained similar, MAPE improved across most deciles. The range narrowed to 13.34%–23.37%, with small reductions of 0.2–0.6 percentage points in the lower and middle segments. Improvements were more pronounced in the upper range: the ninth decile dropped from 19.91% to 18.26%, and the tenth from 25.58% to 23.37%.

To better understand the modeling challenge in the upper price segment, we also examined the variability of input features and the target variable across price quintiles. Figure 20 presents the accumulated standard deviation across all structured input features (left) and the standard deviation of the target variable, log price (right).

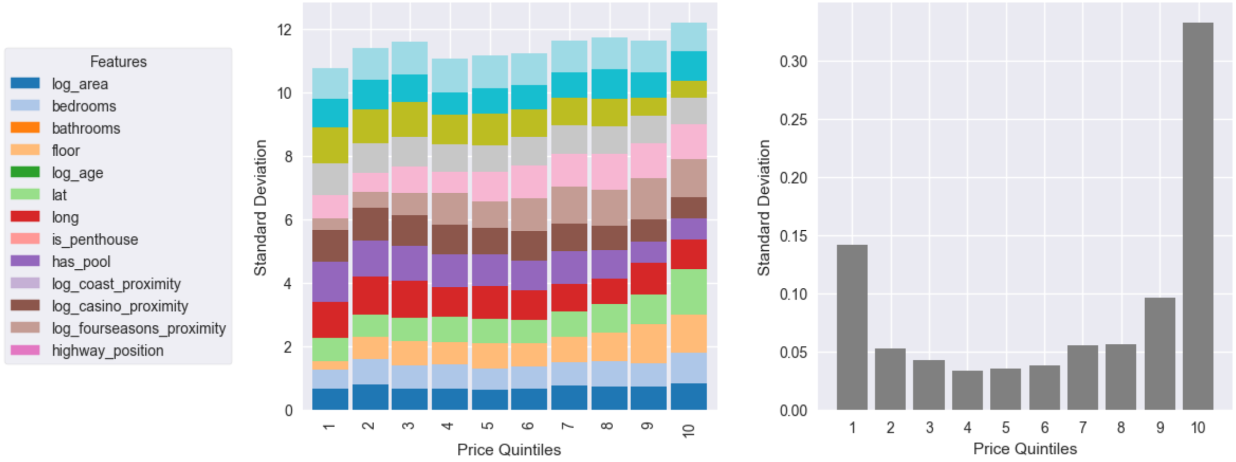


Figure 20. Feature and Target Variability Across Price Quintiles

The results show that input variability remains relatively stable across segments, while target variability increases significantly in the highest quintile. This indicates that, in the top segment, the model is required to predict over a broader output range without a corresponding increase in input variation.

As a complementary analysis, feature importance was examined based on the GB model trained only on structured property features. As shown in Figure 21, property size is clearly the most influential predictor, followed by proximity to the coast and longitude. This shows that size and location dominate value formation. Other features like floor level, property age, and landmark proximity also contribute moderately, while attributes such as being a penthouse or highway position have a very limited impact on price predictions in the Limassol market.

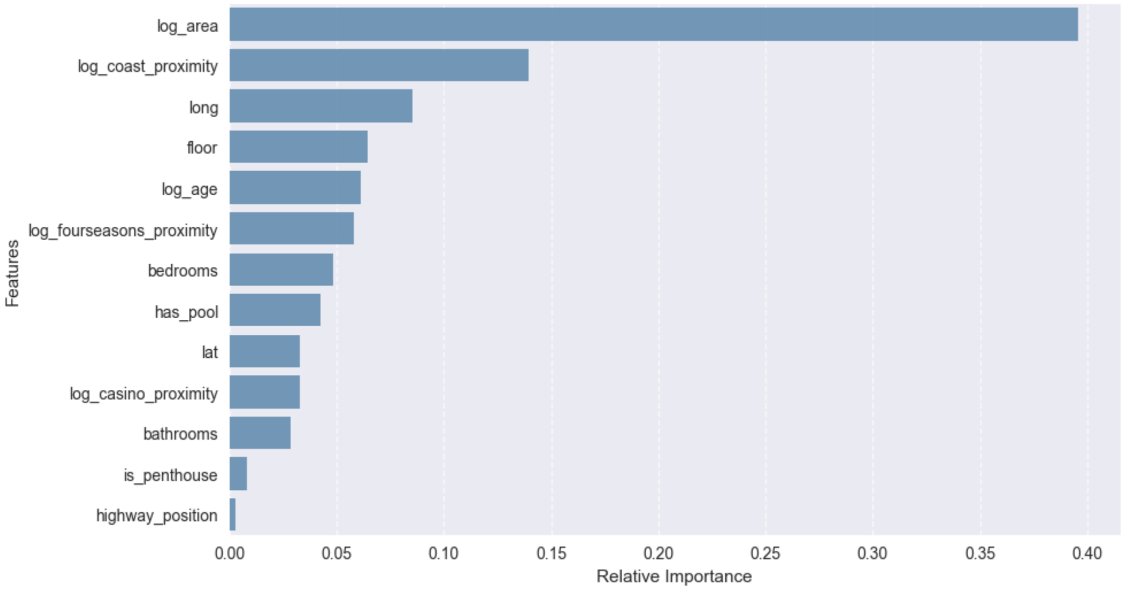


Figure 21. Feature Importance Based on Structured Property Attributes

5. Discussion

This section discusses the study's main findings, highlighting model performance, the contribution of textual features, differences across price segments, and how these results relate to existing research.

Firstly, all the machine learning models explored in this study showcased superior performance metrics than the benchmark linear regression model. This supports the idea, shown in past research (Peterson and Flanagan, 2009; Ho et al., 2020; Zhang et al., 2023), that machine learning methods are more effective than traditional hedonic models. These models are better at capturing complex patterns and relationships in the data, which makes them more suitable for predicting house prices.

Secondly, the results show that apartment prices can be predicted quite accurately using only basic property attributes. Gradient Boosting (GB), for example, achieved strong performance on the test set, confirming that features like property size, age, and location explain most of the price variation. Spatial features, such as how close a property is to the coast, were especially important. This is in line with what other studies have found (Hernes et al., 2024; Limsombunchai et al., 2004; Zaki et al., 2022), and confirms that structured data still holds strong predictive value, especially when location information is included (Frew & Wilson, 2002; Rey-Blanco et al., 2024).

Moreover, adding the textual data from property listings gave a small but consistent improvement in performance. The test R^2 increased and MAPE dropped slightly, with SVR showing the biggest benefit. This supports earlier studies (Abdallah, 2018; Nowak & Smith, 2016; Bushuyev et al., 2024), which also found that TF-IDF-based text features can improve prediction accuracy. However, the gains in our study were more modest. This may be due to the repetitive or vague language often used in Cyprus listings and the overlap between textual and structured features. For example, when both mention proximity to the sea. In addition, since the TF-IDF method used here relies on unigrams, it likely cannot capture context or distinguish between phrases like “sea view” and “no view,” which may limit its effectiveness.

Furthermore, GB maintained its lead across both input types, confirming its robustness. This matches earlier research (Zhang et al., 2023; Ja’afar et al., 2021) showing its consistent strength. Its learning process likely helped it deal with noisy or overlapping features better than RF, which performed worse when text was added. RF, although able to process sparse data, lacks the iterative refinement of boosting methods and may have been affected by the presence of low-signal or redundant features. For example, proximity to the coast is a structured continuous feature, while words like “beach” or “sea” often appear in listings located near the shoreline. This results in overlapping signals, where both structured and textual features capture similar location information. Even though care was taken during preprocessing to minimize such redundancy, some overlap is difficult to avoid, given the nature of how real estate descriptions are written. SVR, although not as accurate overall, gained the most from the text data, likely because it handles high-dimensional, sparse data well (Ho et al., 2020). MLP and DNN, on the other hand, performed worse with text, despite extensive tuning and regularization. This supports previous concerns (Root et al., 2023; Mostofi et al., 2022) that deep learning models may overfit when trained on small datasets with many input variables.

Another key finding is that model accuracy dropped in the top price decile. For the most expensive properties, performance worsened compared to the rest of the dataset, with MAPE increasing notably both with and without text. This shows how hard it is to predict luxury prices, where there are fewer listings and much more variation. Most studies report average performance across the whole dataset, but these results highlight that high-end listings may behave differently (Baldominos, 2018; Kalliola et al., 2021). Even though our dataset was too small to build separate models for each segment, this result suggests that future studies should look at segment-specific approaches for luxury homes.

Finally, our results support the conclusion from Zhang et al. (2024) that TF-IDF is a practical and effective text representation method. While more advanced methods like BERT or Word2Vec may work better in larger datasets with better-quality text, they also require more resources and domain-specific tuning. In our case, TF-IDF offered good results with much lower complexity, which is important for smaller markets like Limassol where data is inevitably less abundant.

6. Conclusion

Limassol's real estate market offers a compelling case for predictive modeling, combining rapid growth, varied demand, and a mix of objective and subjective factors. Despite this momentum, the local market's smaller size has kept it largely underexplored in research. This study examined whether apartment prices in this market can be accurately predicted using machine learning and whether listing descriptions provide additional value beyond structured features.

To investigate this, we scraped over 4,000 apartment listings from a leading Cypriot real estate platform and constructed a dataset combining structured variables such as size, location, and floor level with unstructured text descriptions. Location-based features were engineered using geographic data and tailored preprocessing was applied to both types of inputs. Text descriptions were converted into numerical vectors using TF-IDF based on unigrams.

Five machine learning algorithms were tested using grid search with five-fold cross-validation and repeated five times. GB consistently outperformed the alternatives. With only structured inputs, it achieved an R^2 of 0.836 and a MAPE of 16.52 percent. When listing descriptions were added, performance improved slightly, reaching an R^2 of 0.857 and a MAPE of 15.59 percent. This suggests that textual data can provide useful signals, but the overall impact remains modest.

Several limitations of the current study suggest directions for future research. The reliance on listing prices, which are shaped by seller expectations, agent strategies, and market sentiment, introduces label noise. These prices may differ from final transaction values, which were not available. Additionally, listings that remain active for longer periods, often because they are overpriced or unrealistic, are more likely to appear in the dataset. This creates a subtle imbalance that may affect how the models learn. Access to actual sale prices and time-on-market data would help improve label quality and reduce bias.

The textual descriptions also present challenges. They are usually written to persuade, rather than to objectively describe the property. Sellers may highlight positive traits and leave out negative ones. This makes the descriptions subjective and can mislead the models. There is also overlap between structured and textual features. For example, both may mention how close a property is to the sea. The models used unigrams, which means they could not understand phrases or context. As a result, they likely treated terms like "sea view" and "no view" as similar. Future studies could test n-gram models or more advanced text embeddings like Word2Vec or BERT to better capture meaning.

Regarding the models, deep learning approaches such as MLP and DNN were sensitive to the number of input features and showed a higher risk of overfitting. This was likely due to the combination of moderate dataset size and high dimensionality introduced by the textual features.

In addition, these models underperformed in the luxury segment, where listings are fewer and more diverse. Segmenting the market and training different models for each price range, especially for high-end properties, could improve accuracy. The current study also used a single snapshot of the market. Adding time-based variables, such as listing dates or economic indicators, could help capture market changes. Clustering methods could also help reveal hidden market segments. These segments could then be used as inputs for the models. Finally, combining structured and textual data using hybrid models may lead to better predictions.

Overall, our study demonstrates the feasibility and potential of applying machine learning to predict apartment prices in smaller fast-growing markets and highlights the added but still limited value of listing descriptions in this market.

7. References

Abdallah, S. (2018). An intelligent system for identifying influential words in real-estate classifieds. *Journal of Intelligent Systems, 27*(2), 183–194.

Adetunji, A. B., Oloyede, M. O., & Adewale, B. A. (2022). House price prediction using random forest machine learning technique. *Procedia Computer Science, 199*, 806–813.

Akyüz, S. Ö., Erdogan, B. E., Yıldız, Ö., & Ataş, P. K. (2023). A novel hybrid house price prediction model. *Computational Economics, 62*(3), 1215–1232.

Alfano, V., & Guarino, M. (2022). A word to the wise: Analyzing the impact of textual strategies in determining house pricing. *Journal of Housing Research, 31*(1), 88–112.

Antipov, E. A., & Pokryshevskaya, E. B. (2012). Mass appraisal of residential apartments: An application of Random Forest for valuation and a CART-based approach for model diagnostics. *Expert Systems with Applications, 39*(2), 1772–1778.

Aparicio, JT, Romao, M. & Costa, C. (2022) "Predicting Bitcoin prices : The effect of interest rate, search on the internet, and energy prices," *17th Iberian Conference on Information Systems and Technologies*, IEEE, https://doi.org/10.23919/CISTI54924.2022.9820085.

Baldominos, A., Blanco, I., Moreno, A. J., Iturrarte, R., Bernárdez, Ó., & Afonso, C. (2018). Identifying real estate opportunities using machine learning. *Applied Sciences, 8*(11), 2321.

Baur, K., Rosenfelder, M., & Lutz, B. (2023). Automated real estate valuation with machine learning models using property descriptions. *Expert Systems with Applications, 213*, 119147.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Bushuyev, S., Bushuiev, D., Kravtsov, D., Poletaev, N., & Malaksiano, M. (2024). Machine learning model for house price predicting based on natural language text data analysis. *Unpublished manuscript*.

Chau, K. W., & Chin, T. L. (2003). A critical review of literature on the hedonic price model. *International Journal for Housing Science and Its Applications, 27*(2), 145–165.

Court, A. T. (1939). Hedonic price indexes with automotive examples. In *The Dynamics of Automobile Demand* (pp. 99–117). General Motors Corporation.

Costa, C. J., & Aparicio, J. T. (2020). POST-DS: A methodology to boost data science. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–6). IEEE. https://doi.org/10.23919/CISTI49556.2020.9140932

Costa, C. J., & Aparicio, M. (2023). Applications of Data Science and Artificial Intelligence. *Applied Sciences*, *13*(15), 9015. https://doi.org/10.3390/app13159015

Deloitte Limited. (2023). *Cyprus Real Estate Review 2023.* Retrieved June 21, 2025, from https://www.deloitte.com/cy/en/Industries/realestate/analysis/cyprusrealestatereview2023.html

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (pp. 4171–4186). https://doi.org/10.48550/arXiv.1810.04805

Eurostat (2024). *House sales statistics*. Retrieved from https://ec.europa.eu/eurostat/statistics-explained on 21 June 2025.

Feldman, R., & Sanger, J. (2007). *The text mining handbook: Advanced approaches in analyzing unstructured data*. Cambridge University Press.

Frew, J., & Wilson, B. (2002). Estimating the connection between location and property value. *Journal of Real Estate Practice and Education, 5*(1), 17–25. https://doi.org/10.1080/10835547.2002.12091579

Glaeser, E. L., Gyourko, J., & Saiz, A. (2008). Housing supply and housing bubbles. *Journal of Urban Economics, 64*(2), 198–217.

Goodman, A. C., & Thibodeau, T. G. (1998). Housing market segmentation. *Journal of Housing Economics, 7*(2), 121–143. https://doi.org/10.1006/jhec.1998.0229

Herath, S., & Maier, G. (2015). Informational efficiency of the real estate market: A meta-analysis. *Journal of Economic Research, 20*(2), 117–168. https://doi.org/10.17256/jer.2015.20.2.001

Ho, W. K. O., Tang, B. S., & Wong, S. W. (2020). Predicting property prices with machine learning algorithms. *Journal of Property Research, 38*(1), 48–70. https://doi.org/10.1080/09599916.2020.1832558

Hong, J., Choi, H., & Kim, W. S. (2020). A house price valuation based on the random forest approach: The mass appraisal of residential property in South Korea. *International Journal of Strategic Property Management, 24*(3), 140–152.

Ja'afar, N. S., Mohamad, J., & Ismail, S. (2021). Machine learning for property price prediction and price valuation: A systematic literature review. *Planning Malaysia, 19*, 125–140.

Kalliola, J., Kapočiūtė-Dzikienė, J., & Damaševičius, R. (2021). Neural network hyperparameter optimization for prediction of real estate prices in Helsinki. *PeerJ Computer Science, 7*, e444.

Khani Dehnavi, M., Khani Dehnoi, M., & Ashrafi Amiri, H. (2025). Economic analysis of the real estate market using artificial intelligence. *International Journal of Applied Research in Management, Economics and Accounting, 2*(2), 56–70. https://doi.org/10.63053/ijmea.41

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML)* (pp. 1188–1196). PMLR.

Limsombunchai, V. (2004). House price prediction: Hedonic price model vs. artificial neural network. Lincoln University.

Malpezzi, S. (2002). Hedonic pricing models: A selective and applied review. In O'Sullivan, A., & Gibb, K. (Eds.), *Housing Economics and Public Policy* (pp. 67–89). Blackwell Science.

Manasa, J., Gupta, R., & Narahari, N. S. (2020). Machine learning based predicting house prices using regression techniques. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 624–630). IEEE.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv*. https://doi.org/10.48550/arXiv.1301.3781

Monson, M. (2009). Valuation using hedonic pricing models. *Cornell Real Estate Review, 7*, 62–73.

Mostofi, F., Toğan, V., & Başağa, H. B. (2022). Real-estate price prediction with deep neural network and principal component analysis. *Organization, Technology & Management in Construction, 14*(1), 2741–2759.

Nowak, A., & Smith, P. (2017). Textual analysis in real estate. *Journal of Applied Econometrics, 32*(4), 896–918.

Nouriani, A., & Lemke, L. (2022). Vision-based housing price estimation using interior, exterior, and satellite images. *Intelligent Systems with Applications, 14*, 200081.

Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert Systems with Applications, 42*(6), 2928–2934. https://doi.org/10.1016/j.eswa.2014.11.040

Paszke, A., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems, 32*.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of EMNLP* (pp. 1532–1543).

Peterson, S., & Flanagan, A. (2009). Neural network hedonic pricing models in mass real estate appraisal. *Journal of Real Estate Research, 31*(2), 147–164.

Rampini, L., & Re Cecconi, F. (2022). Artificial intelligence algorithms to predict Italian real estate market prices. *Journal of Property Investment & Finance, 40*(6), 588–611.

Rey-Blanco, D., Arbués, P., López, F. A., & Páez, A. (2024). Using machine learning to identify spatial market segments: A reproducible study of major Spanish markets. *Environment and Planning B: Urban Analytics and City Science, 51*(1), 89–108. https://doi.org/10.1177/23998083231166952

Root, T. H., Strader, T. J., & Huang, Y.-H. (2023). A review of machine learning approaches for real estate valuation. *Journal of the Midwest Association for Information Systems, 2023*(2), Article 2. https://doi.org/10.17705/3jmwa.000082

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24*(5), 513–523.

Samadani, S., & Costa, C. J. (2021.). "Forecasting real estate prices in Portugal : A data science approach," 2021 16th Iberian Conference on Information Systems and Technologies (CISTI), Chaves, Portugal, 2021, pp. 1-6, https://doi.org/10.23919/CISTI52073.2021.9476447

Sirmans, S., Macpherson, D., & Zietz, E. (2005). The composition of hedonic pricing models. *Journal of Real Estate Literature, 13*(1), 1–44.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199–222.

Tschannen, M., et al. (2019). Skorch: A scikit-learn compatible neural network library that wraps PyTorch. https://skorch.readthedocs.io

Vargas-Calderón, V., & Camargo, J. E. (2022). Towards robust and speculation-reduction real estate pricing models based on a data-driven strategy. *Journal of the Operational Research Society, 73*(12), 2794–2807. https://doi.org/10.1080/01605682.2021.2023672

Wei, C., Fu, M., Wang, L., Yang, H., Tang, F., & Xiong, Y. (2022). The research development of hedonic price model-based real estate appraisal in the era of big data. *Land, 11*(3), 334. https://doi.org/10.3390/land11030334

Zaki, M., et al. (2022). Gradient boosting for real estate price prediction: Empirical evidence from Egypt. *Unpublished manuscript*.

Zhang, Y., Li, Q., & Branco, R. (2023). A comparative study of text-based representations in real estate price prediction. *Unpublished manuscript*.

Zhao, H., & Wang, K. (2023). Predicting real estate price using stacking-based ensemble learning. *American Journal of Information Science and Technology, 7*(2), 70–75.

Zhao, Y., Chetty, G., & Tran, D. (2019). Deep learning with XGBoost for real estate appraisal. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1396–1401). IEEE. https://doi.org/10.1109/SSCI44817.2019.9002790

8. Appendix

Table 7. Algorithm results and best parameter settings from grid search

Regression algorithm	**Input set**	**Validation R2**	**Validation RMSE**	**Best parameters**
MLR	Structured features	0.7255	0.302	'fit_intercept': True
	All features	-	-	-
SVR	Structured features	0.7922	0.2627	'C': 1, 'epsilon': 0.2, 'gamma': 'auto', 'kernel': 'rbf'
	All features	0.828	0.2391	C': 10, 'epsilon': 0.01, 'gamma': 'auto', 'kernel': 'rbf'
RF	Structured features	0.8351	0.234	bootstrap': False, 'max_depth': 15, 'max_features': 'sqrt', 'n_estimators': 500
	All features	0.8204	0.2443	bootstrap': False, 'max_depth': 30, 'max_features': 'sqrt', 'n_estimators': 1000
GB	Structured features	0.8386	0.2314	learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 500, 'subsample': 0.8
	All features	0.8499	0.2234	learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 1000, 'subsample': 0.8
MLP	Structured features	0.785	0.2672	activation': 'relu', 'alpha': 0.1, 'batch_size': 64, 'hidden_layer_sizes': (128,), 'learning_rate_init': 0.001, 'max_iter': 500, 'solver': 'adam'
	All features	0.7648	0.2794	activation': 'relu', 'alpha': 0.1, 'batch_size': 32, 'hidden_layer_sizes': (128, 64), 'learning_rate_init': 0.001, 'max_iter': 500, 'solver': 'adam'
DNN	Structured features	0.78	0.2703	batch_size': 32, 'lr': 0.001, 'max_epochs': 150, 'module__activation': 'ReLU', 'module__dropout_rate': 0.2, 'module__neurons': 50
	All features	0.7708	0.2749	batch_size': 10, 'lr': 0.001, 'max_epochs': 150, 'module__activation': 'ReLU', 'module__dropout_rate': 0.1, 'module__neurons': 200