Investigating Predictive Techniques in House Prices using Machine Learning (ML) algorithms

LaTeX template adapted from:
European Conference on Artificial Intelligence

Giulio Dajani[1]

Other group members:
Nicola Ria[2] **Enis Yasharoglu**[3] **Mohamed Mohamed**[4]

Abstract. Housing price prediction is a significant challenge in real estate and urban planning. They affect where people live, their opportunities, and even the economy. This report presents a machine learning solution using XGBoost algorithm to predict housing prices based on features like salaries, population, and housing data. The project also explores alternative algorithms comparing their strengths and weaknesses, and it aims to enhance prediction accuracy and provide a starting point for real-world applications. [7]

1 Introduction

Housing price prediction can help individuals make informed choices, enabling policymakers to design better housing policies, and assist real estate investors in identifying profitable opportunities. However, accurate prediction of housing prices is very challenging due to the complex mixture of factors like income, population dynamics, and local amenities. [4] The goal of this project was to develop a predictive tool for housing prices using machine learning. By leveraging one of the best datasets from Kaggle [2] containing features such as median salary and population size, I aimed to design a model that could uncover meaningful patterns and provide actionable insights. To keep everything organized, I split my project into four main parts: Dataset, where the Kaggle files are stored; Outputs, where everything the project produces such as logs, trained models, and graphs are saved; Scripts, where the main logic is implemented, and XGBoost, downloaded from GitHub. [5]

2 Background

Predicting housing prices involves understanding complex relationships influenced by several factors and to tackle this challenge, I researched a variety of machine learning algorithms tailored for regression tasks highlighting their strengths and weaknesses, and focusing on three main characteristics: adaptability, performance, and interpretability. [3] The dataset sourced from Kaggle, offered a detailed view of London's housing market, including numerical and categorical features. Among the explored methods, Gradient Boosting Models stood out, with XGBoost excelling due to its speed, accuracy, and ability to handle missing data. In a similar way, LightGBM demonstrated efficiency and scalability with categorical features. Both models offered strong feature importance metrics. [6] An algorithm evaluated for its ability to model non-linear relationships using kernel functions is Support Vector Regression (SVR) but it struggles with large datasets. [1] Another way to achieve the project's goal is using Neural Networks (MLP), known for capturing intricate patterns [8] or CatBoost, which is the algorithm used by my colleague Nicola. It's an algorithm designed to handle categorical data natively reducing overfitting and could be mixed up with methods like Elastic Net Regression and Bayesian Ridge Regression. After analysing the strengths and weaknesses of all these methods and after comparing them based on interpretability and performance, XGBoost emerged as the best fit for this project. Its success in handling structured data and providing actionable insights through feature importance analyse made it an ideal choice, even though its sensibility with hyperparameter tuning.

3 Experiments and results

Modeling without preprocessing and feature engineering was impossible due to inconsistencies like missing values, mixed data types, and outliers in the Kaggle dataset which needed to be cleaned and transformed. Missing numerical values were replaced with the column median to minimize bias, and data types were standardized to prevent computational errors. Percentages in the 'recycling pct' column were converted into a numerical format, while categorical features like boroughs were converted into numerical equivalents. To better highlight the dataset's complexities and produce a higher accuracy percentage, derived features were introduced, including price-to-income ratio, used to measure housing affordability, relative house price, used to normalize house prices for borough-to-borough comparisons, income per job, used to represent economic productivity, and price-income interaction, used to combine housing prices, income, and population for capturing complex interdependencies. Additionally, a custom formula has been used to calculate the average house price, considering key factors such as salary, population, and housing density. The formula weighted these factors using coefficients fine-tuned through testing:

[1] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: gd2482b@gre.ac.uk

[2] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: nr2280i@gre.ac.uk

[3] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: ey0171c@greenwich.ac.uk

[4] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: mm2612t@gre.ac.uk

$$\begin{aligned}\text{avg\_house\_price} = {} & (\text{median\_salary} \times W_1) + (\text{population\_size} \times W_2) \\ & + (\text{no\_of\_houses} \times W_3) - (\text{area\_size} \times W_4) \\ & + (\text{borough\_flag} \times W_5)\end{aligned}$$

With the creation of a clean dataset the next step was training the model following six key steps: importing preprocessed, validated, and tested data and splitting them into features and target variables, implementing a systematic feature selection process with hyperparameter tuning; testing the elimination of low-importance features using Recursive Feature Elimination; dividing the dataset into training (84%) and testing (16%) sets after experimenting with different ratios to identify the most effective split; optimizing parameters like 'learning rate' (to control step size), 'max depth' (to manage tree size), subsample (to prevent over-fitting), and regularization terms ('reg alpha', 'reg lambda'). GridSearchCV was also tested to refine these parameters further. The model was evaluated using metrics like Root Mean Squared Error (RMSE), R² (coefficient of determination), and accuracy. XGBoost can be mathematically represented as an optimization problem, where the objective function is formulated as below.

$$\text{obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where K is the number of trees, L(yi, ˆyi) is the loss function, and (f) is the regularization term that is used to control the complexity of the model and prevent the model from overfitting. The trained model was saved in .ubj format for future deployment and its performance it's visible through three type of graphs.

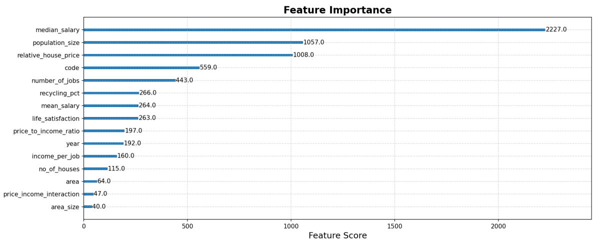

Figure 1. Feature Importance Graph

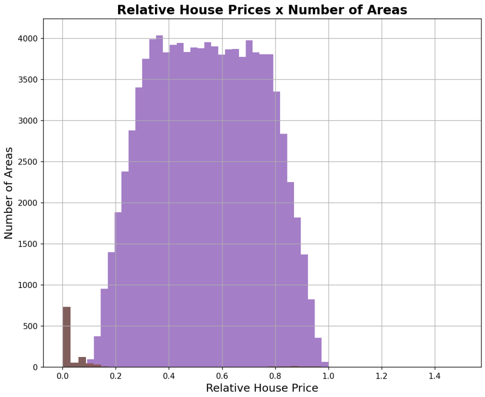

Figure 2. Relative House Price x Number of Areas

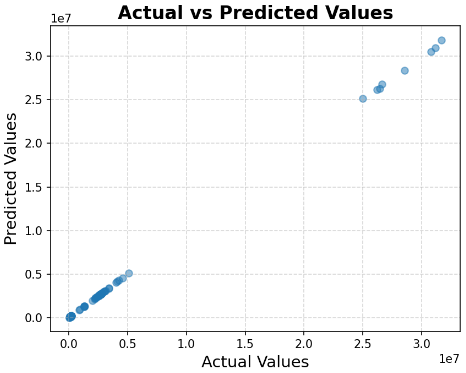

Figure 3. Actual vs Predicted Scatter Plot

To bring the project closer to real-world applications, a basic web application using Flask was initiated. The app takes user inputs in JSON format and predicts house prices in real-time using the trained XGBoost model. While still under development, it is a solid foundation for a practical tool for housing price predictions.

4 Discussion

One of the early challenges I have encountered was managing the amount of data displayed in the terminal. This was resolved by implementing a logging system, allowing detailed tracking of each step without displaying too much information over the terminal. The feature engineering process involved creating and testing over 30 derived features. Complexities emerged in managing missing data, which were addressed by median imputation, and categorical data, resolved through Label Encoding. Advanced techniques like early stopping, polynomial features, and recursive feature elimination were also explored to refine the model. XGBoost demonstrated high accuracy and robustness, outperforming traditional algorithms like Linear Regression. The use of advanced tuning techniques and hyperparameter optimization allowed the model to capture complex relationships in the data. Outputs like the feature importance graph and scatter plot confirmed the model's effectiveness in predicting housing prices.

5 Conclusion and future work

This project showed the power of machine learning in tackling real-world problems like housing price predictions. However, this is just the beginning. Besides deploying the web application for real-time housing predictions, the project opens the doors for several future enhancements. By including additional data like crime rates and transportation accessibility, the model's predictive power can be improved. Additionally, combining XGBoost with algorithms like CatBoost or Random Forest could help reduce errors further, providing a more comprehensive analysis. Implementing advanced techniques can refine hyperparameter tuning, while enabling the model to retrain periodically would ensure it stays up to date with evolving housing trends. The potential doesn't stop at London. Expanding the application to other cities or countries can make this tool a universal resource for individuals, policymakers, and investors. In a few words, this project highlights the possibilities of transforming data into meaningful predictions using the power of AI, focusing on the power of innovation and opening doors to a future of smarter, data-driven solutions.

REFERENCES

[1] Jirong Gu, Mingcang Zhu, and Liuguangyan Jiang, ‘Housing price forecasting based on genetic algorithm and support vector machine’, *Expert Systems with Applications*, **38**(4), 3383–3386, (2011).

[2] Justinas. Housing in london, 2024.

[3] CH Raga Madhuri, G Anuradha, and M Vani Pujitha, ‘House price prediction using regression techniques: A comparative study’, in *2019 International conference on smart structures and systems (ICSSS)*, pp. 1–5. IEEE, (2019).

[4] Joel Marsden, ‘House prices in london–an economic analysis of london’s housing market’, *Greater London Authority Economics*, **72**, 1–62, (2015).

[5] Hemlata Sharma, Hitesh Harsora, and Bayode Ogunleye, ‘An optimal house price prediction algorithm: Xgboost’, *Analytics*, **3**(1), 30–45, (2024).

[6] Racheal Sibindi, Ronald Waweru Mwangi, and Anthony Gichuhi Waititu, ‘A boosting ensemble learning based hybrid light gradient boosting machine and extreme gradient boosting model for predicting house prices’, *Engineering Reports*, **5**(4), e12599, (2023).

[7] Quang Truong, Minh Nguyen, Hy Dang, and Bo Mei, ‘Housing price prediction via improved machine learning techniques’, *Procedia Computer Science*, **174**, 433–442, (2020).

[8] Ayush Varma, Abhijit Sarma, Sagar Doshi, and Rohini Nair, ‘House price prediction using machine learning and neural networks’, in *2018 second international conference on inventive communication and computational technologies (ICICCT)*, pp. 1936–1939. IEEE, (2018).