

Enhancing Automated Valuation Models with LLM-based Feature Engineering and Spatial Indices and a RAG-Integrated Web Application for Trójmiasto Residential Market

Piotr Kłopotowski
0009-0002-1775-4354
Gdansk University of Technology
Gdansk, Poland
Email: s184941@student.pg.edu.pl

Olgun Aydin
0000-0002-7090-0931
Gdansk University of Technology
Gdansk, Poland
Email: olgun.aydin@pg.edu.pl

Abstract—This paper presents a methodology for building robust Automated Valuation Models (AVMs) for residential properties in the Tricity (Gdańsk, Gdynia, Sopot) metropolitan area. Utilising a dataset of over 54,000 listings scraped over a nine-month period. We proposed an enriched feature engineering pipeline that integrates spatial indices (Points of Interest and public transport proximity) with unstructured data extracted via Large Language Models (LLMs). We evaluated the performance of XGBoost and random forest algorithms for both sale and rental price prediction. Results indicate that while traditional features like size and location remain dominant, the inclusion of custom spatial indices and LLM-extracted variables, such as utility fees for rentals, provides measurable improvements in model precision. Additionally, the paper focuses on a web application for analysing the residential market and outlines a practical application of RAG implementation for enhanced residential listing search based on the description of listings through the web application.

I. INTRODUCTION

THE housing market in Poland has experienced substantial fluctuations driven by socioeconomic changes, such as the recovery following the pandemic and the ongoing war in Ukraine. Because of these dynamics, there is an urgent demand for precise, real-time property valuation instruments. Conventional hedonic pricing models frequently fail to capture the fast-paced shifts and intricacies of these environments, prompting a shift toward Machine Learning (ML) approaches. This study centres on the Tricity region, a distinct polycentric market defined by a wide variety of housing submarkets, spanning from high-end coastal apartments to budget-friendly family homes [1].

Automated Valuation Models (AVMs) rely on Hedonic Pricing theory, operating on the premise that any product can be defined by n objective characteristics, where each holds varying levels of importance and subsequently exerts a distinct influence on the overall price [2]. Within the real estate domain, these characteristics typically include the total floor area, the room count, the construction year of the building, and the specific neighbourhood.

In the past, AVMs relied entirely on the foundational principles of hedonic pricing. Their primary benefit was simplicity, relying heavily on multiple linear regression and similar straightforward algorithms [3]. However, they also possessed notable drawbacks, particularly their susceptibility to heteroscedasticity [4]. These limitations, coupled with the necessity to incorporate variables that are harder to quantify objectively – such as distance to Points of Interest (POIs) or the quality of public transport connections – along with the fast-paced evolution of the property sector, necessitated the development of more sophisticated mass appraisal models capable of delivering higher accuracy [5], [6].

Modern AVMs often integrate Machine Learning models due to their superior capability in processing vast and diverse datasets [7], alongside their ability to handle non-tabular inputs like property images [8].

Existing studies highlight XGBoost-driven AVMs as the most precise. [9]. Furthermore, AVMs built on XGBoost can effectively incorporate spatial data regarding public transit and POI proximity [10].

Random forest is another highly effective ML algorithm for mass appraisal applications. In terms of RMSE, this method often outperforms AVMs relying on Neural Networks or Quantile Regression [11]. Nevertheless, thorough data pre-processing remains a strict requirement to guarantee the high performance of these models [12].

Both random forest and XGBoost have demonstrated significant success in real estate mass appraisals across major metropolitan areas, including Seoul, South Korea [13], China's evolving urban landscapes [14] and Istanbul, Turkey [15]. For this paper, we selected XGBoost and random forest to construct our AVMs due to their widespread recognition in the literature and their proven track record of delivering top-tier accuracy in comparable applications. Moreover, models employing these two algorithms can readily process spatial indicators. This includes the proximity of POIs, which demonstrably impacts property transaction prices [16], as well as

transport infrastructure connections. The latter is particularly relevant to our research, as non-bus transport networks often exert a distinct influence on valuations [17].

Property listings typically include numerous structured data fields viable for use as predictive features, such as the total area, room count, and district, since property portals mandate these inputs during the submission process. However, the vast majority of listings also feature a long, unstructured text description. These descriptions often hold additional details important to prospective buyers, like the presence of a balcony or the estimated monthly utility costs. Because this text lacks structure, extracting these specific features poses a considerable challenge. Manual extraction is an option, though it remains highly resource-intensive and time-consuming. We used an LLM-based methodology because it offers high reliability, low cost, and minimal initial configuration, proving that LLM-driven feature engineering is a practical solution for uncovering new variables in tabular prediction workflows [18].

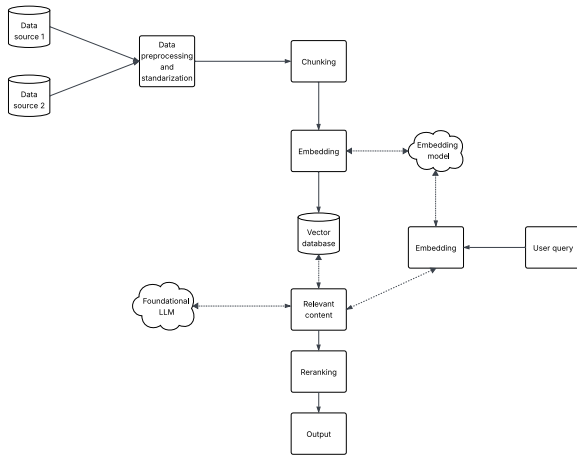


Fig. 1. A diagram of an example RAG system [19]

II. RETRIEVAL AUGMENTED-GENERATION (RAG)

Retrieval-Augmented Generation (RAG) offers a practical usage LLMs in production by supplementing models with domain-specific knowledge, as shown in Figure 1. These systems can process unstructured text, including articles, reports, and internal knowledge bases [19], [20]. We implemented a RAG architecture to evaluate its viability for an intelligent real estate recommendation system.

III. METHODOLOGY

A. Data Collection

We obtained data via automated web scraping from a major regional real estate portal covering the Tricity area. Over nine months, this process captured detailed listing attributes, including location, price, size, and unstructured text descriptions.

An ETL pipeline running every night was implemented to extract all Tricity advertisements. This system also tracked

price fluctuations and the duration each listing remained active on the platform.

The pipeline gathered both rental and sale listings alongside these core features:

- Price [PLN];
- Size [m²];
- Number of rooms;
- Floor level;
- Construction year;
- Description;
- 12 closest POIs;
- 3 closest public transit stops.

One additional feature applies exclusively to sale listings:

- Market type (primary or secondary).

Most listings include data on the 3 nearest transit stops and the 12 closest POIs across categories like retail, dining, and recreation. Although these details lacked granularity – for example, failing to distinguish between tram and bus stops, despite the fact that buyers usually tend to value tram stops more than bus stops – we collected them to evaluate their significance in AVM performance.

B. Spatial Feature Engineering

To measure the impact of local amenities, we created custom indices for POIs and public transit. The POI index (ϵ_{POI}) relies on the distance (d_i) to the 12 nearest POIs across various categories, including dining, education, and recreation.

$$\epsilon_{POI} = \sum_{i=1}^{12} \frac{1}{\max(d_i, 10)}$$

where d_i represents the distance to the i -th POI in meters. We formulated a comparable index (ϵ_{PT}) to capture the proximity and mode of public transit stations.

$$\epsilon_{PT} = \sum_{i=1}^3 \frac{\alpha}{\max(d_i, 10)}$$

C. LLM-based Feature Extraction

We utilised Large Language Models (LLMs) to extract structured features from unstructured listing descriptions. The models were prompted to identify specific attributes frequently omitted from standard data fields, such as balcony presence, parking availability, and monthly utility fees. This facilitated the conversion of natural language text into categorical and numerical data suitable for the machine learning pipeline.

To achieve this, we evaluated and compared the accuracy and feasibility of two LLMs. The first model, Bielik-7B-Instruct-v0.1 [21], was selected because it is specifically trained on Polish corpora and optimised for Polish prompts, aligning with the predominant language of the dataset. The second candidate was gpt-5-nano. Although not exclusively tailored for the Polish language, it demonstrates strong multilingual capabilities, effectively interpreting Polish prompts and providing high-quality responses at a relatively low completion cost.

To determine the most effective model, we manually annotated a random sample of 30 sale listing descriptions to establish ground-truth values. We then engineered a prompt instructing the LLMs to extract the following features:

- Binary features (evaluated as true if explicitly supported by the text, and false otherwise):
 - Presence of a balcony.
 - Availability of a parking space.
 - Presence of a basement.
 - Whether the apartment is ready for immediate move-in.
- Numerical features (extracted as a numeric value, or null if absent):
 - Monthly utility fees (in PLN).

Both models were evaluated on the 30-description sample using identical prompts. The comparative performance metrics are presented in Table I.

TABLE I
LLM METRICS COMPARISON ($N = 30$)

Model	Binary features				Numerical feature	
	Accuracy	Precision	Recall	F-score	Fee	exact match
Bielik	53%	51%	75%	61%		57%
GPT-5 Nano	84%	77%	95%	85%		100%

As observed, despite being explicitly designed for Polish text, the Bielik model exhibited significantly lower performance compared to the cost-effective GPT-5 Nano. Consequently, GPT-5 Nano was selected for the final feature extraction process across all sale listings.

IV. MODEL DEVELOPMENT AND RESULTS

Our literature review indicated these are the most prevalent algorithms and deliver superior accuracy in comparable tasks, outperforming both traditional hedonic models and other ML approaches like neural networks.

We processed 45,000 sale and 9,000 rental listings. Categorical variables, like district, underwent target encoding, and we engineered new features from the base dataset. We computed POI and public transit indices using the scraped distance data, and finally applied an LLM to extract specific attributes from the unstructured listing descriptions.

To enhance model quality, we excluded clear outliers, such as properties priced suspiciously low (under 100,000 PLN). These typically represent data entry errors, mislabeled non-residential spaces, or units with complicated legal situations (e.g., fractional ownership or existing tenants). We also removed listings under 15 m² – while Polish law sets the minimum apartment size at 25 m² [22], smaller micro-apartments are increasingly popular, justifying the inclusion of those above the 15 m² threshold.

We applied hyperparameter optimisation for both random forest and XGBoost. To prevent overfitting, we incorporated early stopping and validated the models via 5-fold cross-validation.

Both algorithms delivered strong results for sale listings, achieving $R^2 = 0.96$ with XGBoost and $R^2 = 0.95$ with random forest. For rental listings, both models reached $R^2 = 0.81$. These metrics confirm that the selected algorithms successfully explain a substantial portion of the price variance across both market segments.

A. Algorithm Comparison

Sale listing AVMs performed best for properties under 2,000,000 PLN, yet maintained a MAPE below 8% even for more expensive apartments. Overall, XGBoost outperformed random forest, particularly for higher-priced properties. Table II presents the validation results for both models grouped by price bracket.

TABLE II
PERFORMANCE OF THE MODEL FOR SALES PRICES: XGBOOST VS. RANDOM FOREST (UNITS IN THOUSANDS OF PLN)

Price Range	N	XGBoost			Random forest		
		RMSE	MAPE %	Bias	RMSE	MAPE %	Bias
250–600	3347	46.5	5.1	+12.6	49.2	4.7	+13.8
600–1,000	3672	62.1	4.4	+2.0	64.0	4.1	+4.0
1,000–2,000	1614	148.7	5.3	–11.2	149.3	5.6	–11.7
2,000+	287	377.3	6.6	–128.6	412.9	7.7	–175.6

Rental model results, summarised in Table III, indicate the most consistent performance in the 2,000–3,000 PLN segment, where both algorithms share a 267 PLN RMSE. Within this bracket, random forest shows slightly better precision (MAPE of 6.6% vs. 7.1% for XGBoost), though this is offset by a higher positive bias (+56 PLN vs. +49 PLN). Both models struggle with luxury rentals (5,000+ PLN), showing MAPEs over 12% and heavily negative biases, indicating a common tendency to undervalue high-end properties.

TABLE III
PERFORMANCE OF THE MODEL FOR RENTAL VALUE: XGBOOST VS. RANDOM FOREST (UNITS IN PLN)

Price Range	N	XGBoost			Random forest		
		RMSE	MAPE %	Bias	RMSE	MAPE %	Bias
1,000–2,000	239	379	14.9	+242	374	14.1	+239
2,000–3,000	1207	267	7.1	+49	267	6.6	+56
3,000–5,000	561	477	8.0	–134	458	7.5	–139
5,000+	69	1195	12.7	–627	1220	13.1	–697

B. Feature Importance (SHAP Analysis)

As shown in Figure 2, property size is the dominant driver of value across all four models, as indicated by a wide horizontal spread of SHAP values. For both sales and rentals, larger areas (pink dots) strongly increase prices, while smaller properties (blue dots) decrease predicted values. Location and building age serve as the primary secondary factors. Older buildings generally lower a property’s value (evidenced by high feature values on the negative SHAP axis), whereas modern builds and highly rated districts command a premium.

The models weigh secondary features slightly differently for sales versus rentals. Sale models heavily weigh floor level and room count, indicating buyer sensitivity to layout

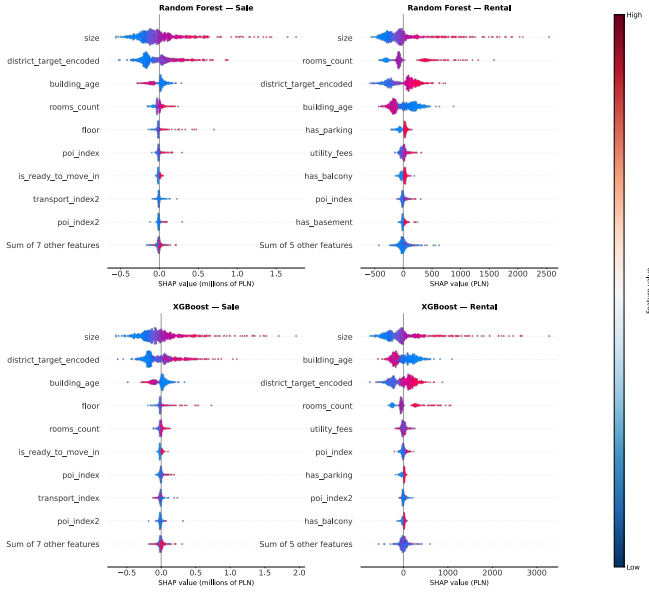


Fig. 2. SHAP beeswarm plots for all models

and elevation. On the other hand, rental models prioritise immediate utility, with room count becoming increasingly important. While LLM-generated features carry secondary weight, they remain significant predictors; the most impactful of these include utility fees, parking availability, and move-in readiness.

C. Performance Metrics

All four models demonstrated high reliability, especially in the most liquid market segments. Sale AVMs achieved strong explanatory power, with an R^2 of 0.96 for XGBoost and 0.95 for random forest. Despite a smaller dataset, rental models maintained a consistent R^2 of 0.81. Sale models kept MAPE below 8% across all brackets, though XGBoost better minimised RMSE and bias for properties over 2 million PLN. Rental models performed best in the mid-range (2,000–3,000 PLN) segment; performance naturally dropped in the luxury category (5,000+ PLN) due to a smaller sample size ($N=69$). Ultimately, XGBoost proved slightly better at handling variance in high-end listings, while random forest offered marginally better precision in the most common rental price brackets.

The predictive performance across all four models demonstrates high reliability, particularly within the most liquid segments of the market. For the sale listings, the models achieved exceptional explanatory power with an R^2 of 0.96 for XGBoost and 0.95 for random forest. The rental models, while operating on a smaller dataset, maintained a consistent R^2 of 0.81 for both algorithms. In terms of precision, the sale AVMs remained highly accurate with an MAPE below 8% even for high-value properties, though XGBoost proved to be better in minimising RMSE and Bias in the 2 million+ PLN bracket. In the rental market, both models proved most effective in the mid-range 2000–3000 PLN segment, though performance

naturally degraded in the luxury (5000+ PLN) category where sample sizes were significantly smaller ($N=69$). Overall, XGBoost marginally outperformed random forest in handling the variance of high-end listings, while random forest provided slightly better precision (lower MAPE) in the densest rental price brackets.

V. WEB APPLICATION

The developed real estate intelligence web application is a comprehensive platform designed to analyse property listings from a prominent portal specific to the Polish Tricity metropolitan area (Gdańsk, Gdynia, and Sopot). The application processes this data to deliver granular, neighbourhood-level market statistics, enabling users to monitor housing stock, price distributions, and listing decay rates across daily, weekly, monthly, and quarterly intervals.

A. Architecture of the web application

The Spring Boot backend provides a RESTful API for frontend communication and schedules nightly scraping tasks using CRON jobs. Extracted data is cleansed, transformed, and distributed across two data sources: MariaDB powers the analytics dashboard, while PostgreSQL with pgvector stores listing description embeddings for the RAG chatbot. The application utilizes Hibernate (JPA) for Object-Relational Mapping to streamline database operations. However, due to Hibernate's current limitations regarding vector columns, specific operations like fetching the k closest descriptions are handled via custom native queries. The frontend user interface communicates directly with the backend RESTful API, enabling users to explore the aggregated real estate data through interactive visualisations. Developed using the React framework, the UI integrates Leaflet to handle geospatial mapping, providing interactive, clickable district markers over the region. Additionally, the Chart.js library is utilised to render responsive, dynamic plots within the localised district dashboards.

B. RAG-powered search

We developed an RAG pipeline based on the description of the listings. During the ETL process, as each new listing is acquired, its text description is split into chunks and embedded.

When a user submits a natural language query, it is processed through the same embedding model. The system evaluates the query vector against the stored chunk embeddings using cosine similarity to identify the most relevant matches. Because a single listing document contains multiple sentence chunks, the retrieval mechanism first identifies the chunks with the highest similarity scores and subsequently retrieves their parent documents.

In order to guarantee high-quality results, rather than immediately returning the document containing the top-scoring chunk, we apply a final LLM-based reranking step. The top k retrieved documents are passed to a Large Language Model alongside the original user query and a strict system prompt,

tasking the model to evaluate the context and select the single listing that best satisfies the user's criteria.

Figure 3 presents the RAG module, which would allow users to bypass manual filters and search for properties based on intangible characteristics. By enabling natural language semantic search, the application can process highly specific requests – such as "a quiet apartment ideal for remote work" or "a sunny living room with a balcony." Consequently, this method aligns rigid database filtering with actual human intent, greatly enhancing the relevance and practical utility of property recommendations.

C. RAG Evaluation

Evaluation of the RAG module can be a gruesome task, as it requires setting up ground truths, which takes a significant amount of time. Ragas is a framework for automated RAG module assessment. It comes with various metrics, some of which do not require references (ground truths). Those include faithfulness, which measures how the result is relevant to the retrieved context and answer relevance, measuring the applicability of the answer to the given user query. On the other hand, it contains other metrics, such as context recall, that still need ground truths to assess the quality [23].

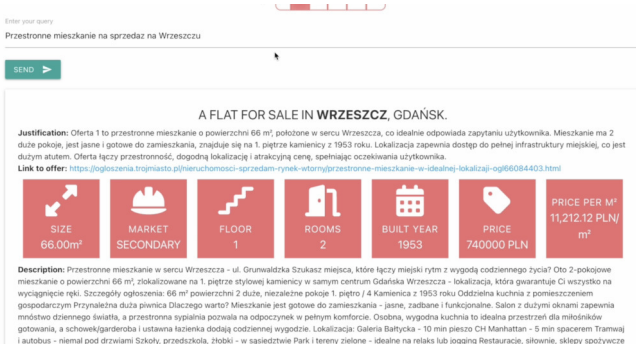


Fig. 3. Screenshot of example RAG query

In order to ensure the optimal performance of the RAG module, we have compared different combinations of settings. Each combination consisted of one of the following chunking strategies:

- Whole description becomes a single chunk;
- The description is divided into 200-character-long chunks;
- Each sentence is a separate chunk.

Additionally, one of the following embedding models was selected.

- *silver-retriever-v1* – model transforming Polish sentences and paragraphs into 768-dimensional vectors [24];
- *st-polish-paraphrase-from-distilroberta* – another model designed for the Polish language, also producing 768-dimensional vectors [25];
- *paraphrase-multilingual-MiniLM-L12-v2* – a multilingual model producing 384-dimensional vectors [26].

Finally, two LLMs were tested: *GPT-5 Nano*, as well as *Bielik-Minitron-7B-v3.0-Instruct* – a newer model from the aforementioned Bielik family, which also utilises Nvidia Minitron technology, allowing us to reduce the parameter count without sacrificing performance [27].

TABLE IV
RAGAS METRICS FOR EACH EVALUATION SETTING ($n = 20$ PER EACH SETTING).

Chunking	Embedding	LLM	Faithfulness	Answer relevancy	Context recall
Fixed 200 chars	Multilingual MiniLM	Bielik	29.7%	81.8%	26.5%
Fixed 200 chars	Multilingual MiniLM	GPT-5 mini	42.8%	77.8%	28.4%
Fixed 200 chars	Silver Retriever	Bielik	25.4%	80.4%	38.0%
Fixed 200 chars	Silver Retriever	GPT-5 mini	31.6%	78.3%	28.5%
Fixed 200 chars	ST Polish Paraphrase	Bielik	32.7%	80.4%	26.7%
Fixed 200 chars	ST Polish Paraphrase	GPT-5 mini	38.7%	77.1%	27.4%
Full description	Multilingual MiniLM	Bielik	48.0%	81.4%	41.7%
Full description	Multilingual MiniLM	GPT-5 mini	49.0%	78.5%	52.8%
Full description	Silver Retriever	Bielik	57.1%	81.0%	64.7%
Full description	Silver Retriever	GPT-5 mini	48.0%	82.4%	41.5%
Full description	ST Polish Paraphrase	Bielik	66.5%	81.1%	57.4%
Full description	ST Polish Paraphrase	GPT-5 mini	61.8%	81.3%	71.2%
Sentence	Multilingual MiniLM	Bielik	25.3%	75.5%	12.5%
Sentence	Multilingual MiniLM	GPT-5 mini	20.7%	78.6%	13.2%
Sentence	Silver Retriever	Bielik	21.2%	79.5%	17.1%
Sentence	Silver Retriever	GPT-5 mini	26.5%	82.6%	22.9%
Sentence	ST Polish Paraphrase	Bielik	23.1%	81.0%	17.7%
Sentence	ST Polish Paraphrase	GPT-5 mini	15.9%	66.5%	17.8%

Ragas scores, as reported in Table IV, show a clear gap between how well answers address the query and how faithfully they are grounded in retrieval. Answer relevancy remains high across settings (typically 77–82%, with a mean of about 79% across the 18 configurations), so the LLM responses generally stay on topic. Faithfulness is much weaker (16–66% per setting; about 37% on average) and depends strongly on chunking: full-description chunks reach 55%, fixed 200-character chunks about 34%, while sentence chunking falls to 22% (mean values), suggesting that shorter, fragmented context gives the judge less support for faithful answers. Context recall shows a similar pattern (17–71% per setting; about 34% on average): full description reaches 55%, fixed 200-character chunks 29%, and sentence chunking only 17% (mean values), indicating that finer-grained chunks often fail to surface enough ground-truth content. Differences between LLMs are small – GPT-5 mini is marginally stronger on faithfulness, whereas Bielik scores slightly higher on answer relevancy – and embedding choice has only a small effect (Silver Retriever leads slightly on answer relevancy and context recall). Overall, the chunking strategy appears to matter more than the choice of LLM or embedding model for retrieval-oriented Ragas metrics.

VI. CONCLUSION

This research demonstrates that combining advanced tree-based ML algorithms with innovative feature engineering – specifically spatial indices and LLM-based extraction – significantly enhances AVN accuracy in a complex metropolitan market like Tricity. While physical attributes of the property remain the most critical predictors, the ability of LLMs to "read" descriptions and capture hidden costs like utility fees adds a layer of intelligence to modern valuation engines. Future work will explore enriching the model with better-quality geospatial data that can be obtained by integrating

external geoinformation data sources, as well as computer vision to refine these models further.

Additionally, the research confirmed that both XGBoost and random forest are powerful ML algorithms when it comes to automated valuation tasks for the real estate market. Both have similar performance, except for high-end apartments, where XGBoost have slightly better predictions. Those models may be built upon data collected using web scraping technology. Furthermore, this research proves that they can successfully utilise LLM-extracted features retrieved from listing descriptions in order to further improve the prediction accuracy.

Finally, the research connects AVMs and RAG by LLMs to transform unstructured listing descriptions into actionable intelligence. For the AVM, LLMs extract hidden features like utility fees to improve price prediction accuracy. Simultaneously, the RAG system uses those same descriptions to enable semantic search for "intangible" buyer preferences that traditional filters miss. Together, they connect quantitative valuation with qualitative search utility.

VII. ACKNOWLEDGMENTS

During the preparation of this work, the authors used Bielik-7B-Instruct-v0.1 and gpt-5-nano models for feature extraction (as described in the *LLM-based Feature Extraction* section. Furthermore, Google Gemini was used for editorial correction, but the tool did not contribute to the research design, data analysis, or the core arguments presented. The authors take full responsibility for the content of this publication.

REFERENCES

- [1] O. Aydin and K. Zieliński, *Analysing the Residential Market Using Self-Organizing Map*. Cham: Springer International Publishing, 2022, pp. 465–478. ISBN 978-3-030-85254-2. [Online]. Available: https://doi.org/10.1007/978-3-030-85254-2_28
- [2] S. Rosen, "Hedonic prices and implicit markets: product differentiation in pure competition," *Journal of political economy*, vol. 82, no. 1, pp. 34–55, 1974. doi: 10.1086/260169
- [3] D. Tchuente, "Real estate automated valuation model with explainable artificial intelligence based on shapley values," *The Journal of Real Estate Finance and Economics*, pp. 1–39, 2024. doi: 10.1007/s11146-024-09998-9
- [4] M. Fletcher, P. Gallimore, and J. Mangan, "Heteroscedasticity in hedonic house price models," *Journal of Property Research*, vol. 17, no. 2, pp. 93–108, 2000. doi: 10.1080/095999100367930
- [5] A. Valier, "Who performs better? avms vs hedonic models," *Journal of property investment & finance*, vol. 38, no. 3, pp. 213–225, 2020. doi: 10.1108/jpif-12-2019-0157
- [6] P. Kłopotowski, "Building prediction engine for residential property prices in tricity," Master's thesis, Gdańsk University of Technology, Gdańsk, Poland, 2026, field of study: Informatics (Machine learning). Supervisor: dr Olgun Aydin.
- [7] H. Choi and H. Varian, "Predicting the present with google trends," *Economic record*, vol. 88, pp. 2–9, 2012. doi: 10.1111/j.1475-4932.2012.00809.x
- [8] O. Poursaeed, T. Matera, and S. Belongie, "Vision-based real estate price estimation," *Machine Vision and Applications*, vol. 29, no. 4, pp. 667–676, 2018. doi: 10.1007/s00138-018-0922-2
- [9] A. J. Pollestad, A. B. Næss, and A. Oust, "Towards a better uncertainty quantification in automated valuation models," *The Journal of Real Estate Finance and Economics*, pp. 1–37, 2024. doi: 10.1007/s11146-024-10002-7
- [10] Y. Zhao, J. Zhao, and E. Y. Lam, "House price prediction: A multi-source data fusion perspective," *Big Data Mining and Analytics*, vol. 7, no. 3, pp. 603–620, 2024. doi: 10.26599/bdma.2024.9020019
- [11] M. Steurer, R. J. Hill, and N. Pfeifer, "Metrics for evaluating the performance of machine learning based automated valuation models," *Journal of Property Research*, vol. 38, no. 2, pp. 99–129, 2021. doi: 10.1080/09599916.2020.1858937
- [12] P. Sobana, M. Balakumaran, S. Bharathkumar, P. Boopathi, and J. Harish, "House price prediction using machine learning," in *Challenges in Information, Communication and Computing Technology*. CRC Press, 2024, pp. 704–708.
- [13] J. Hong and W.-s. Kim, "Combination of machine learning-based automatic valuation models for residential properties in south korea," *International Journal of Strategic Property Management*, vol. 26, no. 5, pp. 362–384, 2022. doi: 10.3846/ijspm.2022.17909
- [14] Y. Jiang, L. Yue, S. Yang, and B. Yan, "Predicting real estate prices: A gradient boosting tree approach," in *Proceedings of the 2025 International Conference on Digital Transformation and Management*, ser. DTM '25. New York, NY, USA: Association for Computing Machinery, 2026. doi: 10.1145/3795546.3795587. ISBN 9798400721571 p. 288–293. [Online]. Available: <https://doi.org/10.1145/3795546.3795587>
- [15] M. Tekin and I. U. Sari, "Real estate market price prediction model of istanbul," *Real Estate Management and Valuation*, vol. 30, no. 4, pp. 1–16, 2022. doi: 10.2478/remav-2022-0025
- [16] R. Cellmer, M. Belej, and R. Trojanek, "Housing prices and points of interest in three polish cities," *Journal of Housing and the Built Environment*, vol. 39, no. 3, pp. 1509–1540, 2024. doi: 10.1007/s10901-024-10124-7
- [17] C. Chwiałkowski and A. Zydroń, "The impact of urban public transport on residential transaction prices: A case study of poznań, poland," *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, p. 74, 2022. doi: 10.3390/ijgi11020074
- [18] N. Abhyankar, P. Shojaei, and C. K. Reddy, "Llm-fe: Automated feature engineering for tabular data with llms as evolutionary optimizers," *arXiv preprint arXiv:2503.14434*, 2025.
- [19] S. R. Ahmad, "Enhancing multilingual information retrieval in mixed human resources environments: A rag model implementation for multi-cultural enterprise," *arXiv preprint arXiv:2401.01511*, 2024.
- [20] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz, "A survey on rag with llms," *Procedia computer science*, vol. 246, pp. 3781–3790, 2024. doi: 10.1016/j.procs.2024.09.178
- [21] K. Ociepa, Łukasz Flis, K. Wróbel, A. Gwoździej, and R. Kinas, "Bielik 7b v0.1: A polish language model – development, insights, and evaluation," 2024. [Online]. Available: <https://arxiv.org/abs/2410.18565>
- [22] Minister Infrastruktury, "Rozporządzenie ministra infrastruktury z dnia 12 kwietnia 2002 r. w sprawie warunków technicznych, jakim powinny odpowiadać budynki i ich usytuowanie," Dz. U. 2002 nr 75 poz. 690, t.j. Dz. U. z 2022 r. poz. 1225, § 94, Poland, 2002.
- [23] S. Es, J. James, L. E. Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," in *Proceedings of the 18th conference of the european chapter of the association for computational linguistics: system demonstrations*, 2024, pp. 150–158.
- [24] P. Rybak and M. Ogrodniczuk, "Silver retriever: advancing neural passage retrieval for polish question answering," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024. doi: 10.63317/364iqsayfxvs pp. 14 826–14 831.
- [25] HuggingFace, *sdadas/st-polish-paraphrase-from-mpnet*. [Online]. Available: <https://huggingface.com/sdadas/st-polish-paraphrase-from-mpnet>
- [26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.
- [27] R. Kinas, P. Kiszczak, S. P. Perez, K. Ociepa, Ł. Flis, K. Wróbel, and A. Gwoździej, "Bielik-minitron-7b: Compressing large language models via structured pruning and knowledge distillation for the polish language," *arXiv preprint arXiv:2603.11881*, 2026.