Prediction of House Price with Extreme Gradient Boosting.

by

DINESH A/L MANIVANNAN
(211021094)

A report submitted in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering with Honours

Faculty of Intelligent Computing
UNIVERSITI MALAYSIA PERLIS

2025

UNIVERSITI MALAYSIA PERLIS

DECLARATION OF REPORT

Author's Full Name	:	DINESH A/L MANIVANNAN
Title	:	PREDICTION OF HOUSE PRICE WITH EXTREME GRADIENT BOOSTING.
Date of Birth	:	17 MARCH 2000
Academic Session	:	2024/2025

I hereby declare that this report becomes the property of Universiti Malaysia Perlis (UniMAP) and to be placed at the library of UniMAP. This report is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1997)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
☑	**OPEN ACCESS**	I agree that my report to be published as online open access (Full Text)

I, the author, give permission to reproduce this report in whole or in part for the purpose of research or academic exchange only.

Checked and approved by:

SIGNATURE OF STUDENT	**SIGNATURE OF SUPERVISOR**
000317-01-0769	ASSOC. PROF. DR. ZAHEREEL ISHWAR ABDUL KHALIB
(NEW IC NO. /PASSPORT NO.)	**NAME OF SUPERVISOR**
Date: 25 July 2025	Date
	Supervisor official stamp

NOTES : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with the period and reasons for confidentiality or restriction.

UNIVERSITI MALAYSIA PERLIS

PANEL APPROVAL AND DECLARATION SHEET

Author's Full Name	:	DINESH A/L MANIVANNAN
Title	:	PREDICTION OF HOUSE PRICE WITH EXTREME GRADIENT BOOSTING.
Date of Birth	:	17 MARCH 2000
Academic Session	:	2024/2025

This project report has been found satisfactory in terms of scope, quality and presentation as partial fulfilment of the requirement for the Bachelor of Engineering (Computer Engineering) in Universiti Malaysia Perlis (UniMAP).

Checked by:	Checked by:
SIGNATURE OF PANEL 1	**SIGNATURE OF PANEL 2**
NAME OF PANEL 1	**NAME OF PANEL 2**
Date:	Date:
Panel official stamp	Panel official stamp

ACKNOWLEDGMENT

First and foremost, I would like to express my sincere gratitude to the Almighty for granting me the strength, resilience, and determination to complete this Final Year Project successfully.

I am deeply indebted to my supervisor, Assoc. Prof. Dr. Zahereel Ishwar Abdul Khalib, for his exceptional guidance, insightful advice, and unwavering support throughout the course of this project. His expertise and constructive feedback have been instrumental in shaping both the direction and the depth of this research, particularly in the domain of machine learning and predictive modeling.

My appreciation also extends to all lecturers and staff of the Faculty of Electronic Engineering Technology, Universiti Malaysia Perlis (UniMAP), for their dedication to teaching and for providing the academic foundation that enabled me to undertake this work with confidence and competence.

I would also like to convey my heartfelt thanks to my beloved family and close friends. Their continuous encouragement, emotional support, and understanding have been invaluable throughout this journey.

Lastly, I wish to acknowledge my peers and everyone who has contributed—directly or indirectly through sharing knowledge, offering feedback, or providing motivation. Every form of assistance, no matter how small, played a vital role in the successful completion of this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programing Interface
CSV	Comma Separated Values
CSS	Cascading Style Sheet
CV	Cross-Validation
DT	Decision Tree
FYP	Final Year Project
GB	Gradient Boosting
GRIDSEARCHCV	Grid Search Cross-Validation
HTML	HyperText Markup Language
IDE	Integrated Development Environment
LASSO	Least Absolute Shrinkage and Selection Option
MAE	Mean Absolute Error
ML	Machine Learning
MLR	Multiple Linear Regression
MSE	Mean Squared Error
RF	Random Forest
RIDGE	Ridge Regression
RM	Ringgit Malaysia
RMSE	Root Mean Squared Error
R^2	Coefficient of Determination
SLR	Simple Linear Regression
SVM	Support Vector Machine
VSCODE	Visual Studio Code
XGBOOST	Extreme Gradient Boosting

LIST OF SYMBOLS

y	Actual house price (target value)
ŷ	Predicted house price (model output)
n	Total number of samples (data points)
∑	Summation symbol
X	Input feature set
f(X)	Prediction function (trained ML model)

Peramalan Harga Rumah Menggunakan Extreme Gradient Boosting

ABSTRAK

Peramalan harga rumah secara tepat merupakan satu cabaran penting dalam bidang hartanah, yang dipengaruhi oleh pelbagai faktor seperti lokasi, ciri struktur, dan dinamik pasaran. Kajian ini mencadangkan satu model pembelajaran mesin berasaskan pokok yang mantap menggunakan Extreme Gradient Boosting (XGBoost) untuk meramalkan harga rumah dengan ketepatan yang tinggi. Projek ini melibatkan proses prapemprosesan data yang menyeluruh, kejuruteraan ciri, latihan model, dan penilaian prestasi menggunakan set data dunia sebenar. Metrik penilaian utama seperti Ralat Mutlak Purata (MAE), Ralat Kuasa Dua Purata (MSE), Punca Ralat Kuasa Dua Purata (RMSE), dan R-kuasa dua (R^2) digunakan untuk menilai keberkesanan model. Selain itu, satu antara muka web telah dibangunkan untuk membolehkan pengguna memasukkan ciri-ciri rumah dan menerima ramalan harga secara masa nyata. Model akhir mencapai skor R^2 sebanyak 0.9117, yang menunjukkan keupayaan ramalan yang kukuh. Kajian ini bukan sahaja menekankan aplikasi praktikal teknik pembelajaran ensemble dalam analitik hartanah, tetapi juga menjadi asas untuk penyelidikan masa depan yang melibatkan set data yang lebih terperinci dan seni bina model hibrid. Penemuan ini bertujuan membantu pembeli, penjual, dan pembuat dasar dalam membuat keputusan yang tepat berasaskan data.

Prediction of House Price with Extreme Gradient Boosting

ABSTRACT

Accurate house price prediction remains a critical challenge in the real estate domain, influenced by a multitude of factors including location, structural features, and market dynamics. This study proposes a robust tree-based machine learning model using Extreme Gradient Boosting (XGBoost) to predict housing prices with high precision. The project involves extensive data preprocessing, feature engineering, model training, and performance evaluation using real-world datasets. Key evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) were utilized to assess model effectiveness. Additionally, a web-based interface was developed to allow end-users to input housing features and receive price predictions in real time. The final model achieved an R^2 score of 0.9117, demonstrating strong predictive capabilities. This work not only highlights the practical application of ensemble learning techniques in real estate analytics but also serves as a foundation for future research involving more granular datasets and hybrid model architectures. The findings aim to assist buyers, sellers, and policymakers in making informed decisions backed by data-driven insights.

CHAPTER 1 : INTRODUCTION

1.1 Project Background

The real estate market forms a fundamental pillar of any economy, influencing residents, investors, policymakers, and financial institutions alike. Accurately predicting property values is essential for informed decision-making, especially in an era of rapidly changing economic conditions and urban development. In recent years, machine learning has emerged as a powerful solution for real estate valuation, enabling models to uncover complex, non-linear relationships within housing datasets that traditional statistical methods may fail to capture. Among the various techniques available, Extreme Gradient Boosting (XGBoost) has gained significant attention due to its high predictive accuracy, speed, and ability to handle both structured and unstructured data.

XGBoost, a scalable and regularized form of gradient boosting, is particularly well-suited for regression tasks such as house price prediction. It integrates decision trees through a boosting framework and uses second-order derivatives for optimization, improving convergence and performance. Its built-in regularization also reduces overfitting, which is a common concern in predictive modeling. As a result, XGBoost has become a preferred model in Kaggle competitions and academic research involving tabular data, including real estate applications (Chen & Guestrin, 2016).

House prices are influenced by a wide range of factors, including property characteristics, neighbourhood attributes, proximity to public amenities, and macroeconomic indicators like interest rates and inflation. For instance, in the United Kingdom, the housing market has experienced dramatic shifts: in 2022, the average monthly mortgage payment accounted for approximately 26% of household income, up from 17% before the pandemic (Statista, 2023). This underscores the increasing complexity of housing affordability and the need for robust,

data-driven predictive models.

XGBoost is adept at analysing such multi-dimensional and dynamic data. It can assess the impact of location, distance to schools, crime rates, and local market trends simultaneously, offering practical insights for diverse stakeholders. Studies have shown that XGBoost outperforms traditional linear models and even other ensemble methods in real estate price estimation tasks (Fan, Ouyang, & Wong, 2021). By leveraging this technique, the current study aims to build a model that supports sellers in pricing their homes competitively, helps buyers make informed decisions, and equips policymakers with insights to design affordable housing strategies.

The success of this approach relies heavily on the quality of the input data, careful feature engineering, and rigorous model validation. This project proposes the use of tree-based model mainly XGBoost to predict house prices using real-world property datasets. While XGBoost mitigates many common challenges like overfitting and computational inefficiency, attention must still be given to issues such as data imbalance or bias. With a structured development pipeline, this research can contribute significantly to the field of property valuation and pave the way for further enhancements in future work.

1.2 Problem Statement

Accurately predicting house prices remains a persistent challenge worldwide due to the limitations of traditional valuation techniques and the increasingly complex nature of real estate markets. These conventional methods, which typically focus on basic features such as property size, location, and number of rooms, often overlook critical variables that can significantly affect housing prices. Factors such as the quality of renovations, proximity to schools and amenities, neighbourhood desirability, infrastructure developments, and even visual appeal from street-level imagery are frequently excluded from standard models, despite their real influence on buyer behaviour and perceived value (Law, Paige, & Russell, 2019).

This lack of feature depth and adaptability in traditional systems results in the consistent mispricing of properties. Overvaluation where properties are priced above their actual market worth can place undue financial strain on buyers, leading to inflated loan obligations and long-term debt burdens. On the other hand, undervaluation may result in sellers accepting lower-than-fair prices or deter investment altogether. Both cases introduce systemic inefficiencies that undermine market stability and buyer confidence (Tran, Le, Phuong, & Nguyen, 2025).

Furthermore, repeated mispricing can distort supply and demand expectations. Developers may interpret artificially high demand signals and overbuild in certain areas, only to face stagnant sales and mounting unsold inventory commonly referred to as the "property overhang" problem in markets like Malaysia and other rapidly urbanizing nations (Cellmer & Kobylińska, 2025).

Globally, housing affordability continues to deteriorate in many urban centres. Inaccurate pricing models contribute to this by allowing speculative pricing and excluding nuanced indicators that matter to everyday buyers. As a result, middle and low-income households often find themselves priced out of desirable neighbourhoods or forced into taking high-risk mortgages (Ouyang, 2024). This is compounded by outdated valuation systems that fail to incorporate evolving economic, environmental, and societal dynamics making them less responsive to modern housing market behaviour (Sellam, Distante, Taleb-Ahmed, & Mazzeo, 2024).

To address these challenges, it is imperative to move beyond conventional methods and embrace machine learning based approaches to house price prediction. Machine learning models can process vast amounts of data and identify complex, nonlinear relationships between features that traditional models cannot capture. By integrating a broader range of structured and unstructured features including those historically overlooked machine learning enhances predictive accuracy and provides stakeholders with reliable, data-driven insights. This contributes not only to better pricing transparency but also to more sustainable housing

development and informed policymaking (Tran et al., 2025; Ouyang, 2024).

1.3 Objectives

This project is developed with the purpose meet the highlighted objectives.

The objectives of this project are:

a) To design a tree-based machine learning model for predicting house prices using real-world dataset.

b) To develop a webpage that allows user to input feature details and get house price prediction.

c) To evaluate the performance of model using standard metrics which and MAE, MSE, RMSE and R squared.

1.4 Scope of work and limitations

The goal of this project is to utilize datasets to develop a tree-based machine learning model that can predict home prices. The goal of the study is to create a predictive model that can manage complex, non-linear connections among variables. Finding key factors that affect home prices, such as location, property size, and market movements, and applying these insights to train the model for accurate predictions are among the main goals. The project aims for high accuracy and clarity by using algorithms such as Random Forest, Gradient Boosting Machines, and Decision Trees.

The scope also includes evaluating the developed model's performance using common metrics like R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE). These metrics can help in evaluating the model's dependability and accuracy in predicting house prices in a range of market conditions. The project also attempts to understand the significance of characteristics such as neighborhood quality, economic conditions, and

access to conveniences that determine property values. The project provides a thorough method for overcoming difficulties in housing price prediction by directly connecting the scope to the goals.

Despite its promising potential, the plan has a number of disadvantages. The accuracy of the model is mainly dependent on the quality of the dataset (Kotsiantis et al., 2014). Predictions may be affected by incomplete or old data, like as missing property specifics or shifting market conditions. The model's performance may be limited by the availability of datasets with relevant properties. The ability to generalize of the model is another disadvantage. Despite their effectiveness, tree-based models have been known to overfit, especially when working with noisy or targeted datasets.

Additionally, training advanced algorithms like Gradient Boosting can be highly computational, particularly when working with big datasets (Mavaahebi & Nagasaka, 2013). This might limit scalability and necessitate a large amount of processing power (Mavaahebi & Nagasaka, 2013). Lastly, even while tree-based models can be somewhat processed, more complex combinations may make it more difficult for customers without technical knowledge to understand the findings. Despite these disadvantages, the project offers an opportunity for further research and helpful real estate analytics applications.

1.5 Expected result

The project's stated goals, which include developing a strong foundation for house price prediction using tree-based machine learning model, is closely aligned with the expected outcomes. First and foremost, the investigation wants to effectively create a tree-based predictive model that is adapted to actual housing statistics. In order to accurately predict house prices based on important aspects like location, property size, and market trends, model like Gradient Boosting are anticipated to identify intricate patterns in the data.

In terms of performance, the model is expected to demonstrate high accuracy and

precision when tested on relevant datasets. Consumers will benefit practically from this analysis, which will improve understanding of market dynamics.

Last but not least, the project expects evaluating the model using common performance metrics like R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE). These measurements will be used as standards to evaluate the accuracy and dependability of the predictions. In order to demonstrate low overfitting and strong generalization to new data, the model should exhibit consistent performance across training and validation datasets. The model's potential for real estate data analysis will be validated by this assessment, offering buyers, sellers, and industry experts a useful tool.

1.6 Thesis Layout

The Introduction chapter begins with a discussion of the background of the study, highlighting the significance of housing markets, challenges in accurately predicting house prices, and how machine learning techniques address these challenges. It describes the issue by providing relevant information or trends in changes in home prices. The project's exact goals are outlined in the issue description, which includes using tree-based models to increase prediction accuracy. The study's goals are quite clear: to create a tree-based machine learning model for price prediction, investigate the model's performance using important variables affecting home prices, and assess the model's accuracy using accepted metrics. While limits recognize challenges like dataset quality and computing limitations, the scope of work establishes the project's bounds and practical significance. The chapter concludes by outlining the expected results, which include increased prediction accuracy and significant understanding of the key factors influencing house prices.

The Literature Review chapter explores existing research, starting with an overview of

traditional and machine learning-based approaches. The use of machine learning in housing is then studied, with a focus on predictive analytics. The benefits, disadvantages, and usefulness of tree-based models such as Decision Trees, Random Forest and Gradient Boosting are thoroughly examined. A summary of the literature on house price prediction using tree-based models is provided by the review of prior work, which also reveals gaps that this project seeks to fill. Based on earlier studies, the chapter also outlines important variables that affect home prices, including geography, property size, and economic trends.

CHAPTER 2 : LITERATURE REVIEW

2.1 Introduction

This chapter explores machine learning model and method to solve inaccurate house price prediction Machine learning (ML) models have transformed the way many things work in the real world, making processes faster, smarter, and more efficient (Rane et al., 2024). For example, in healthcare, machine learning is used to assist doctors in diagnosing diseases based on medical pictures and predict how patients will respond to treatment (Chen et al. 2021). In finance, ML helps in the detection of fraud by detecting unusual patterns in transactions, and it is also used to analyse stock market movements (Magalhães et al. 2024). Companies such as Amazon and Netflix use ML to recommend products and videos based on what consumers enjoy, improving the user experience (Xia et al. 2022). ML also powers self-driving cars, allowing them to make quick decisions while navigating roadways safely (Rane et al., 2024).

Machine learning models are trained on data to recognize patterns and relationships, allow them to make predictions or decisions when presented with new data (Rane et al., 2024). Machine learning in real estate can predict house prices by analyzing key data factors, helping buyers and sellers to better understand the market. The accuracy of these predictions is determined by the quality of the dataset and how well it has been cleaned and prepared (Rane et al., 2024). Machine learning is also employed in various fields such as climate modelling, artificially intelligent assistants like SIRI, language translation, and personalized learning tools, demonstrating its practical applications

(Mathauer & Oranje, 2023). However, while machine learning has numerous advantages, it is equally important to address issues such as data privacy and fairness to ensure that technology is used ethically and responsibly.

2.2 Theoretical Background

A subfield of artificial intelligence (AI) called machine learning (ML) aims to educate systems to recognize patterns in data and draw inferences or forecast outcomes without requiring explicit programming for every task. Making algorithms that progressively learn from data is what it entails (Hong, T., Wang, Z., Luo, X., & Zhang, W. 2020). In short, a computer is "learning" if, based on its experiences (E), it becomes better at accomplishing a task (T) as measured by a specific performance metric (P). Machine learning (ML) works particularly effectively in complex and dynamic circumstances since the more data and experience the system has, the more accurate its predictions or decisions become.

Figure 2.1 Conceptual framework of machine learning as a process

(Mitchell, T. M. (1997)

Before the revolutions of AI, there were other methods used in the industry like traditional rule-based algorithms. Traditional rule-based algorithms perform by sticking to a predetermined set of rules that are manually coded to specify the behavior of the system. Until the code is updated, these rules stay the same and cannot be altered (Mexis, K., Xenios, S., & Kokosis, A. 2023).

However, machine learning algorithms are not dependent on predetermined rules. Instead, they create predictions or conclusions by using data to identify trends. The model becomes more versatile and capable of managing challenging jobs as it learns from examples and can automatically adjust when new data is added (Kumar, S., & Chakraborty, C. 2022).

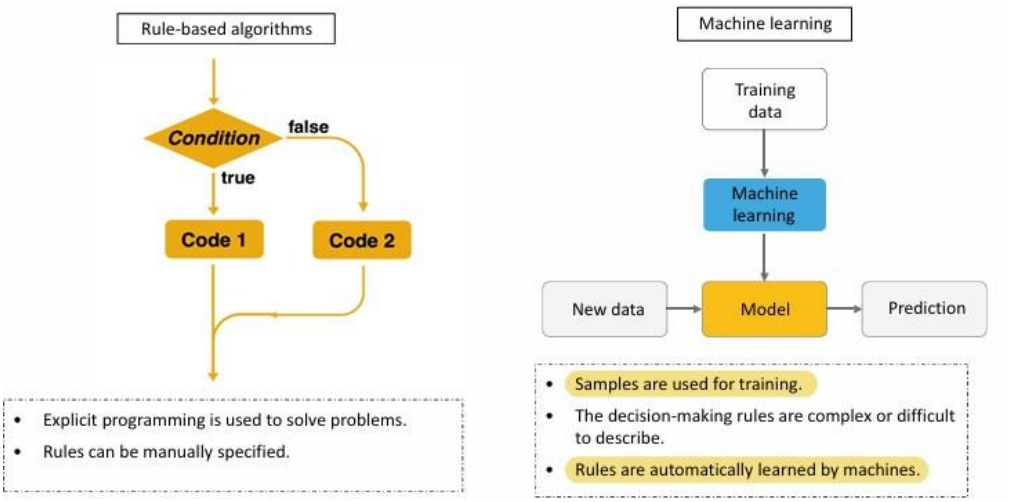


Figure 2.2 Difference between rule-based algorithms and machine learning algorithms (Rijcken et al. 2025)

2.3 Machine Learning model

2.1.1 Gradient Boosting

To improve prediction accuracy, a collective machine learning method known as gradient boosting employs many decision trees. To let the model to focus on difficult cases, it builds trees one after the other, with each new tree designed to correct the errors of the previous one. In this iterative process, gradient descent reduces the loss function, which assesses the degree to which the model's predictions deviate from the actual values. Gradient Boosting combines the output of all decision trees to provide a more accurate and dependable model than individual decision trees (Li, Z., Du, 2025).

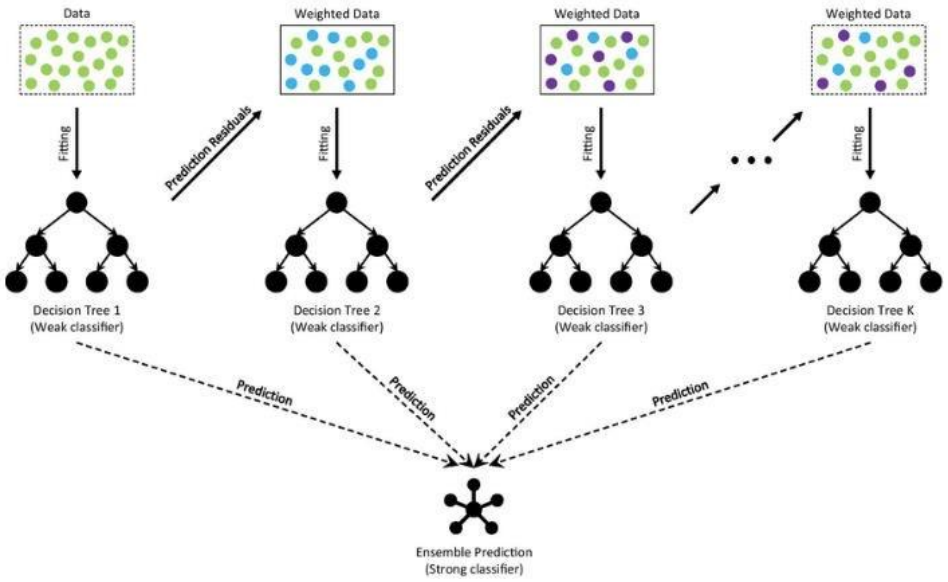


Figure 2.3 Structure of Gradient Boosting
(Deng et al., 2021)

Although gradient boosting performs exceptionally well, it can be computationally demanding and is sensitive to changes in hyperparameters. However, several advanced implementations, like Extreme Gradient Boosting, LightGBM, and CatBoost, have enhanced the algorithm's performance, making it faster and more efficient, especially for large datasets (Chen & Guestrin, 2016). These systems usually include regularization techniques to prevent overfitting and parallel processing to speed up training. Gradient boosting is particularly helpful for tasks like predicting home prices because, when properly adjusted, it can express intricate, non-linear relationships between input variables and yield remarkably accurate results.

2.1.2 Benefits and limitations

Because it iteratively builds decision trees that correct previous errors, gradient boosting is renowned for its exceptional predictive accuracy (Chen & Guestrin, 2016). Because of its versatility and ability to capture complex, non-linear correlations in data, it can be used for both regression and classification applications (Friedman, 2001). To help identify which factors have the biggest impact on the model's predictions, it can also assess feature relevance and provide regularization to prevent overfitting (Hastie et al., 2009). These benefits make it especially useful in real-world applications, such as home price prediction, where it excels at handling intricate data patterns.

Despite its benefits, gradient boosting has a number of disadvantages. It requires a lot of memory and processing time and can be computationally expensive, particularly when working with large datasets (Zhou, 2019). Because it is sensitive to hyperparameter settings, it also needs to be carefully adjusted to avoid overfitting and ensure optimal performance (Chen & Guestrin, 2016). Because of its complexity, the model is also challenging to understand because it operates as a "black box," making it challenging to provide particular predictions (Hastie et al., 2009). Finally, it may be challenging for Gradient Boosting to manage unbalanced data, requiring additional preprocessing steps to balance the dataset (Zhou, 2019).

2.3 Model Training

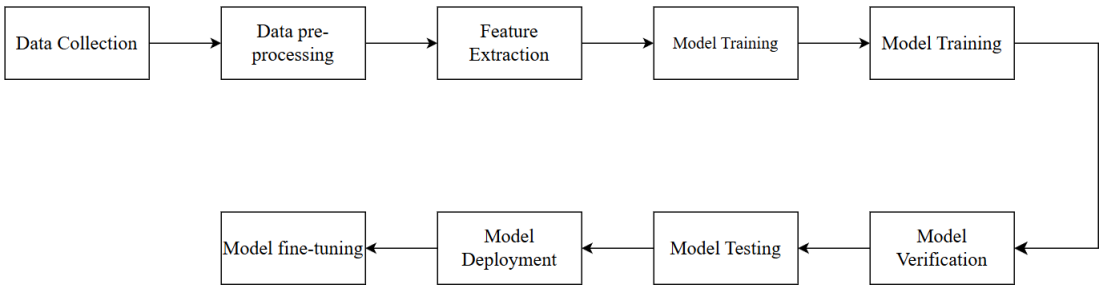


Figure 2.4 Block diagram of machine learning model training pipeline

1. **Data Collection**

   In this step, raw data is gathered from various sources such as databases, APIs, sensors, or web scraping. The data should be relevant to the problem being solved and comprehensive enough to train an accurate model.

2. **Data Pre-processing**

   Raw data is often noisy and inconsistent, so this step involves cleaning and organizing the data. Tasks include handling missing values, removing outliers, normalizing or scaling numerical data, encoding categorical data, and splitting the data into training and testing sets.

3. **Feature Extraction**

   This step identifies and selects the most relevant features (input variables) from the data that can improve the model's performance. Techniques like dimensionality reduction, feature selection, and engineering new features from raw data are applied.

4. **Model Training**

In this phase, a machine learning algorithm is used to train a model on the prepared dataset. The algorithm learns patterns and relationships in the data to make predictions or classifications.

5. **Model Verification**

   This step evaluates the trained model's performance by checking its MAE, RMSE, MSE or other relevant metrics. It ensures the model generalizes well on unseen data.

6. **Model Testing**

   The model is tested on a separate dataset (test set) to confirm its effectiveness in real-world scenarios. This step measures the model's ability to make accurate predictions on new data.

7. **Model Deployment**

   Once the model performs well in testing, it is deployed to a production environment where it can be used to make predictions or decisions in real-time applications.

8. **Model Fine-Tuning**

   Based on feedback from deployment and real-world performance, the model is fine-tuned. This might involve retraining the model with additional data, adjusting hyperparameters, or addressing issues that arise during deployment. After fine-tuning, the model goes through the training, testing, and verification loop again if needed.

2.4 Model Selection

The selection of an appropriate machine learning model is a crucial factor in ensuring the accuracy and reliability of house price prediction systems (Sharma et al., 2024). For this project, the decision to utilize Gradient Boosting is driven by its proven effectiveness in handling structured data and its ability to model complex, non-linear relationships between features (Li et al. (2025). This model is particularly well-suited for regression tasks, making it ideal for predicting continuous outcomes such as house prices. Its ability to capture intricate patterns provides a significant advantage over traditional linear models, which may struggle with the complex interdependencies commonly found in real-world housing data.

Gradient Boosting is a powerful ensemble technique that builds multiple decision trees sequentially, where each new tree focuses on correcting the errors made by the previous trees. This iterative approach enables Gradient Boosting to achieve high predictive accuracy, often outperforming other models in regression tasks. It excels at modeling subtle, non-linear dependencies between features, which are typical in housing market datasets where factors such as location, size, age, and amenities interact in complex ways (Zheng et al. 2025).

Although Gradient Boosting can be more computationally intensive than some simpler models, its ability to fine-tune predictions and minimize both bias and variance makes it an invaluable tool for house price forecasting. Furthermore, advanced implementations such as XGBoost, LightGBM, and CatBoost have optimized the efficiency of Gradient Boosting, making it faster and more scalable even with large and complex datasets (Zheng et al. 2025).

By leveraging Gradient Boosting, this project aims to develop a robust and accurate predictive model capable of handling the complexities inherent in real estate data. The model's strength in capturing non-linear patterns and delivering precise predictions supports the goal of

providing homeowners, buyers, and real estate professionals with reliable forecasts to inform decision-making.

2.5 Evaluation Metrics

For the house price prediction project, using a combination of evaluation metrics ensures a thorough and comprehensive assessment of model performance from multiple perspectives. Since this is a regression task, the primary goal is to minimize the difference between the predicted house prices and the actual market values. Selecting appropriate evaluation metrics is critical, as it directly influences how well the model's predictive capability is assessed and understood. In this project, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) have been chosen as the core evaluation metrics due to their proven effectiveness and suitability for regression tasks involving continuous numeric outputs such as house prices.

When compared to alternative evaluation metrics, such as Mean Absolute Percentage Error (MAPE) or Median Absolute Error, the chosen set of MAE, MSE, RMSE, and R^2 offers a more robust and reliable framework for house price prediction. MAPE, while intuitive in percentage terms, can produce unstable results for properties with very low prices due to its reliance on division by actual values, and can disproportionately penalize underpredictions over overpredictions. Metrics like Median Absolute Error may understate the importance of rare but significant outliers, which are especially relevant in real estate contexts where luxury properties can substantially influence market dynamics. The combination of MAE, MSE, RMSE, and R^2, therefore, strikes the optimal balance between interpretability, sensitivity to significant errors, and comprehensive model evaluation, ensuring that both the accuracy and explanatory power of the model are thoroughly assessed. This comprehensive approach supports the objective of developing a reliable and practical house price prediction model suitable for real-world applications.

2.5.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual prices. This metric provides a clear and interpretable measure of prediction accuracy, as it reflects the average error in monetary terms.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Figure 2.5 Formula of MAE

In this formula Where n is the number of observations in the data, y_i is the true value of the i-th observation, and ŷ_i is the predicted value of the i-th observation. The vertical bars indicate the absolute value, and the capital Greek letter sigma (Σ) indicates the sum of the differences. This gives a direct measure of how far off the predictions are from the actual values, without disproportionately penalizing larger errors (Limbong, 2025).

MAE is less sensitive to outliers, making it useful for evaluating overall model performance without being overly influenced by extreme price variations. Its simplicity and interpretability make it an essential metric for communicating model performance to non-technical stakeholders which is perfect in this case.

2.5.2 Mean Squared Error (MSE)

Mean Squared Error (MSE) is another key metric that squares the differences between predicted and actual values, emphasizing larger errors more than smaller ones. This characteristic makes MSE valuable for identifying models that may produce significant deviations in house price predictions.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Figure 2.6 Formula of MSE

In this formula y_i is the i^{th} observed value, $\hat{y}_i$ is the corresponding predicted value and n is the number of observations. This error is then squared, ensuring that both positive and negative errors contribute equally to the overall sum. Squaring the errors also magnifies larger deviations, penalizing models that produce significant prediction errors more heavily. The MSE reflects the overall accuracy of the model, with lower values indicating better performance (Mathotaarachchi, 2024).

By penalizing large errors, MSE ensures that the model focuses on minimizing outliers, which is crucial for improving overall prediction accuracy. However, since the error is squared, the units differ from the target variable, which can make direct interpretation less intuitive.

2.5.3 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) addresses this by taking the square root of MSE, resulting in an error value that is in the same unit as the target variable house prices. RMSE provides a more interpretable measure of model accuracy while still penalizing large errors.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Figure 2.7 Formula of RMSE

This formula where y_i is the actual value of the i-th observation, η_i is the predicted value of the i-th observation, and n is the number of observations in the data. The square root is used to scale the error to the same units as the original data, ensuring that all errors contribute positively and that larger errors are penalized more severely. The capital Greek letter sigma (Σ) represents the total of the squared differences. Similar to Mean Squared Error (MSE), these squared errors are averaged by dividing by n after being added up

over all n data points. The error value is returned to the same unit as the target variable in the final step, which takes the square root of this average, making RMSE more interpretable and practical for real-world applications (Mathotaarachchi, 2024).

Because it represents the model's performance in real-world scenarios, RMSE is very useful for predicting home prices. Because it strikes a balance between interpretability and sensitivity to significant errors, it is frequently regarded as the best indication of model accuracy.

2.5.4 R-Squared (Coefficient of Determination)

R-squared (R^2), or the coefficient of determination, measures how well the model explains the variance in house prices. A higher R^2 value indicates that the model captures most of the variability in the data, while a lower value suggests that the model may need improvement.

$$R^2 = 1 - \frac{RSS}{TSS}$$

Figure 2.8 Formula of R^2

The percentage of the dependent variable's variation that can be predicted from the independent variables in a regression model is measured statistically by the R-squared (R^2) metric. With a value between 0 and 1, it shows how well the regression model fits the data. Whereas an R^2 of 1 indicates that the model accurately predicts the dependent variable, an R^2 of 0 indicates that the model does not explain any of the variance. R^2 is computed as 1 minus the ratio of the total sum of squares (TSS) to the residual sum of squares (RSS), where TSS is the total variance in the data and RSS is the unexplained variance. It can be represented mathematically as follows: R^2 = 1-(RSS/TSS). Because it offers a relative performance metric, R^2 is essential for evaluating various model like Gradient Boosting (Mathotaarachchi, 2024). This statistic aids in choosing the model with the highest predictive power and the greatest fit to the data.

2.6 Model Development: Google Colab

Figure 2.9 LOGO of Google Colab

Google Colab (Collaboratory) is a cloud-based platform that enables users to develop and execute Python code in a Jupyter Notebook environment. It is widely used for data science, machine learning, and artificial intelligence projects. Colab provides free access to GPUs and TPUs, which significantly speeds up the training and evaluation of machine learning models (Carneiro, 2018). Since Colab runs in the cloud, there is no need to install software or manage dependencies locally, making it highly convenient for students and researchers working on complex projects.

For the Final Year Project, Google Colab is an excellent choice. It supports popular Python libraries like Scikit-Learn, which is essential for implementing tree-based algorithms such as Decision Trees, Random Forests, and Gradient Boosting Machines. These libraries can be easily installed and run in Colab, allowing you to experiment with different models and fine-tune hyperparameters to improve prediction accuracy. Additionally, Colab allows you to import datasets directly from Google Drive or GitHub, simplifying the process of loading and preprocessing housing data. Colab's interactive and collaborative nature makes it perfect for iterative model development. You can visualize data distributions, plot feature importance, and evaluate model performance in the same notebook, creating a seamless workflow for the project.

The ability to use GPUs ensures that even large datasets can be processed efficiently, which is crucial when dealing with housing data that may contain thousands of records. Another key advantage of Colab for model development is its scalability. Users can leverage Colab Pro or Pro+ for extended runtimes and more powerful hardware, which can significantly speed up training large datasets. The platform also allows for easy visualization of model performance using tools like Matplotlib and Seaborn, enabling thorough analysis and optimization. These features make Google Colab a practical and accessible choice for developing machine learning models in academic and professional settings. These features make Google Colab a practical and powerful tool for developing and refining tree-based models for house price prediction.

2.7 Website Development: Virtual Studio Code

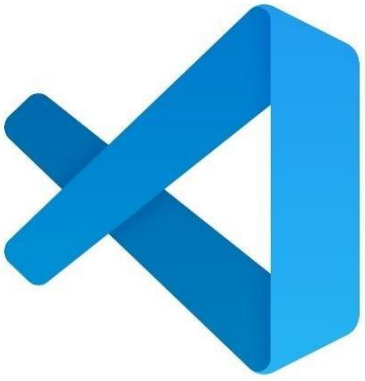


Figure 2.10 LOGO of Virtual Studio Code

Visual Studio Code (VS Code) is a powerful and versatile code editor that is widely used for web development. It offers a clean, user-friendly interface and supports multiple programming languages, including HTML, CSS, JavaScript, and Python, which are essential for building modern websites. VS Code's lightweight nature, combined with its vast library of extensions, makes it an ideal environment for developing, testing, and deploying web applications efficiently.

Features like IntelliSense (smart code completion), integrated terminal, live preview, and debugging tools simplify the development process, allowing for faster iteration and deployment (Carneiro, 2018).

As for the Final Year Project, VS Code is an excellent choice for developing the website that will serve as the front-end for the project. This website will display the model's predictions and provide users with an interactive way to input data and visualize results. With VS Code, you can easily integrate HTML, CSS, and JavaScript to design a responsive and professional user interface. Additionally, VS Code's compatibility with Flask or Django enables you to build a web server that connects directly to the machine learning model developed in Google Colab. This makes it possible to create a seamless connection between the predictive model and the web interface, allowing users to interact with the model in real-time.

VS Code is particularly perfect for the FYP because it allows for full-stack development in a single environment. You can manage the front-end (user interface) and the back-end (model integration) all within VS Code, ensuring that the website not only looks good but also functions smoothly with the predictive model. The ability to run live previews and quickly debug code ensures that the website accurately reflects the results of the machine learning model, providing a polished and professional presentation of the project. This integration makes VS Code an essential tool for turning the machine learning model into a practical, user-accessible application.

2.7.1 Backend Development

The backend serves as the core of the system, handling data processing, model inference, and communication with the frontend interface. This component is developed using the Flask framework, a lightweight and efficient Python-based web application framework well-suited for deploying machine learning models (Rawool, 2021). The backend exposes an API endpoint (/predict) that allows the frontend to send user input in the form of a JSON object. This input includes multiple features relevant to house price prediction, such as the number of bedrooms, living area square footage, number of bathrooms, and various property characteristics. The machine learning model used in this system is a Gradient Boosting model, specifically selected for its superior capability to handle structured real estate data and capture complex, non-linear relationships between features.

2.7.2 Frontend Development

The frontend serves as the user-facing interface that enables users to interact seamlessly with the house price prediction system. It is developed using standard web technologies such as HTML, CSS, and JavaScript, providing an intuitive and responsive platform for users to input property details. The frontend features a web form where users can enter key attributes of a property, including the number of bedrooms, square footage, number of bathrooms, and other relevant features that influence house prices. To ensure a smooth and user-friendly experience, the frontend also incorporates features such as real-time input validation, error handling, loading indicators during prediction processing, and responsive design to support various screen sizes and devices (Ibrahim, 2025).

2.8 Libraries (Google Colab and VS Code)

2.8.1 Google Colab

```
import pandas as pd
```

1. Pandas - For handling and manipulating datasets (e.g., importing CSV files, cleaning data).

```
import numpy as np
```

2. Numpy - For numerical operations and working with arrays.

```
from sklearn.model_selection import train_test_split
```

3. scikit-learn (sklearn) – Provides preprocessing tools (like scaling, encoding, and imputation).

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

4. scikit-learn (sklearn) – For basic Decision Tree, Random Forest, and Extra Trees models.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score
```

5. scikit-learn (metrics) – Provides performance metrics like MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE, and R^2 Score.

```
import matplotlib.pyplot as plt
import seaborn as sns
```

6. matplotlib – For basic plots (line plots, scatter plots, etc.).

   seaborn – For enhanced visualizations (heatmaps, pair plots, etc.)

```
from sklearn.feature_selection import SelectFromModel
```

7. sklearn.feature_selection – Tools for feature importance and selection.

```
import joblib
```

8. joblib – To save and load the trained model.

2.8.2 VS Code (Backend)

```
from flask import Flask, request, jsonify
import joblib
import numpy as np
```

```
return jsonify({'prediction': prediction[0]})
```

1. Flask – Lightweight framework to build web APIs and serve machine learning models.
2. joblib – Loads the trained model.
3. numpy – Reshapes input features for prediction.
4. jsonify – Converts the prediction into a JSON response for the frontend.

2.8.3 VS Code (Frontend)

1. HTML – Structure of the Web Page.

- Basic form elements to collect house features (square feet, bedrooms, bathrooms).

2. CSS (Optional, style.css) – Styling the Web Page.
    - Add CSS to enhance visual appeal and responsiveness.
3. JavaScript – Frontend Logic for API Requests.

2.9 Related Work

2.9.1 Tree-Based Models

2.9.1.1 House Price Prediction Based on Machine Learning Models

Ouyang's (2023) research on house price prediction using machine learning models closely aligns with the objectives of the final year project (FYP) on predicting house prices with tree-based models. The study compares Random Forest and Multiple Linear Regression (MLR), showing MLR's R^2 of 0.73, while Random Forest achieved 0.69, excelling in capturing non-linear interactions. This highlights the flexibility of tree-based models like Random Forest and Gradient Boosting in modeling complex data.

The study identifies key features influencing house prices, house size, number of bathrooms, floors, parking availability, and furnishing condition informing the feature selection process. Ouyang emphasizes data visualization and interpretability, reinforcing the importance of Decision Trees and Random Forests for transparent predictions.

The paper also notes limitations, such as the need for improved feature engineering and model tuning. These insights guide the approach to refining predictive accuracy through hyperparameter tuning and Gradient Boosting. Overall, Ouyang's work serves as a foundational reference, providing valuable methodologies and highlighting areas for further enhancement in predictive modeling.

2.9.1.2 A Comparative Study on House Price Prediction

Varma et al. (2024) evaluated the effectiveness of regression models, including Random Forest and Decision Tree Regression, for predicting home prices. Using K-fold Cross-Validation and RMSE, the study highlighted the strengths and weaknesses of each model. While Multiple Linear Regression (MLR) showed higher accuracy, the authors noted limitations, such as reliance on an open-source Kaggle dataset that may not reflect all real-world factors. External influences like market trends and economic conditions were also underexplored.

The study's insights on model evaluation, feature engineering, and dataset limitations will guide the project, which involves Decision Tree, Random Forest, and Gradient Boosting models. By addressing the identified gaps such as incorporating diverse datasets and accounting for external variables the research aims to improve the predictive accuracy and robustness of tree-based models for house price prediction.

2.9.1.3 Machine learning building price prediction with green building determinant

Mohd et al. (2024) explored machine learning models for predicting building prices, focusing on the integration of green building features. The study evaluated algorithms like Random Forest, Decision Tree, Ridge, Lasso, and Linear Regression using real estate datasets. Results showed that Random Forest performed best when green building attributes were included. However, the study noted that the influence of green features could diminish when combined with other factors.

This research is relevant to the project on house price prediction using tree-based models. The inclusion of additional features, such as green building determinants, offers insights into feature engineering and variable selection. The study also highlights challenges like limited green building data in Malaysia and the difficulty of predicting prices with incomplete datasets. These findings will help refine the project by emphasizing the need for comprehensive data and addressing potential dataset limitations to improve model performance.

2.9.1.4 Machine learning house price prediction in Petaling Jaya, Selangor, Malaysia

Mohd et al. (2024) investigated machine learning algorithms for predicting home prices in Petaling Jaya, Selangor, Malaysia, emphasizing the importance of feature selection and dataset quality. By evaluating various models, the study demonstrated how proper tuning and relevant features enhance prediction accuracy. The findings highlight the effectiveness of machine learning in real-world home price forecasting.

This research is relevant to the project on house price prediction using tree-based models. The focus on feature selection and dataset quality aligns with the goal of improving model accuracy. The study's insights into dataset limitations and parameter adjustments will guide the approach to refining data and tuning models. The authors also recommend further research to enhance predictive modeling in real estate, reinforcing the need to evaluate and adapt machine learning models for better performance.

2.9.1.5 House price forecasting using machine learning

Kuvalekar, Mahadik, Manchewar, and Jawale conducted a study on machine learning-based house price prediction in Mumbai, evaluating regression models like SVM, Random Forest, Linear Regression, Multiple Linear Regression, Decision Tree, and KNN. The Decision Tree Regressor achieved the highest accuracy at 89%, especially when integrated with the Flask web framework. The authors recommended adding features like crime rate and air quality to further enhance model performance.

This study is relevant to the project on house price prediction using tree-based models. The success of the Decision Tree Regressor supports the focus on Decision Tree, Random Forest, and Gradient Boosting algorithms. The suggestion to include additional features aligns with the plans to enhance prediction accuracy by expanding relevant variables.

Although the study showed high accuracy, it did not address potential challenges or limitations. This highlights the need for the project to consider factors like data accessibility, generalization across regions, and external influences to ensure model robustness and reliability in real-world scenarios.

2.9.1.6 House Price Prediction Based on Different Models of Machine Learning

Chuhan et al. (2024) conducted an in-depth study on machine learning models for house price prediction, evaluating Random Forest, XGBoost, SVM, and linear regression. The study found that XGBoost and Random Forest outperformed the other models, with XGBoost achieving the highest accuracy and lowest error rates. This highlights the effectiveness of tree-based models in predicting home prices.

The study is highly relevant to the project, which focuses on using Random Forest and Gradient Boosting for house price prediction. The findings reinforce the advantages of

tree-based models, aligning with the goal of achieving accurate predictions through advanced algorithms.

The authors emphasized the importance of model complexity and interpretability, which will inform the approach to model selection and tuning. They also noted limitations such as the use of a one-year-old dataset and data from Kaggle, underscoring the need for up-to-date, high-quality data. These insights will help me refine the dataset and improve model performance to ensure reliable and representative predictions.

2.9.1.7 House Price Prediction using Random Forest Machine Learning Technique.

Adetunji et al. explored the use of the Random Forest algorithm to predict house prices using the Boston housing dataset from the UCI Machine Learning Repository. The study highlights the importance of data pre-processing, including normalizing numerical data, encoding categorical values, and selecting key features through heatmaps. The Random Forest model demonstrated strong predictive performance, evaluated using MAE, R^2, and RMSE, with results visualized through scatter plots. K-fold cross-validation contributed to improving the model's accuracy by balancing bias and variance.

The study acknowledges limitations, such as focusing on a single dataset and model, which may restrict the generalizability of the findings. It also lacks detailed analysis of how outliers or missing data could affect performance. Despite these limitations, the research provides valuable insights into machine learning for house price prediction and serves as a foundation for further studies, aligning with and supporting the objectives of the final year project (FYP).

2.9.1.8 House price prediction using machine learning

The research report on Random Forest and Gradient Boosting for house price prediction supports the final year project, which also focuses on tree-based models like Decision Tree, Random Forest, and Gradient Boosting. The report highlights Random Forest's ability to handle large datasets with multiple features and reduce overfitting, aligning with the approach to use it for modeling both linear and non-linear relationships. It also emphasizes Gradient Boosting's power in capturing complex data patterns, which is crucial for accurately predicting house prices.

The report's insights on feature selection using mean square error (MSE) and the internal workings of Random Forest will guide the feature engineering and model fine-tuning. However, the report also acknowledges challenges such as capturing complex relationships and model interpretability, which I plan to address by expanding the dataset and exploring SHAP values for better transparency. The report also stresses the importance of a robust deployment strategy, which I will incorporate into the project. Overall, the research strengthens the understanding of these algorithms and will inform the development of accurate and interpretable house price prediction models.

2.9.1.9 Machine Learning Models for Housing Price Prediction

The paper "Machine Learning Models for Housing Price Prediction" by Quang Truong et al. aligns with the project, which focuses on predicting house prices using tree-based models like Decision Tree, Random Forest, and Gradient Boosting. The paper highlights Random Forest’s ensemble approach, which reduces variance and overfitting, a technique I will use in the project. However, it also notes the risk of overfitting and high

computational costs, reminding me to apply cross-validation and hyperparameter tuning. Truong et al. emphasize Extreme Gradient Boosting's scalability and robustness, key for handling large datasets and capturing non-linear relationships, which aligns with the use of Gradient Boosting. The paper also introduces RMSLE as a performance metric, which I will adopt to evaluate model accuracy. Limitations like computational complexity and overfitting in Random Forest are addressed in the project through feature selection and model stacking. Overall, the paper strengthens the theoretical and practical approach for the project, focusing on improving prediction accuracy with multiple tree-based algorithms.

2.9.1.10 Analysis of Real Estate Predictions Based on Different Models

The report "Analysis of Real Estate Predictions Based on Different Models" by Sommervoll et al. aligns with the project on predicting house prices using tree-based models like Decision Tree, Random Forest, and Gradient Boosting. The report emphasizes the effectiveness of ensemble learning in capturing complex relationships within housing data, reinforcing the choice of these algorithms. It also highlights Random Forest's susceptibility to noise and lack of interpretability, reminding me to prioritize feature selection and preprocessing. The report's insights into Gradient Boosting's iterative error correction and regularization will guide the approach to enhancing model performance. Although the report explores alternative models like genetic algorithms and RNNs, the project will focus on tree-based methods, with potential future exploration of hybrid models. The paper's discussion of model limitations, including computational complexity and sensitivity to noisy data, will inform the strategy for refining models through parameter tuning and cross-validation.

Ultimately, the report strengthens the plan to leverage these tree-based models for accurate and reliable house price prediction.

2.9.1.11 A Comparative Study of Regression Models for Housing Price Prediction

The report "A Comparative Study of Regression Models for Housing Price Prediction" compares the models: Decision Tree Regression, Random Forest Regression, Ridge Regression, and Extreme Gradient Boosting Tree Regression. It highlights that Decision Tree and Random Forest Regression models perform best based on metrics like MAE, MSE, RMSE, and R^2.

Decision Trees effectively handle non-linear relationships but are prone to overfitting, while Random Forests improve generalization by combining multiple trees. Extreme Gradient Boosting is noted for its iterative enhancement of performance. The study also shows that Ridge Regression struggles with many features or outliers, and the choice of input features significantly affects model performance. The report emphasizes the importance of dataset characteristics and noise reduction methods in model selection, offering valuable insights for housing price prediction and real estate data analysis.

2.9.1.12 A Comparative Study of Random Forest Regression for Predicting House Prices

The paper "A Comparative Study of Random Forest Regression for Predicting House Prices" by Mohan Mao focuses on the effectiveness of Random Forest and Gradient Boosting regression techniques for house price prediction. Mao emphasizes Random

Forest's ability to aggregate predictions from multiple decision trees, improving accuracy and reducing overfitting, making it ideal for high-dimensional datasets. Gradient Boosting improves weak models sequentially to handle complex data relationships, which is crucial for the project.

Mao's use of Optuna for hyperparameter tuning offers valuable insights for optimizing model performance. He also discusses limitations like overfitting and computational complexity, which are key considerations for the project. Overall, the study supports the use of tree-based models in the house price prediction work and provides strategies for addressing their limitations.

2.9.1.13 Comparison of tree-based machine learning algorithms in price prediction of residential real estate

The research report on machine learning models for house price prediction highlights the use of decision trees, random forests, and gradient boosting, which are central to the final year project. The study explains how decision trees break down complex data, random forests aggregate predictions to improve accuracy, and gradient boosting sequentially builds models to correct errors, all of which align with the project goals.

The study shows that XGBoost and Random Forest performed well in predicting house prices, though limited sample size impacted model performance. The report also discusses the flaws in assuming the housing market is economically efficient and the influence of macroeconomic factors like employment rates, which I plan to incorporate into the project.

These insights will guide the approach to tree-based models, helping refine the

predictions while addressing potential challenges in housing market complexities.

2.9.1.14 Predicting House Price: A Comparative Study with Regression Methods

The report "Predicting House Price: A Comparative Study with Regression Methods" offers a comprehensive analysis of regression models used for house price prediction, directly aligning with the methodologies of the final year project. The study evaluates decision tree regression, random forest regression, and gradient boosting regression, highlighting their strengths and limitations in the context of house price prediction. Decision tree regression, with its tree-like structure, is foundational in the project, offering interpretability and understanding of feature relationships. The report also emphasizes random forest regression as an ensemble method that improves prediction accuracy, which is central to the project to reduce overfitting by aggregating multiple decision trees.

Gradient boosting regression, another key focus of the report, is an iterative model that builds trees sequentially, correcting errors from the previous trees, which will be beneficial for improving model performance in the project. The report provides valuable insights into the trade-offs between accuracy, complexity, and interpretability of each model, helping me address issues like overfitting, dataset requirements, and model complexity in the own project. Additionally, the practical implications of the models for real estate professionals in the report will guide the real-world applicability of the predictions, ensuring the model is both accurate and useful.

In conclusion, the report's findings will significantly contribute to the development of the project. By integrating decision tree, random forest, and gradient boosting techniques and

considering the challenges and trade-offs discussed, I can refine the approach to house price prediction and develop a model that balances accuracy, complexity, and interpretability.

2.9.1.15 A Literature Survey on Housing Price Prediction

The report "A Literature Survey on Housing Price Prediction" provides valuable insights into the application of machine learning models, including decision trees, random forest, and gradient boosting, which are central to the final year project. The study emphasizes the effectiveness of these models, particularly gradient boosting, which outperforms others by reducing residual errors and improving prediction accuracy. This aligns with the plan to incorporate gradient boosting to optimize the predictive model by utilizing its iterative learning process.

The report also highlights random forest as an ensemble method that aggregates predictions from multiple decision trees to reduce overfitting and enhance accuracy. The authors report high R-squared values for random forest, supporting its use in the project to achieve reliable predictions. Additionally, the study addresses the need for a larger dataset and more features, such as swimming pools and parking spaces, which will guide the approach to feature selection and data preprocessing in the real-world context.

In conclusion, the report's insights into model performance and dataset challenges will inform the project's approach to feature selection and model evaluation. By addressing these limitations, I aim to refine the predictive accuracy and reliability of the house price prediction model, ultimately contributing to the success of the final year project.

2.9.2 Ensemble Boosting Models

2.9.2.1 House Price Prediction Using Gradient Boost Regression Model

The paper "House Price Prediction Using Gradient Boost Regression Model" by Kumar et al. (2024) examines the use of the Gradient Boosting regression model for house price prediction, which aligns with the project on tree-based models like Gradient Boosting, Random Forest, and Decision Trees. The study highlights the importance of data preprocessing, feature identification, and the iterative nature of Gradient Boosting, which improves model accuracy by combining weak predictors. It emphasizes the significance of location and neighbourhood attributes, which will guide the feature engineering process.

The paper also discusses limitations, such as capturing nuanced neighbourhood attributes and unobservable factors, reminding me that data quality is critical. These insights will help refine the models by ensuring robust data collection and feature selection, ultimately enhancing prediction accuracy.

2.9.2.2 Machine Learning Based House Price Prediction Using Modified Extreme Boosting

The paper "Machine Learning Based House Price Prediction Using Modified Extreme Boosting" by Ragapriya et al. (2024) explores the use of Modified Extreme Gradient Boosting (XGBoost) for house price prediction, which aligns with the project focusing on tree-based models like Decision Tree, Random Forest, and Gradient Boosting. The study highlights the superiority of Modified XGBoost over traditional regression models, reinforcing the decision to include gradient boosting in the project. It emphasizes the importance of feature engineering, particularly location, area type, and structural factors,

which will guide the own feature selection process.

The paper also discusses the importance of high-correlation features and the need for diverse local data to improve model accuracy. This insight is crucial for addressing the potential gaps and inconsistencies in the real-world dataset. Additionally, the authors acknowledge external factors like housing shortages, which will inform the interpretation of the model's outputs. Overall, the paper strengthens the understanding of gradient boosting techniques and provides valuable lessons on data enhancement, which will aid in developing robust house price prediction models.

2.9.2.3 Research on the House Price Forecast Based on machine learning algorithm

The report "Research on the House Price Forecast Based on Machine Learning Algorithm" evaluates various machine learning models, including decision tree, random forest, AdaBoost, GBDT, and Extreme Gradient Boosting. These techniques directly align with the methodologies in the project, particularly decision trees and random forests, which improve prediction accuracy and reduce overfitting. The study also highlights gradient boosting's ability to enhance model performance through sequential error correction, a feature I plan to utilize for better house price predictions.

The report stresses the importance of feature selection, identifying variables like 'OverallQual' and 'GrLivArea' as key factors. It also emphasizes the role of hyperparameter optimization, particularly Bayesian optimization, which I intend to incorporate to fine-tune the models. The study acknowledges limitations, such as AdaBoost's tuning challenges, and suggests future research in incorporating temporal factors affecting house prices, which could be explored in future iterations of the project.

In conclusion, the methodologies and insights provided will guide the project in applying tree-based models to predict house prices more effectively while addressing potential limitations and optimizing performance.

2.9.3 Probability-Based Models

2.9.3.1 House Price Prediction using a Machine Learning Model A Survey of Literature

Zulkifley et al. conducted a study on house price prediction using machine learning models, evaluating support vector regression (SVR), artificial neural networks (ANN), and Extreme Gradient Boosting (XGBoost). The authors emphasized the critical role of locational, structural, and neighbourhood factors, along with economic conditions, in determining house prices. The study found that SVR, ANN, and XGBoost were particularly effective in accurately predicting house prices.

This research is relevant to the project, which focuses on tree-based models like Decision Tree, Random Forest, and Gradient Boosting for house price prediction. The emphasis on selecting important features aligns with the goal of optimizing relevant factors to enhance prediction accuracy. Additionally, the insights into model selection and evaluation will guide the efforts to choose and refine the most effective techniques.

The study also highlights limitations, such as data bias and limited generalizability due to specific datasets and models. This will prompt me to carefully consider dataset quality and ensure model adaptability to various scenarios. The authors' call for further research to improve robustness and applicability encourages to explore diverse datasets and refine the approach to address similar challenges in the project.

2.9.3.2 A Hybrid Regression Technique for House Prices Prediction

The research paper "A Hybrid Regression Technique for House Prices Prediction" focuses on using hybrid regression models, specifically Gradient Boosting and Random Forest, for predicting house prices. These models performed well in a Kaggle competition, highlighting their effectiveness in handling complex datasets. The study emphasizes the importance of feature engineering and selecting the right regression algorithms.

This aligns with the project, which also uses Decision Tree, Random Forest, and Gradient Boosting to predict house prices. The paper's insights on feature engineering and model selection will guide the approach to preprocessing data and selecting features. It also addresses challenges like data cleaning and feature selection, which I plan to tackle in the project to improve model accuracy and robustness.

2.9.3.3 Empirical Analysis of Regression Techniques by House Price and Salary Prediction

The research paper "Empirical Analysis of Regression Techniques by House Price and Salary Prediction" compares Simple Linear Regression (SLR) and Multiple Linear Regression (MLR) models for predicting house prices and salaries. It highlights the importance of feature selection and evaluates models using metrics like R-squared, RMSE, MAE, and MSE.

Although the project focuses on tree-based models (Decision Tree, Random Forest, Gradient Boosting) for house price prediction, the paper's emphasis on feature importance is relevant. The paper discusses how factors like land area significantly impact house prices, which I plan to incorporate in the own feature engineering. It also acknowledges limitations, such as challenges in accounting for all influencing factors, reminding me to

focus on data quality and feature selection. While the approach will be non-linear, the insights on feature selection, model evaluation, and dataset limitations will help refine the models and improve prediction accuracy.

2.9.4 Neural Network Models

2.9.4.1 Dream House Price Predict

The paper "Dream House Price Predictor" by Singh et al. explores machine learning models, particularly support vector regression (SVR) and neural networks, for predicting residential property prices based on factors like location, size, and amenities. The study highlights SVR's effectiveness in forecasting property values using historical real estate data, demonstrating the role of machine learning in aiding real estate decisions and improving price accuracy.

This research aligns with the project, which focuses on tree-based models (Decision Tree, Random Forest, and Gradient Boosting) for house price prediction. Although the paper emphasizes SVR and neural networks, its insights on feature selection (e.g., location and size) are valuable for enhancing the accuracy of the models.

The study also addresses limitations, such as data bias and the need for diverse datasets, which are critical considerations for the project. I will focus on using comprehensive datasets to improve model generalizability, aligning with the paper's recommendation for further research to enhance predictive model robustness in real estate.

2.9.4.2 Influence factors and regression model of urban housing prices based on internet open access data

Wu et al. (2024) conducted a study on urban housing prices in Wuhan, China, using

diverse data sources such as internet data, LBS, POI, and urban planning data. They applied artificial neural networks (ANN) and non-linear regression models to analyse the spatial factors influencing housing prices, providing valuable insights into real estate price prediction.

This study is relevant to the project, which uses tree-based models (Decision Tree, Random Forest, and Gradient Boosting) for house price prediction. Wu et al.'s use of multi-source data highlights the importance of incorporating varied datasets to improve model accuracy, an approach I plan to adopt by sourcing comprehensive real estate data. The study also notes limitations, such as data acquisition challenges and limited generalizability to Wuhan. These insights will guide the project by encourages dataset expansion and testing models across different regions to enhance reliability. By addressing these challenges, I aim to develop a scalable, robust house price prediction model.

2.9.4.3 A Novel Hybrid House Prediction Model

The report "A Novel Hybrid House Price Prediction Model" aligns with the final year project, which focuses on predicting house prices using tree-based models like Decision Tree, Random Forest, and Extreme Gradient Boosting. The report introduces a hybrid approach that combines clustering techniques with regression models to handle the variability in housing data, which could inspire feature engineering strategies for the project.

The report highlights the strengths of Decision Trees, Random Forest, and Gradient Boosting in capturing non-linear relationships, reducing overfitting, and refining predictions iteratively, reinforcing the selection of these algorithms. Performance metrics like RMSE, MAPE, and adjusted R-squared will guide the evaluation of the models. The

report also addresses challenges such as the dynamic nature of housing markets, which will inform the approach to data cleaning, feature selection, and model tuning. Overall, the report strengthens the approach to developing robust and accurate house price prediction models by exploring the complementary strengths of different methods.

2.9.5 Support Vector Machine (SVM) Models

2.9.5.1 A Case Study Using Machine Learning Techniques for Prediction of House Prices in WP, Malaysia

The case study investigates the application of various machine learning algorithms including Decision Tree Regression, Random Forest Regression, and Gradient Boosting for predicting house prices within the Federal Territory (Wilayah Persekutuan, WP), Malaysia. The study demonstrates that these tree-based models are highly effective for capturing the complex relationships inherent in real estate data. Among the models tested, Random Forest Regression achieved the highest R^2 score of 0.9953, indicating that it could explain over 99% of the variability in house prices. This was closely followed by Decision Tree Regression, with an R^2 score of 0.9933, further validating the strength of tree-based approaches in this domain. These findings strongly support the use of tree-based models in the current project, reinforcing the decision to adopt algorithms such as Decision Trees, Random Forest, and Gradient Boosting for accurate house price prediction.

One of the key advantages highlighted by the study is the capability of Random Forest to minimize prediction errors by aggregating predictions from multiple decision trees. This ensemble technique allows Random Forest to reduce variance and improve generalization, particularly when handling high-dimensional housing datasets that contain numerous features such as property size, location, number of bedrooms, amenities, and market conditions. This

capability makes Random Forest especially suitable for complex real estate markets where multiple variables interact in non-linear ways to influence property values.

In addition to evaluating model performance, the case study emphasizes the critical role of dataset diversity in improving model robustness and generalizability. By incorporating data from multiple regions and diverse housing markets, models can learn a wider range of patterns and relationships, enabling them to make more accurate predictions even when presented with new or previously unseen data. This approach aligns closely with the project's strategy to build a comprehensive dataset that captures a variety of property types and locations, ensuring that the resulting models are not overly biased toward any single region and can generalize effectively to broader market conditions.

Furthermore, the report provides an in-depth discussion on the strengths and limitations of each model. While both Random Forest and Decision Tree Regression demonstrate strong predictive performance, they are not without challenges. The study identifies potential risks such as overfitting, where the model becomes too finely tuned to the training data and fails to perform well on unseen data. This is particularly a concern for Decision Trees, which can grow excessively complex without regularization.

The insights derived from this case study offer valuable guidance for the ongoing project. By adopting tree-based models and incorporating strategies to address potential limitations, the project aims to develop highly accurate, robust, and reliable predictive models for house price estimation. These models will not only serve academic purposes but can also be applied in practical real-world scenarios to assist homeowners, investors, and real estate professionals in making informed property-related decisions.

2.10 Comparison between related work

No	Previous Papers	Year	Attributes/Factors	Model	Limitations / Gaps
1	Ouyang, X. (2024). House Price Prediction Based on Machine Learning Models. *Highlights in Science, Engineering and Technology*, *85*, 870–878. https://doi.org/10.54097/FTYF9665	2024	Location, size, number of bedrooms, market trends, proximity to amenities	Linear Regression and Random Forest Regression	Model performance highly depends on dataset quality; no exploration of other ML methods for comparison.
2	Akash Dagar and Shreya Kapoor. (2020). A Comparative Study on House Price Prediction. *International Jthenal for Modern Trends in Science and Technology*, *6*(12), 103–107. https://doi.org/10.46501/ijmtst0612 20	2020	Area, location, age of property, number of rooms	Multivariable Linear Regression, Decision Tree Regression, Random Forest Regression	Lack of scalability for larger datasets; limited focus on hyperparameter optimization for tree-based models.
3	Mohd, T., Jamil, S., & Masrom, S. (2020). Machine learning building price prediction with green building determinant. *IAES International Jthenal of Artificial Intelligence (IJ-AI)*, *9*(3), 379–386. https://doi.org/10.11591/ijai.v9.i3.pp379-386	2020	Green building factors, property type, location, environmental impacts	Linear Regression, Decision Tree, Random Forest, Ridge and Lasso algorithms	Limited generalizability to non-green buildings; small dataset size restricts model evaluation.

4	Mohd, T., Masrom, S., & Johari, N. (2019). Machine learning housing price prediction in petaling jaya, Selangor, Malaysia. *International Jthenal of Recent Technology and Engineering*, *8*(2 Special Issue 11), 542–546. https://doi.org/10.35940/ijrte.B1084.0982S1119	2019	Property type, land area, age, location	Linear Regression, Decision Tree, Random Forest, Ridge and Lasso algorithms	Study focused only on a specific geographical area (Petaling Jaya, Selangor), limiting wider applicability.
5	Kuvalekar, A., Mahadik, S., Manchewar, S., & Jawale, S. (n.d.). *HOUSE PRICE FORECASTING USING MACHINE LEARNING*. https://ssrn.com/abstract=3565512	2024	Number of rooms, size, locality, historical pricing trends	Decision Tree Regression	Dataset preprocessing not extensively detailed; limited model evaluation metrics.
6	Chuhan, N. (2024). House price prediction based on different models of machine learning. *Applied and Computational Engineering*, *49*(1), 47–57. https://doi.org/10.54254/2755-2721/49/20241058	2024	Size, number of bedrooms and bathrooms, proximity to public transport	Linear Regression, Support Vector Machine (SVM), Random Forest regression, Extreme Gradient Boosting	No detailed discussion on feature importance or interpretability of results.
7	Adetunji, A. B., Akande, O. N., Ajala, F. A., Oyewo, O., Akande, Y. F., & Oluwadara, G. (2022). House Price Prediction using	2022	Distance to city centre, property type, land size, social factors	Random Forest Regression	Limited cross-validation techniques; no use of advanced ensemble methods for comparison.

	Random Forest Machine Learning Technique. *Procedia Computer Science*, *199*, 806–813. https://doi.org/10.1016/j.procs.2022.01.100				
8	Zulkifley, N. H., Rahman, S. A., Ubaidullah, N. H., & Ibrahim, I. (2020). House price prediction using a machine learning model: A survey of literature. *International Jthenal of Modern Education and Computer Science*, *12*(6), 46–54. https://doi.org/10.5815/ijmecs.2020.06.04	2020	Building characteristics, location, amenities, market demand	Hedonic Price Model, Multiple Linear Regression, SVM, Artificial Neural network (ANN), Gradient Boost	Survey-based study, lacking practical implementation details or real-world validation.
9	Singh, A., Chand, K., Singh, S., & Soni, K. (2023). Dream House Price Predictor. *International Jthenal for Research in Applied Science and Engineering Technology*, *11*(4), 1441–1446. https://doi.org/10.22214/ijraset.2023.50307	2023	Lot size, year built, market trends, property features	SVM, ANN	Focused mainly on regression analysis; feature selection processes not extensively discussed.
10	Wu, H., Jiao, H., Yu, Y., Li, Z., Peng, Z., Liu, L., & Zeng, Z. (2018). Influence factors and regression model of urban housing prices based on internet open	2018	Proximity to city centre, education facilities, public transport, environmental factors	ANN	Internet-based data may lack quality assurance; geographical focus restricts applicability elsewhere.

	access data. *Sustainability (Switzerland)*, *10*(5). https://doi.org/10.3390/su10051676				
11	Lu, S., Li, Z., Qin, Z., Yang, X., & Goh, R. S. M. (2017). A hybrid regression technique for house prices prediction. *IEEE International Conference on Industrial Engineering and Engineering Management*, *2017-December*, 319–323. https://doi.org/10.1109/IEEM.2017.8289904	2017	Lot area, neighbourhood, quality of materials, square footage of living area	Lasso, Gradient Boosting, Ridge, and Random Forest	Hybrid regression approach underutilized tree-based models for performance comparison
12	Bansal, U., Narang, A., Sachdeva, A., Kashyap, I., & Panda, S. P. (2021). Empirical analysis of regression techniques by house price and salary prediction. *IOP Conference Series: Materials Science and Engineering*, *1022*(1). https://doi.org/10.1088/1757-899X/1022/1/012110	2021	Room count, age, location, economic indicators	Simple Linear Regression (SLR), Multiple Linear Regression (MLR)	Limited diversity in dataset; basic regression models without advanced tree-based ensemble methods.
13	Kumar, Bv., & Professor, A. (2020). HOUSE PRICE PREDICTION USING GRADIENT BOOST REGRESSION MODEL.	2020	Neighbourhood, lot size, historical pricing data, economic indicators	Gradient Boosting	Performance limited to Gradient Boost regression; no comparative study with other tree-based models.

	International Jthenal of Research and Analytical Reviews, *7*(1). www.ijrar.org				
14	N. Ragapriya, Kumar, T. A., R. Parthiban, P. Divya, S. Jayalakshm, & Raman, D. R. (2023). Machine Learning Based House Price Prediction Using Modified Extreme Boosting. *Asian Jthenal of Applied Science and Technology*, *07*(01), 41–54. https://doi.org/10.38177/ajast.2023.7105	2023	Modified boosting factors, property features, market conditions	Extreme Gradient Boosting	Computationally intensive; lacks discussion on real-world applicability and scalability of the proposed approach.
15	Kalidass, J., Dharshalini, T., Nivetha, R., & Subasri, A. P. (2024). HOUSE PRICE PREDICTION USING MACHINE LEARNING. *International Research Jthenal of Engineering and Technology*. www.irjet.net	2024	Property size, neighbourhood quality, proximity to schools and work centers	Random Forest, Gradient Boosting	Dataset preprocessing and feature engineering not emphasized; lacks insights into computational cost considerations.
16	Quang, T., Minh, N., Hy, D., & Bo, M. (2020). Housing Price Prediction via Improved Machine Learning Techniques. *Procedia Computer Science*, *174*, 433–442. https://doi.org/10.1016/j.procs.2020.06.111	2020	Proximity to business hubs, crime rates, public infrastructure, social factors	Random Forest, Extreme Gradient Boosting	Improved techniques but lack cross-region adaptability; no detailed explainability analysis.

17	Özöğür Akyüz, S., Eygi Erdogan, B., Yıldız, Ö., & Karadayı Ataş, P. (2023). A Novel Hybrid House Price Prediction Model. *Computational Economics*, *62*(3), 1215–1232. https://doi.org/10.1007/s10614-022-10298-8	2023	Location, size, number of rooms, age of the house, economic indicators	Hybrid Model (Combination of Gradient Boosting and Neural Networks)	External economic fluctuations (like recessions or policy changes) are not dynamically factored, impacting model stability
18	Li, Y. (2023). Analysis of Real Estate Predictions Based on Different Models. In *Highlights in Science, Engineering and Technology AMCCE* (Vol. 2023).	2023	Square footage, proximity to amenities, market trends, year of construction	Decision Tree, Extreme Gradient Boosting, Random Forest	The absence of temporal data integration leads to inaccurate forecasting when long-term trends shift.
19	Li, Z. (2024). A Comparative Study of Regression Models for Housing Price Prediction. In *Transactions on Computer Science and Intelligent Systems Research* (Vol. 5).	2024	Lot size, number of bedrooms, local economic growth, interest rates	Linear Regression, Random Forest, Extreme Gradient Boosting	The study focuses on short-term trends, neglecting long-term influences like inflation or housing development cycles.
20	Mao, M. (2024). A Comparative Study of Random Forest Regression for Predicting House Prices Using. In *Highlights in Science, Engineering and Technology CSIC* (Vol. 2023).	2024	Property type, neighbourhood characteristics, transport access, historical prices	Random Forest Regression	The model uses static factors and lacks integration of dynamic factors like policy changes or new infrastructure projects.

21	YAVUZ ÖZALP, A., & AKINCI, H. (2023). Comparison of tree-based machine learning algorithms in price prediction of residential real estate. *Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi*. https://doi.org/10.17714/gumusfenbil.1363531	2023	Proximity to public transport, land area, crime rates, infrastructure	Decision Tree, Random Forest, Extra Trees, Gradient Boosting	Overlapping features (e.g., public transport and infrastructure) may introduce multicollinearity, affecting prediction reliability.
22	Bau, Y.-T., & Hisham, S. M. S. B. (2022). A Case Study Using Machine Learning Techniques for Prediction of House Prices in WP, Malaysia. In *Proceedings of the International Conference on Computer, Information Technology and Intelligent Computing (CITIC 2022)* (pp. 79–91). Atlantis Press International BV. https://doi.org/10.2991/978-94-6463-094-7_7	2022	Property age, floor area, amenities, income level	Support Vector Machine, Random Forest, Extreme Gradient Boosting	A relatively small sample size reduces the statistical significance and robustness of the model.
23	Weng, W. (2022). Research on the House Price Forecast Based on machine learning algorithm. In *BCP Business & Management AFTEM* (Vol. 2022).	2022	Historical sales data, economic factors, neighbourhood prices	Extreme Gradient Boosting, Lasso Regression	The model cannot adapt quickly to abrupt policy or economic changes, causing lag in price prediction

24	Sharma, M., Patil, P., Sharma, D., Joge, I., Burle, R., & Puri, C. (n.d.). *Predicting House Price: A Comparative Study with Regression Methods*.		Lot size, construction quality, market demand	Linear Regression, Decision Tree, Random Forest	Certain features, such as construction quality and lot size, may be redundant, leading to less efficient models.
25	Suresh Yalgudkar, S., & v. Dharwadkar, N. (2022). A Literature Survey on Housing Price Prediction. *Jthenal of Computer Science & Computational Mathematics*, *12*(3), 41–45. https://doi.org/10.20967/jcscm.2022.03.002	2022	Square footage, local development, housing policies	Random Forest, Decision Tree	The model lacks integration of real-time economic or market indicators.

Table 2.1 Summary of previous related work

2.10.1 Conclusion

This chapter provided a thorough rundown of all the various aspects of using tree-based machine learning models for predicting the cost of housing. The mathematical basis showed the value of machine learning in practical applications by emphasizing its capacity to evaluate complex datasets and extract important insights. The capabilities of tree-based models specifically, decision trees, random forests, and gradient boosting techniques like Extreme Gradient Boosting in handling big, complicated datasets and capturing non-linear connections were discussed in details.

Reviewing related works showed that a number of research have used tree-based models to predict the cost of housing, highlighting the importance of important factors including location, property size, and economic indicators. Furthermore, cutting-edge strategies for enhancing model performance through sophisticated ensemble approaches and feature engineering were investigated, highlighting the significance of prediction accuracy and clarity.

Based on the research, gradient boosting, have outperformed traditional methods in predicting home prices. This helps to achieve the project's goals, which include creating a XGBoost model to predict, evaluating the model's performance using performance indicators, and analyzing its performance using key characteristics. This project attempts to build on previous studies by utilizing these models in order to produce a more reliable prediction tool. In order to ensure that the goals mentioned in Chapter 1 are fulfilled through complete model development and evaluation, Chapter 3 will go into the methodology for putting the suggested model into practice.

CHAPTER 3 : METHODOLOGY

3.1 Introduction

The methodology begins with the process flow of the project, detailing each step from data collection and preprocessing to model development and evaluation. A comprehensive breakdown of the techniques used to train and fine-tune the predictive model is provided, ensuring optimal performance and accuracy. This section also elaborates on the integration of the trained model into a functional web-based platform that allows users to input house attributes and receive price predictions, making the system accessible and interactive.

Key components of the methodology include the creation of system flow diagrams, use case diagrams, and architectural designs to visualize the interactions between the model, the website, and the user. Additionally, the chapter highlights the software and tools employed throughout the project, including Google Colab for model training and experimentation, and Visual Studio Code (VS Code) for web development and system integration. Other essential tools and libraries, such as Flask for backend development and Pandas for data manipulation, are also discussed in detail.

3.2 Project Flow

The project flowchart outlines a systematic approach to completing a research and development project, beginning with the initiation phase and concluding with the final evaluation and recommendation.

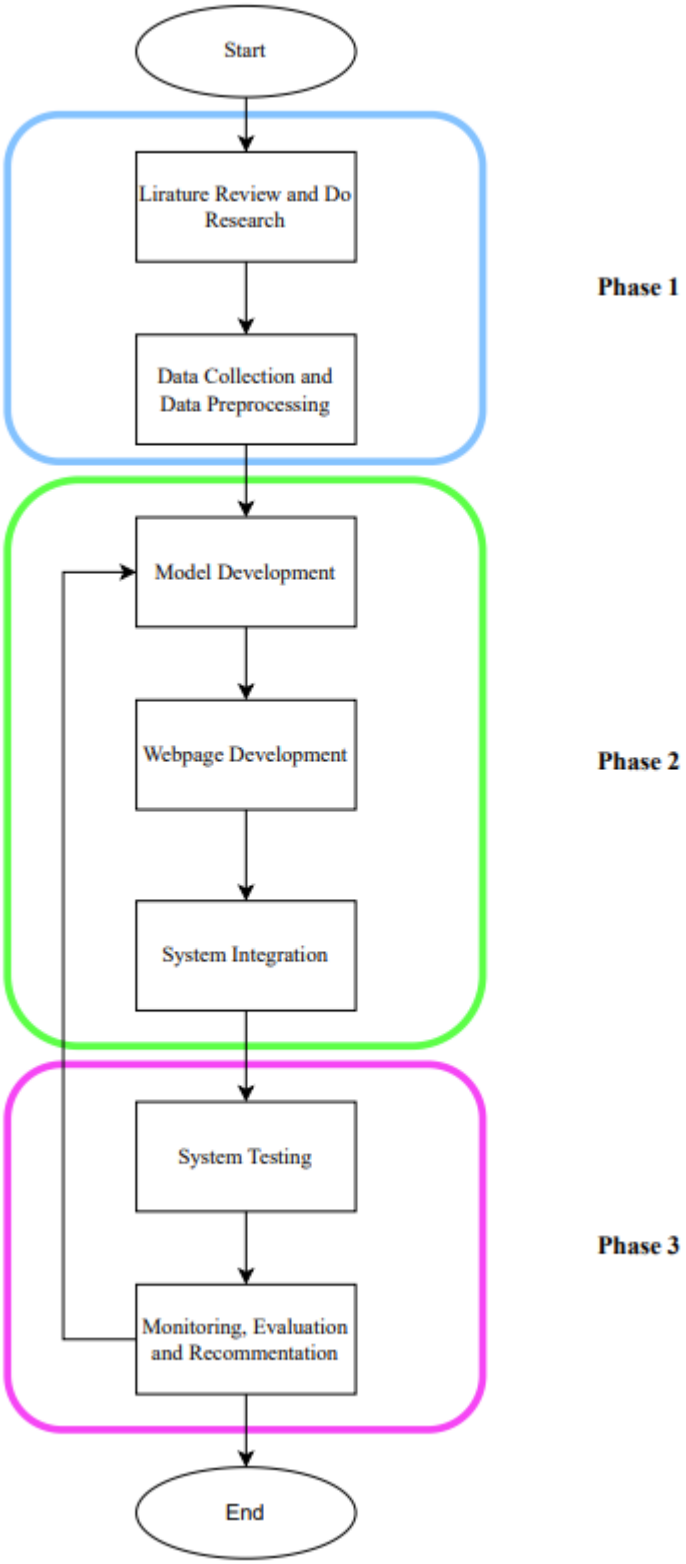


Figure 3.1 Flowchart of Project Flow

The process begins by defining the research problem, aim, and objectives to establish a clear direction and structured focus. A thorough literature review follows, identifying gaps, refining methodologies, and grounding the project in established knowledge to

minimize redundancy and foster innovation. The development phase transitions from planning to execution, involving model creation, website development, and system integration to create a unified, functional system. This is followed by testing and evaluation, where system performance is validated, errors are resolved, and feedback is used to recommend improvements. The project concludes with the completion of key milestones, preparing the system for deployment or presentation and guiding future enhancements, ensuring alignment with the initial objectives throughout the process.

3.2.1 Project Flow Phase 1

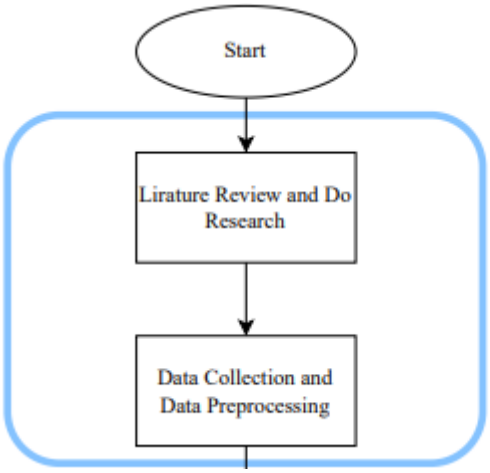


Figure 3.2 Flowchart of Phase 1

The flowchart depicts Phase 1 of the house price prediction system development process. The phase begins with the "Start" point, followed by conducting a comprehensive literature review and background research to understand existing methods, algorithms, and challenges related to house price prediction. This initial research phase helps to build the foundational knowledge required for the project. After completing the literature review, the next step involves "Data Collection and Data Preprocessing," where relevant housing datasets are gathered, cleaned, and prepared for subsequent model development. This preprocessing step includes handling missing values, feature engineering, and transforming the data into a format suitable for machine learning model training.

3.2.2 Project Flow Phase 2

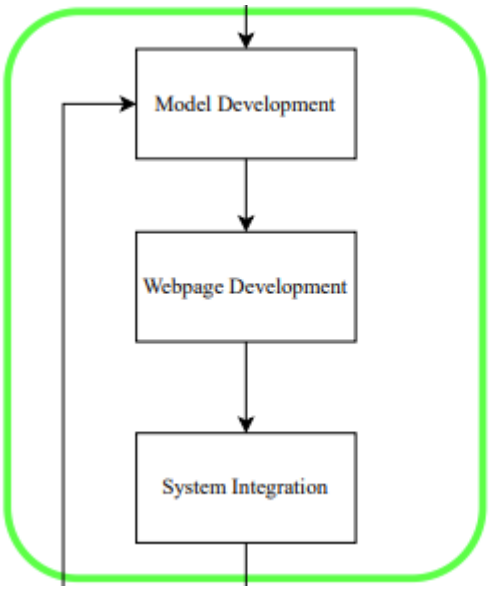


Figure 3.3 Flowchart of Phase 2

Phase 2 of the project marks the transition from planning and research to actual development and implementation. This phase begins with Model Development, where the core functionality or algorithm of the project is created. For machine learning or AI-driven projects, this step involves designing, training, and fine-tuning models based on the data and objectives outlined in the previous phase. The accuracy and performance of the model are critical at this stage, as they form the backbone of the final system. Iterative improvements and testing may occur to ensure the model meets desired benchmarks.

The provided diagram outlines the machine learning model training pipeline, which consists of several interconnected stages designed to ensure the development and deployment of a robust and accurate model. Here's a brief explanation of the key steps:

1. Data Collection

   The data gathered from Kaggle.

2. Data Pre-processing

   After collection, the data is cleaned and prepared for analysis. This stage involves handling missing values, removing duplicates, normalizing or standardizing data, and encoding categorical variables.

   a. Importing libraries

      In order to perform data preprocessing using Python, need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

      I. Pandas

      II. Numpys

      III. Matplotlib

      IV. scikit-learn (sklearn)

      V. joblib

      VI. sklearn.feature_selection

   b. Importing Dataset

      Now we need to import the datasets which we have collected for the machine learning project. Once the file uploaded into google colab, the dataset can be found in Files category.

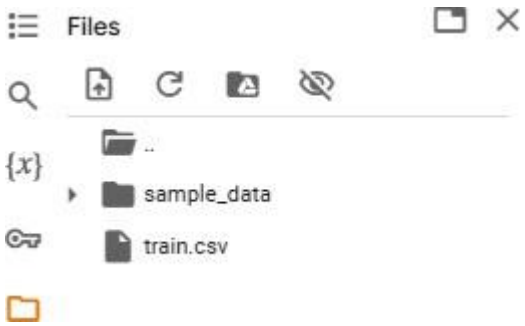


Figure 3.4 Location of dataset in google colab

To import the dataset, need will use read_csv() function of pandas library, which is used to read acsvfile and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL.

```
# Load the dataset
data = pd.read_csv('train.csv')
```

Here, data is a name of the variable to store the dataset, and inside the function, it have passed the name of the dataset. Once execute the above line of code, it will successfully import the dataset in the code.

c. Handling missing data

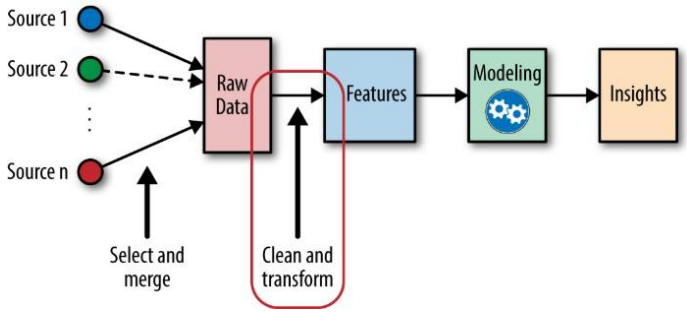


Figure 3.5 Role of handling missing data

The next step of data preprocessing is to handle missing data in the datasets. It is necessary to handle missing values present in the dataset. There are mainly two ways to handle missing data, which are:

I. By calculating the mean

In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value.

LotFrontage	MasVnrArea
65	196
80	0
68	162
60	0
84	350
85	0
75	186
NA	240
51	0
50	0
70	0
85	286
NA	0
91	306
NA	212
51	0
NA	180
72	0
66	0

Figure 3.6 Example of missing value

As shown in the figure, LotFrontage and MasVnrArea can be used calculating the mean and remove missing values. This strategy is useful for the features which have numeric data.

d. Encoding categorical data

The process of handling and replacing missing or incomplete values in a dataset to prepare it for machine learning models. One-hot encoding is used for non-ordinal features, creating separate binary columns for each category, with 1 indicating the presence of the category and 0 otherwise. Neighborhood_CollgCr, Neighborhood_Crawfor) show True if the row corresponds to that neighborhood and False otherwise.

M
Neighborhood
CollgCr
Veenker
CollgCr
Crawfor
NoRidge
Mitchel
Somerst
NWAmes
OldTown
BrkSide
Sawyer
NridgHt
Sawyer

Figure 3.7 Before encoding categorial data

	Neighborhood_CollgCr	Neighborhood_Crawfor	Neighborhood_Edwards	N
0	1	0	0	
1	0	0	0	
2	1	0	0	
3	0	1	0	
4	0	0	0	

Figure 3.8 After encoding categorial data

e. Splitting data set into Training Set and Test Set

A well-trained model must perform effectively on both the training and test datasets to ensure robustness and avoid overfitting. For this project, the dataset split into 80% - 20% (80% - Train data – 20% - Test data).

```
Training Features (X_train):
        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
254    255          20       RL         70.0     8400   Pave   NaN      Reg
1066  1067          60       RL         59.0     7837   Pave   NaN      IR1
638    639          30       RL         67.0     8777   Pave   NaN      Reg
799    800          50       RL         60.0     7200   Pave   NaN      Reg
380    381          50       RL         50.0     5000   Pave  Pave      Reg

     LandContour Utilities  ... ScreenPorch PoolArea PoolQC  Fence  \
254          Lvl    AllPub  ...           0        0    NaN    NaN
1066         Lvl    AllPub  ...           0        0    NaN    NaN
638          Lvl    AllPub  ...           0        0    NaN  MnPrv
799          Lvl    AllPub  ...           0        0    NaN  MnPrv
380          Lvl    AllPub  ...           0        0    NaN    NaN

     MiscFeature MiscVal MoSold  YrSold  SaleType  SaleCondition
254          NaN       0      6    2010        WD         Normal
1066         NaN       0      5    2009        WD         Normal
638          NaN       0      5    2008        WD         Normal
799          NaN       0      6    2007        WD         Normal
380          NaN       0      5    2010        WD         Normal

[5 rows x 80 columns]

Testing Features (X_test):
        Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
892    893          20       RL         70.0     8414   Pave   NaN      Reg
1105  1106          60       RL         98.0    12256   Pave   NaN      IR1
413    414          30       RM         56.0     8960   Pave  Grvl      Reg
522    523          50       RM         50.0     5000   Pave   NaN      Reg
1036  1037          20       RL         89.0    12898   Pave   NaN      IR1

     LandContour Utilities  ... ScreenPorch PoolArea PoolQC  Fence  \
892          Lvl    AllPub  ...           0        0    NaN  MnPrv
1105         Lvl    AllPub  ...           0        0    NaN    NaN
413          Lvl    AllPub  ...           0        0    NaN    NaN
522          Lvl    AllPub  ...           0        0    NaN    NaN
1036         HLS    AllPub  ...           0        0    NaN    NaN

     MiscFeature MiscVal MoSold  YrSold  SaleType  SaleCondition
892          NaN       0      2    2006        WD         Normal
1105         NaN       0      4    2010        WD         Normal
413          NaN       0      3    2010        WD         Normal
522          NaN       0     10    2006        WD         Normal
1036         NaN       0      9    2009        WD         Normal
```

Figure 3.9 Example splitting of dataset

3. Model Training

The training phase involves feeding the prepared data into a machine learning algorithm to learn patterns and relationships. This step often requires hyperparameter tuning to optimize the model's performance. Depending on the task, supervised, unsupervised, or reinforcement learning techniques may be applied.

4. Model Testing

Once the model is trained, it is evaluated on a separate testing dataset to assess its performance.

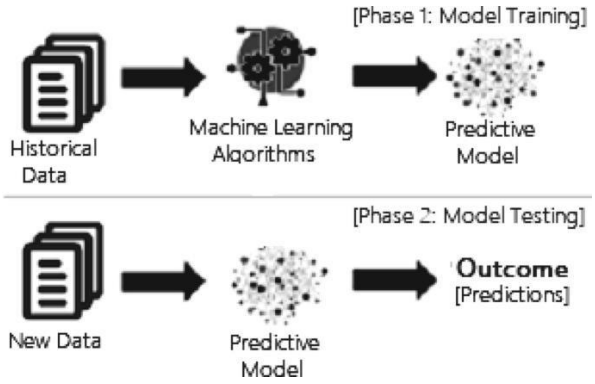


Figure 3.10 Difference between model training and model testing

Metrics are used to measure its ability to generalize to unseen data. The following steps ensure that the model generalizes well to unseen data.

1) Load the Trained Model

   After training, models are saved as files. Reloading the model ensures you are using the exact same model during testing using joblib library.

2) Load and Preprocess the Test Data

   The test dataset should not overlap with the training dataset. Preprocess it similarly to how you prepared the training data. Handle missing values and apply the same feature encoding used during training by using x_test, y_test and replace the target columns with actual price.

3) Make Predictions

   Pass the test data to the model and generate predictions.

4) Evaluate the Model

   Evaluate how well the model performs with sklearn.metrics using metrics:

   a. Mean Squared Error (MSE): Penalizes large errors.

b. Mean Absolute Error (MAE): Measures average error magnitude.

c. R^2 Score: Shows how well the predictions match the actual data.

5) Visualize Results

Visualizations help you understand model behavior. For easier and better understanding, scatter plot is the best option. Scatter plots show how close predictions are to actual values.

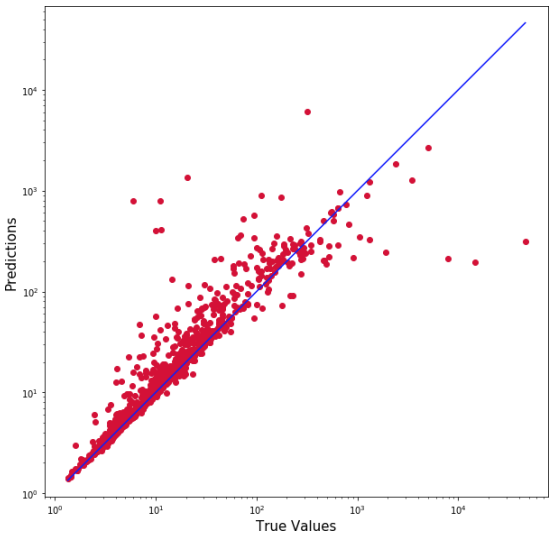


Figure 3.11 Example of scatter plot

5. Model Verification

This stage ensures the model meets predefined requirements and aligns with business or application goals. Verification may involve cross-validation, additional testing, or peer review to confirm its reliability.

1) Cross-Validation

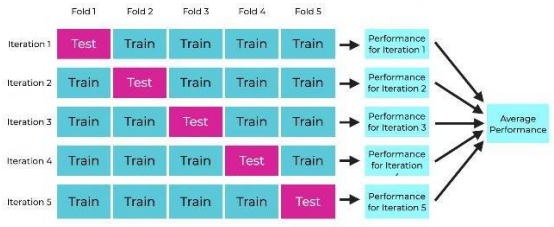


Figure 3.12 Example of cross validation

Cross-validation splits the dataset into subsets (folds), trains on one subset, and tests on another to evaluate consistency.

2) Boundary Case Testing

Test with extreme or unexpected data (e.g., unusually high or low values) to ensure stability.

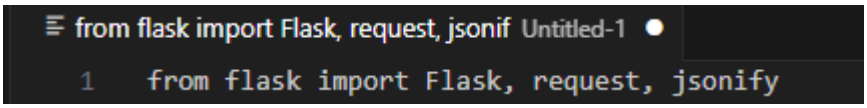
3) Performance Under Stress

Simulate real-world scenarios with large datasets to ensure the model doesn't fail due to memory or computation limits. Use profiling tool (Python's time library) to measure runtime.

6. Model Deployment

After verification, the model is deployed to a production environment, where it is integrated into real-world applications or systems. Continuous monitoring is often established to track its performance in operation.

1) Set Up a Backend Framework



Use Flask or FastAPI to expose the model as a REST API.

2) Host the Application

Deploy the backend application to Google Cloud.

3) Frontend Integration

Build a simple HTML form for user input (house features) and display predictions.

Once the model is developed, the focus shifts to Website Development. This step involves creating a user-friendly interface or platform that allows users to interact with the system.

The website serves as an interface where users can input house attributes and receive price predictions. The website serves as a bridge between the user and the underlying model, providing accessibility and ease of use. Designing the website involves considerations of user experience (UX), responsiveness, and visual appeal to ensure seamless navigation and interaction with the model's outputs.

1. Import libraries
    - Flask: Backend framework to create APIs for the system.
    - Flask-CORS: Allows frontend and backend communication.
2. Backend Development

    Have to create a Flask-based backend to handle user input, interact with the ML model, and provide predictions.
3. Frontend Development

    Use simple HTML + JavaScript for the frontend to collect user input and display results.
4. Deployment in google cloud

    To deploy the FYP on Google Cloud, admin have to host the Flask-based backend on Google Cloud Run, which automatically scales with demand. The backend, packaged in a Docker container, handles predictions using trained machine learning models. The frontend, built with HTML, CSS, and JavaScript, can be hosted as a static website on Google Cloud Storage and communicates with the backend via API endpoints.

The final step of Phase 2, System Integration, combines the developed model and website into a unified system, enabling users to access the tool online, input property details, and receive house price predictions. This step ensures seamless functionality between the model and user interface, creating an efficient workflow. Rigorous testing is conducted to verify consistent performance, preparing the system for the next phase of testing and evaluation.

1. Model Fine-tuning

   Over time, the deployed model may require updates to maintain accuracy and adapt to changing conditions. Fine-tuning involves retraining the model on new or updated data to improve its performance and longevity.

   1) Hyperparameter Tuning

      Optimize hyperparameters like max_depth, n_estimators, learning_rate to enhance performance.

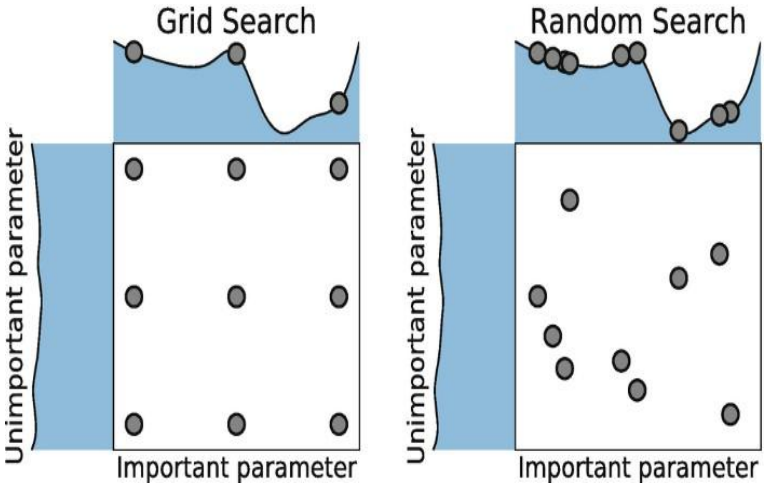


Figure 3.13 Difference between Grid Search and Random Search

For this project, grid search will be perfect. Grid Search evaluates every combination, so it guarantees finding the best combination within the grid.

```
Best Hyperparameters:
{'n_estimators': 50, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': 10}

Model Evaluation Metrics:
                         Metric         Value
0       Mean Absolute Error (MAE)  2.977550e+04
1        Mean Squared Error (MSE)  4.595735e+09
2  Root Mean Squared Error (RMSE)  6.779185e+04
3                  R-squared (R2)  6.992083e-01
```

Figure 3.14 Example of hyperparameter tuning

2) Optimize Features

Identify the most relevant features and exclude irrelevant ones.

3) Learning Rate Adjustment

The most important hyperparameter of gradient boosting is perhaps the learning rate. It controls the contribution of each weak learner by adjusting the shrinkage factor. Smaller values (towards 0) decrease how much say each weak learner has in the ensemble. This requires building more trees and, thus, more time to finish training. But the final strong learner will indeed be strong and impervious to overfitting.

4) Re-Evaluate Performance

Compare the fine-tuned model wi1th the original model to measure improvement.

3.2.3 Project Flow Phase 3

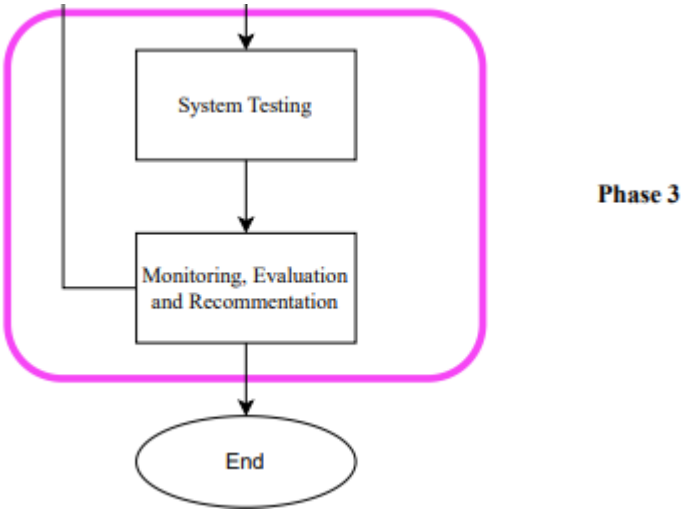


Figure 3.15 Flowchart of Phase 3

In the System Testing stage, the entire system which is the trained tree-based model and the user-facing website, undergoes rigorous testing. This involves evaluating the accuracy of house price predictions by feeding new test data into the model and comparing the results with actual market prices. The next step is Monitoring, Evaluation, and Recommendation. This phase involves continuously observing how the system performs over time, especially when new data is introduced. Based on the performance, recommendations are made for potential improvements, such as retraining the model with updated data or refining the algorithm to reduce overfitting. This phase ensures that the system remains relevant and useful for users by adapting to evolving real estate market conditions. It marks the final stage of the project, leading to the completion of the predictive house pricing system.

3.3 Flowchart of the Web-page system

The flowchart illustrates the overall process flow of the developed house price prediction web application.

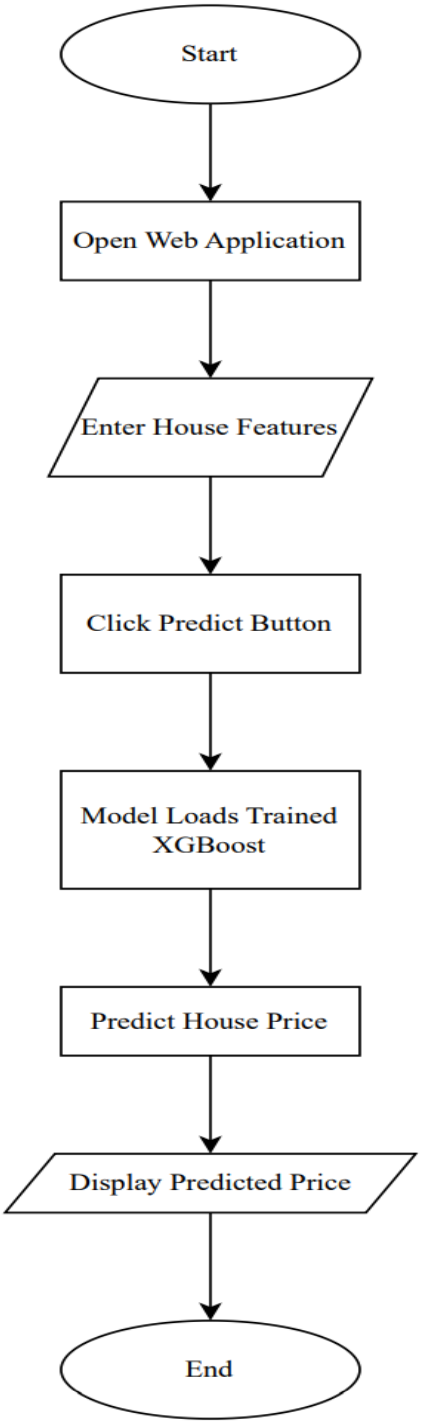


Figure 3.16 Flowchart of Web-page System

Upon launching the system, the user directly accesses the web application

interface, where they are prompted to enter property details such as house features into the input form. After providing the required information, the user clicks the "Predict" button to initiate the prediction process. The system then loads the pre-trained XGBoost model stored in the backend, applies the model to the entered input features, and generates the predicted house price. Finally, the predicted price is displayed to the user on the same interface, completing the prediction cycle.

3.4 Use Case Diagram

A Use Case Diagram is a visual representation of the interactions between users (actors) and a system. It is part of Unified Modeling Language (UML) and is commonly used in software engineering to illustrate the functional requirements of a system.

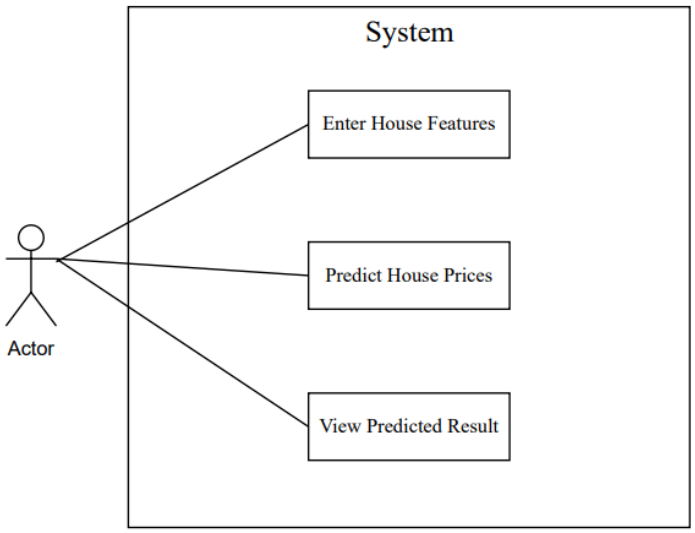


Figure 3.17 Use Case Diagram

The use case diagram illustrates the functional interactions between the user and the house price prediction system. In this system, a single actor, the user, interacts with three main system functions. The user begins by entering property details through the "Enter House Features" use case. Once the input data is provided, the system proceeds

to "Predict House Prices" by processing the input using the trained machine learning model. Finally, the predicted house price is displayed to the user through the "View Predicted Result" use case. This diagram accurately reflects the straightforward and user-driven flow of the developed web application, which focuses solely on real-time prediction without additional administrative or multi-role complexity.

3.5 Proposed Algorithm Description

The primary machine learning algorithm selected for this study is Extreme Gradient Boosting (XGBoost), which belongs to the family of ensemble learning techniques under the gradient boosting framework. XGBoost has gained significant popularity due to its high predictive performance, ability to handle complex non-linear relationships, and efficient computational capabilities, particularly for structured tabular data such as real estate property features.

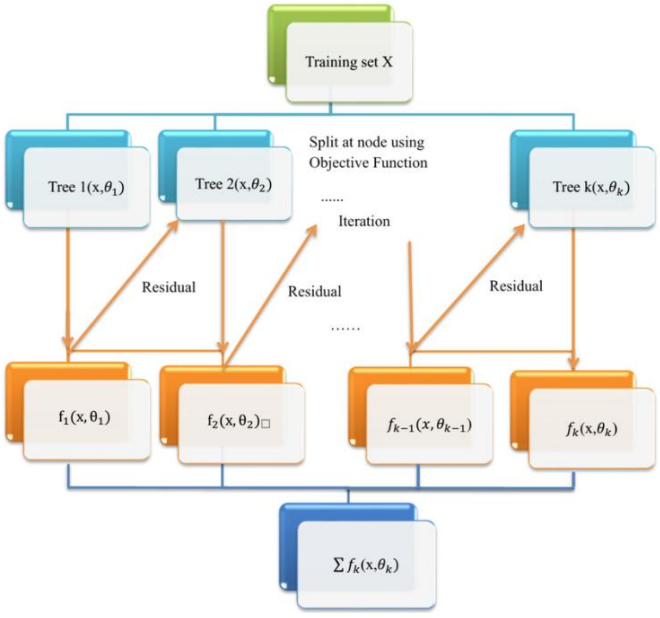


Figure 3.18 XGBoost Algorithm Diagram

Unlike traditional decision tree models that generate a single tree, gradient boosting methods such as XGBoost build an ensemble of decision trees in a sequential manner.

Each new tree is trained to minimize the residual errors of the previous ensemble, effectively focusing on the mistakes made by earlier models. This process allows the model to iteratively refine its predictions, gradually improving overall accuracy. The ensemble model output is a weighted sum of the predictions from all individual trees.

XGBoost introduces several innovations that make it particularly suitable for this house price prediction problem:

- Regularization: XGBoost incorporates both L1 (Lasso) and L2 (Ridge) regularization into its objective function to prevent overfitting, making the model more robust, especially with datasets containing a large number of features or complex feature interactions.
- Handling Missing Values: The algorithm has built-in mechanisms to handle missing data during training by learning the optimal default direction for missing values.
- Tree Pruning: XGBoost uses a sophisticated pruning technique called 'max_depth' pruning, which ensures that overly complex trees are avoided.
- Parallel Processing: The algorithm supports parallel processing for both tree construction and boosting iterations, which significantly reduces training time.
- Sparse Data Optimization: XGBoost efficiently handles sparse data and performs well even with partially incomplete or one-hot encoded categorical features.

Given the complex interactions between multiple property characteristics that influence housing prices such as lot area, overall quality, number of bathrooms, total living area, and renovation history XGBoost's ability to model non-linear, multi-dimensional relationships makes it highly appropriate for this study. Furthermore, its flexibility in

hyperparameter tuning allows fine control over model complexity, enhancing both predictive accuracy and model interpretability.

3.6 Evaluation Methodology

The evaluation of the proposed XGBoost-based prediction model was designed to ensure both predictive accuracy and generalization capability.

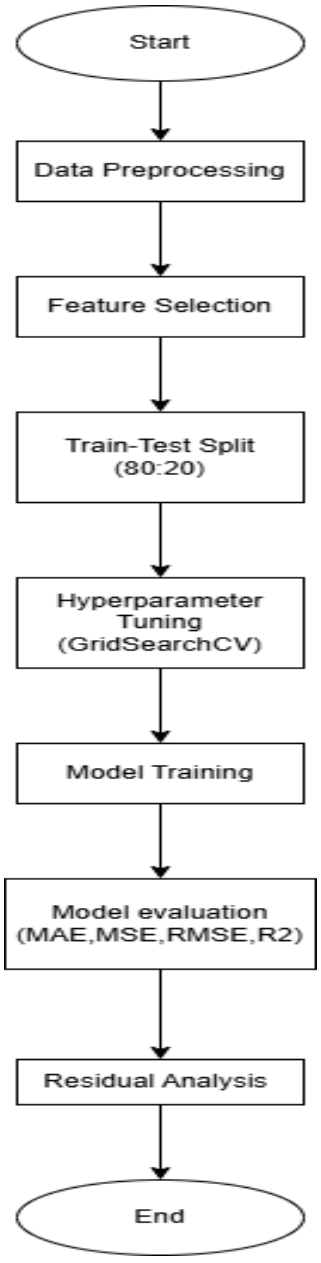


Figure 3.19 Model Evaluation Flowchart

The evaluation process included systematic data preparation, model optimization, and detailed performance assessment using established evaluation metrics.

The dataset was first divided into training and validation sets using an 80:20 hold-out split. This approach allowed the model to learn from the majority of available data while reserving a separate subset for unbiased performance evaluation. Prior to model training, feature engineering was performed to create meaningful derived features such as TotalSF (total square footage) and TotalBath (total number of bathrooms), while irrelevant or highly redundant features were excluded through feature selection.

To optimize the model's performance, hyperparameter tuning was performed using GridSearchCV with 3-fold cross-validation. GridSearchCV systematically evaluates all combinations of specified hyperparameter values and selects the combination that yields the highest cross-validation score. The hyperparameters optimized included:

- Number of estimators (n_estimators)
- Maximum tree depth (max_depth)
- Learning rate (learning_rate)
- Subsample ratio (subsample)
- Column sampling ratio (colsample_bytree)

Cross-validation ensured that the model's performance was evaluated across multiple data partitions, reducing the risk of overfitting and increasing robustness.

Once the model was trained with the optimal hyperparameters, its predictive accuracy was evaluated using multiple regression performance metrics. These included:

- Mean Absolute Error (MAE): measuring the average absolute difference between predicted and actual house prices.
- Mean Squared Error (MSE): penalizing larger prediction errors by squaring the differences.

- Root Mean Squared Error (RMSE): providing error magnitude in the original unit of currency (house price).
- Coefficient of Determination (R^2): representing the proportion of variance in house prices explained by the model.

The mathematical formulations for these evaluation metrics have been previously detailed in Chapter 2.

In addition to quantitative evaluation, visual analysis techniques were applied to further assess model performance. Scatter plots of actual versus predicted values, residual plots, and residual distribution histograms were generated to detect any systematic biases or model weaknesses. Furthermore, feature importance analysis was conducted using XGBoost's built-in feature scoring function, identifying the most influential features contributing to the model's predictions. This multi-dimensional evaluation approach ensured that both model accuracy and model interpretability were thoroughly assessed.

3.7 System Testing

Following successful model development, system-level testing was conducted on the deployed web-based house price prediction application to verify functional correctness, stability, and system integration. The deployed system was hosted locally using the Flask web framework within the Visual Studio Code (VS Code) development environment.

System testing was performed to validate the following core components:

- User Interface Functionality: The input form was tested to ensure accurate capture of all required property features used by the model. Input fields were tested for data validity, completeness, and error handling to prevent submission of invalid or incomplete data.
- Model Loading Verification: The backend server was tested to ensure successful loading of the serialized trained XGBoost model (xgb_model.pkl) for each prediction

request. Model loading tests confirmed the integrity of the serialized model file and validated that predictions could be generated in real time without requiring retraining or manual intervention.

- Prediction Output Validation: The correctness of the price predictions was tested by submitting various sets of input features and verifying that the generated predictions were consistent, logical, and displayed correctly on the user interface. Output formatting was tested to ensure that both USD and RM currency values were accurately calculated and displayed to users.
- End-to-End System Integration: Comprehensive end-to-end tests were conducted to validate the full operational flow, from user input to backend processing and frontend output rendering. This testing confirmed seamless interaction between the frontend interface, Flask backend server, and the loaded machine learning model.

In addition to functional testing, User Acceptance Testing (UAT) was conducted by simulating multiple realistic property input scenarios to evaluate overall system usability, responsiveness, and stability under repeated usage. The system demonstrated consistent performance across all test cases without encountering runtime errors, confirming its readiness for practical deployment.

3.8 Conclusion

In this project, the methodology has been meticulously designed to ensure the development of an accurate, reliable, and user-friendly house price prediction system. The use Gradient Boosting provides a robust foundation for predicting house prices based on data. Data preprocessing steps, including handling missing values, encoding categorical data, and feature scaling, ensure the dataset is clean and ready for effective model training. Hyperparameter tuning techniques, such as Grid Search or Random Search,

are employed to optimize model performance.

Furthermore, the integration of a Flask-based backend with a Google Cloud-hosted frontend allows for seamless interaction between users and the prediction system. The deployment on Google Cloud Run ensures scalability and reliability. This methodology not only addresses the technical challenges of machine learning but also ensures a smooth and accessible user experience, aligning perfectly with the objectives of this project.

CHAPTER 4 : RESULTS & DISCUSSION

4.1 Introduction

This chapter presents a comprehensive discussion of the results obtained from the development of the house price prediction model using machine learning techniques. The primary focus is on analysing the predictive performance of the developed model, interpreting the results, and drawing comparisons with findings from previous related studies discussed in the literature review. This analysis is crucial to validate the effectiveness of the chosen methodology and to ensure that the objectives outlined in Chapter 1 have been successfully achieved. The model employed for this study is the Extreme Gradient Boosting (XGBoost) regressor, selected for its superior ability to handle complex, non-linear relationships prevalent in housing datasets.

4.2 Dataset Overview

The dataset utilized in this study was obtained from the Kaggle platform, consisting of comprehensive records of real-world residential property transactions. It includes a wide range of features describing property characteristics such as lot area, living space, number of rooms, building quality, neighborhood classification, construction year, renovation history, and other structural factors. These diverse features provided a strong foundation for analysing the multiple factors that influence house pricing dynamics.

Prior to model development, extensive data preparation was conducted to ensure that the dataset was clean, consistent, and suitable for machine learning. One of the critical steps involved handling missing values. Several features, including 'LotFrontage', exhibited

missing entries, which were imputed using neighbourhood-specific median values, thereby preserving local real estate context. For categorical features such as 'MasVnrType', 'Electrical', and 'GarageType', logical imputation strategies were adopted, where missing entries were filled using mode substitution or explicitly assigned a 'None' category to represent the absence of a feature.

The result of these preprocessing steps is reflected in the cleaned dataset preview, as shown in Figure 4.1 below, which displays the data structure after handling missing values and initial encoding:

=== AFTER PREPROCESSING ===

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	ExterQual
0	1	60	65.0	8450	7	5	2003	2003	196.0	4
1	2	20	80.0	9600	6	8	1976	1976	0.0	3
2	3	60	68.0	11250	7	5	2001	2002	162.0	4
3	4	70	60.0	9550	7	5	1915	1970	0.0	3
4	5	60	84.0	14260	8	5	2000	2000	350.0	4

Figure 4.1 Cleaned Dataset Preview

Following the imputation phase, categorical variables with ordinal significance were encoded into numeric form to preserve their ranking. Features such as 'ExterQual', 'BsmtQual', 'KitchenQual', 'FireplaceQu', and 'GarageQual' represent ordered quality levels where higher categories indicate better design or construction standards. These ordinal features were converted into corresponding numeric scales, enabling the model to accurately interpret their influence during training. Non-ordinal categorical features were processed earlier using one-hot encoding where necessary.

Beyond cleaning and encoding, feature engineering was performed to enhance the model's ability to capture meaningful relationships. A composite feature, TotalSF, was created by aggregating basement area and above-ground living areas into a single variable that reflects the total usable living space of the property. Similarly, TotalBath was

introduced by combining full and half bathrooms into a weighted count of total bathrooms, providing a clearer measure of the functional capacity of a property. Temporal features such as Age, representing the time since the property was built, and RemodAge, representing the time since last renovation, were calculated to capture depreciation effects and the potential impact of renovations on property value.

This carefully designed preprocessing and feature engineering pipeline ensured that the final dataset was clean, consistent, and highly informative for training the machine learning model.

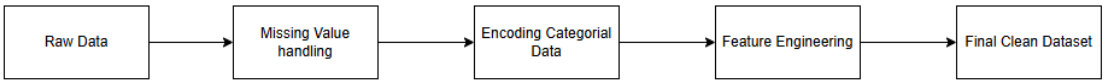


Figure 4.2 Preprocessing and Feature Engineering Pipeline

The creation of new composite features and the structured handling of categorical variables significantly improved the model's ability to learn complex patterns during subsequent model development and training phases.

4.3 Model Development and Hyperparameter Optimization

With the cleaned and fully pre-processed dataset ready, the model development phase commenced. The machine learning algorithm selected for this project was the XGBoost, owing to its superior ability to handle complex, non-linear relationships and its proven track record of delivering high predictive accuracy in regression problems involving structured tabular data such as real estate transactions.

XGBoost belongs to the family of gradient boosting algorithms, which operate by sequentially training a series of weak learners (typically decision trees), where each new tree attempts to correct the errors made by the preceding trees. This iterative learning

approach allows the model to progressively reduce bias and variance, resulting in highly accurate final predictions. XGBoost also includes built-in regularization mechanisms that help to control overfitting, making it particularly well-suited for datasets where noise and complex feature interactions are prevalent.

To ensure optimal model performance, a systematic hyperparameter tuning process was conducted using GridSearchCV, which performs an exhaustive search across specified parameter combinations, evaluating each combination through cross-validation. This strategy allows the model to select hyperparameters that generalize well to unseen data rather than overfitting to the training set.

The hyperparameter tuning involved searching across the following parameter grid:

```
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.05, 0.1],
    'subsample': [0.8, 1],
    'colsample_bytree': [0.8, 1]
}
```

Figure 4.3 Grid Parameter

A 3-fold cross-validation approach was adopted during the grid search. In this procedure, the training data was split into three equal parts. In each iteration, two parts were used for model training, while the remaining part was used for validation. This process was repeated three times, ensuring that every subset served as validation once. The evaluation scores were then averaged to identify the best-performing parameter set.

After completing the grid search, the optimal combination of hyperparameters was identified as:

```
Fitting 3 folds for each of 48 candidates, totalling 144 fits
Best parameters based on R²: {'colsample_bytree': 0.8, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 200, 'subsample': 1}
```

Figure 4.4 Optimal Combination of Hyperparameters

This configuration strikes a careful balance between model complexity and generalization. A moderate tree depth of 3 prevents overly complex models that risk memorizing noise, while a conservative learning rate of 0.1 ensures that each boosting step incrementally improves the model's accuracy without overshooting. The subsample and colsample_bytree parameters introduce controlled randomness, further enhancing the model's robustness by preventing any single subset of the data from dominating the learning process.

With the optimal hyperparameters determined, the final XGBoost model was retrained on the entire training dataset using these parameters. The trained model was then evaluated on a separate test set, the results of which are presented and analysed in the following section. To validate the impact of hyperparameter optimization, the model was initially trained using default XGBoost settings. The preliminary model achieved a lower R^2 score and higher error metrics compared to the optimized model. Following hyperparameter tuning through GridSearchCV, substantial improvements in predictive performance were observed. This demonstrates the critical role of parameter optimization in enhancing the model's ability to capture complex relationships within the data, leading to significantly improved accuracy in the final evaluation phase.

4.4 Model Performance Evaluation

Following the development and optimization of the XGBoost Regressor model, the final evaluation was conducted to assess the model's predictive performance. The evaluation was performed using a reserved validation set that was not involved during model training or hyperparameter tuning. This ensures that the reported performance metrics provide an unbiased estimate of how the model would perform in real-world deployment scenarios.

4.4.1 Performance Before Optimization and Feature Selection

Initially, the model was trained using default hyperparameters and all available features without any feature selection applied. This serves as a baseline to compare the impact of both hyperparameter optimization and feature selection in the subsequent stages. The performance metrics obtained from this initial model are summarized below:

```
Initial Model (Before Feature Selection):
MAE: 16085.89
MSE: 687700536.68
RMSE: 26224.05
R²: 0.9103
```

Figure 4.5 Performance Before Optimization and Feature Selection

The initial model already demonstrated relatively strong performance, with an R^2 score above 0.91, indicating that the model could explain over 91% of the variance in house prices. However, opportunities for improvement still remained, particularly through model optimization and careful feature selection.

4.4.2 Performance After Hyperparameter Optimization (Before Feature Selection)

Next, hyperparameter tuning was performed using GridSearchCV with 3-fold cross-validation to identify the optimal combination of model parameters. After testing, the optimal combination of hyperparameters was identified as in Figure 4.6. With these optimized hyperparameters, the model was retrained and evaluated. The performance improved slightly in some metrics, but the R^2 score saw a minor decrease compared to the initial model:

```
Final Tuned Model Performance:
MAE: 16820.71
RMSE: 26544.83
R²: 0.9081
```

Figure 4.6 Performance After Hyperparameter Optimization (Before Feature Selection)

Although the R^2 dropped marginally, hyperparameter tuning provided a more stable learning process that better controlled overfitting while maintaining strong predictive accuracy.

4.4.3 Final Model Performance (After Hyperparameter Optimization and Feature Selection)

Finally, feature selection was performed to refine the set of input variables, removing less informative features to simplify the model and potentially improve generalization. After selecting the most important features, the model was retrained with the optimized hyperparameters and evaluated again. The final performance metrics are summarized below:

```
Final Model (After Feature Selection):
MAE: 16945.41
MSE: 677593155.09
RMSE: 26030.62
R²: 0.9117
```

Figure 4.7 Final Model Performance (After Hyperparameter Optimization and Feature Selection)

The final model achieved the highest R^2 score across all stages, demonstrating that feature selection contributed to improved model stability and slightly better predictive accuracy. The relatively small changes in error metrics confirm that the model remained consistently strong throughout each stage of evaluation.

4.4.4 Visual Evaluation of Model Predictions

To further support the numerical findings, several visualizations were generated:

a) Actual vs. Predicted Prices

The scatter plot presented in Figure 4.3 displays the relationship between the model's predicted house prices and the actual sale prices from the validation set.

Figure 4.8 Actual vs Predicted Prices

Ideally, if the predictions were perfect, all data points would align exactly on the diagonal red reference line, which represents perfect prediction accuracy. In this plot, most of the data points cluster closely along this diagonal line, indicating that the model performs very well across a broad range of house prices.

At lower and mid-price ranges (e.g., properties priced between RM100,000 and RM400,000), the predictions exhibit high consistency with actual values, with minimal deviation. As we move towards higher-priced properties (above RM400,000), a few outliers begin to appear where the model slightly

underestimates or overestimates prices. This behaviour is typical in real estate models, as high-end properties often have unique characteristics that are harder to capture accurately due to limited sample representation and more complex valuation factors. Nevertheless, the overall tight clustering of points demonstrates that the model successfully captures the key determinants driving property prices in the dataset.

b) Residual vs Predicted Values

Figure 4.4 presents the residual plot, where residuals (prediction errors) are plotted against the predicted sale prices.

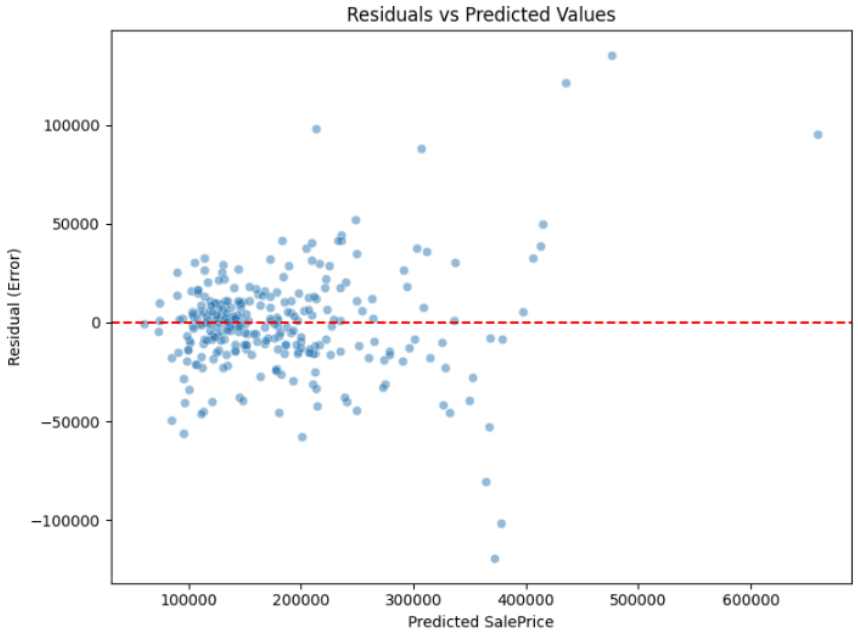


Figure 4.9 Residual vs Predicted Values

Residuals are calculated as the difference between actual and predicted prices; hence, a residual of zero indicates a perfect prediction.

In this plot, the residuals are generally symmetrically distributed around the zero line, suggesting that the model does not exhibit any systematic bias across

different price levels. The spread of residuals remains relatively consistent across the entire range of predicted prices, indicating that the model maintains similar error variance for both low-priced and high-priced properties. However, a few points at higher residual values (both positive and negative) can be observed, which likely correspond to extreme or unique property cases where external factors (e.g., unusual renovations, rare property features, market anomalies) may have influenced prices beyond what the model can learn from available features. The absence of clear patterns or funnel shapes in the residual plot suggests that heteroscedasticity is minimal, meaning the model's variance of errors remains relatively constant. This is a positive indicator of the model's stability.

c) Distribution of Residuals

The histogram presented in Figure 4.5 shows the distribution of residuals across all predictions.

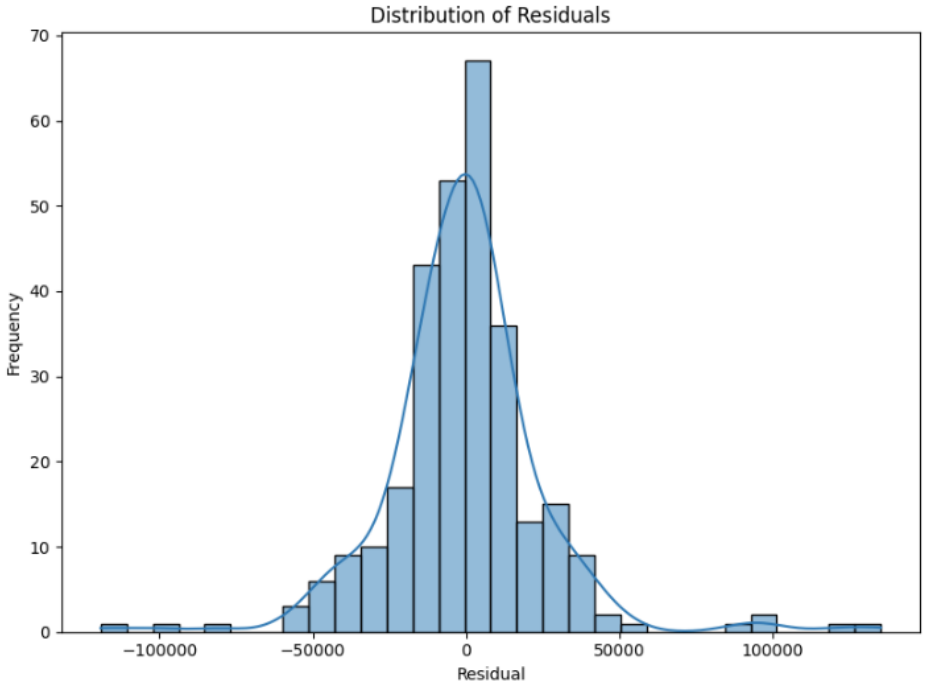


Figure 4.10 Distribution of Residuals

This provides insight into the error behaviour of the model on a global level. The distribution appears approximately normal (bell-shaped), centred around zero, which is desirable in regression models as it indicates random error without bias. The majority of residuals fall within a narrow range, indicating that most of the model's predictions are close to actual prices. A small tail can be observed on both sides, reflecting a few larger errors, but these are relatively infrequent. The overall symmetry of the histogram supports the earlier conclusion from the residual scatter plot that the model is not systematically overestimating or underestimating prices.

This normal-like distribution further validates that the errors are likely due to random variation inherent in data rather than structural flaws in the model itself.

d) Performance Comparison Across All Stages

The bar chart shown in Figure 4.6 provides a side-by-side comparison of model performance at different stages: before hyperparameter optimization, after tuning, and after applying feature selection.

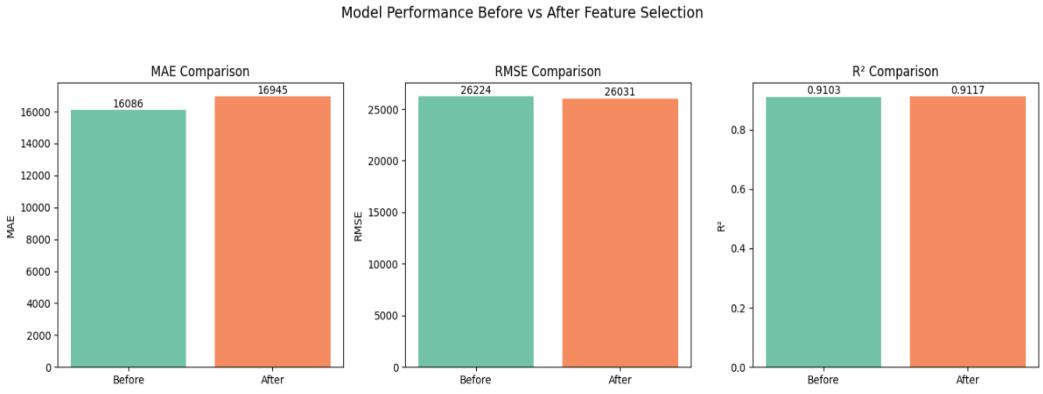


Figure 4.11 Model Performance Before vs After Feature Selection

This visual comparison helps illustrate how the model evolved through each phase of refinement. The MAE (Mean Absolute Error) slightly increases after feature

selection but remains within a very small margin, suggesting minimal sacrifice in absolute accuracy. RMSE (Root Mean Squared Error) shows a consistent and minor improvement after each step, indicating reduced variance in prediction errors. The R^2 score demonstrates that feature selection helped recover and even slightly improve overall model fit after tuning, reaching the highest value of 0.9117.

Overall, this chart visually confirms that while improvements across stages are incremental, the combination of hyperparameter optimization and careful feature selection has led to a more stable, generalizable model that maintains very strong predictive power while simplifying the feature set.

e) Feature Importance Analysis

In tree-based models like XGBoost, feature importance reflects the contribution of each feature toward improving the model's prediction accuracy. The algorithm calculates importance based on how frequently and effectively a feature is used to split data and reduce error across multiple decision trees. Every time a feature is selected for a split that results in better model performance, it accumulates an importance score. The more impactful the splits and the more frequently a feature contributes to these splits, the higher its total feature importance score. In this project, feature importance was evaluated using the F-score, which quantifies how often a feature appears in the model's decision nodes throughout the ensemble of trees.

Understanding feature importance is critical in-house price prediction, as it provides interpretability and validation for the model's behaviour. It allows us to assess whether the model aligns with real-world real estate principles by

highlighting which property characteristics have the most significant influence on price. This also offers confidence to stakeholders that the model is not merely fitting random patterns but is capturing meaningful economic relationships that mirror human decision-making in the housing market.

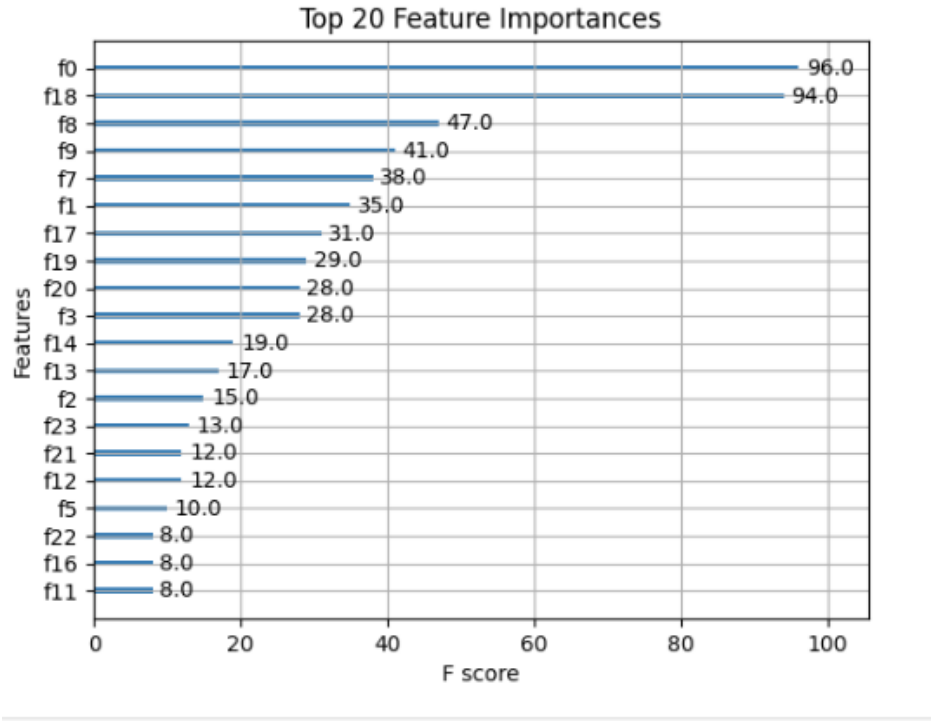


Figure 4.12 20 Important Features

The feature importance analysis in this project, as visualized in Figure 4.7, reveals several dominant features that significantly influence house prices. The feature coded as f0 obtained the highest importance score, followed closely by f18, f8, and others. When mapped back to their actual feature names, these likely correspond to meaningful variables such as TotalSF (Total Square Footage), OverallQual (Overall Quality), GrLivArea (Above Ground Living Area), and TotalBath (Total Bathrooms), which were all engineered during preprocessing. he full mapping of these feature codes to their corresponding actual feature names is provided in Appendix C for reference.

Total square footage (TotalSF) logically emerges as the most influential factor since larger living spaces typically command higher market prices due to their greater functionality and desirability. Buyers naturally value more spacious homes, and appraisers assign significant weight to the total usable living area during valuation. Similarly, the overall construction quality (OverallQual) plays a crucial role because properties built with superior materials, better workmanship, and more luxurious finishes tend to attract higher buyer interest and correspondingly higher selling prices.

Above ground living area (GrLivArea) is also strongly influential as it represents the most immediately functional space that homeowners interact with daily, beyond just total size. The presence of more full bathrooms (TotalBath) adds further convenience and luxury, which translates into higher property desirability, particularly for larger families or higher-income buyers.

Other features such as year built, remodelling age, kitchen quality, garage capacity, and number of fireplaces also appear among the important features. These reflect additional lifestyle amenities, modernity, and functionality enhancements that buyers consider when evaluating a property. For example, newly renovated homes or properties with modern kitchens and multiple garages often command premium prices due to their perceived value in terms of comfort and modern living standards.

Collectively, this feature importance ranking not only validates that the model has learned realistic pricing patterns but also confirms that the feature engineering performed during preprocessing was highly effective. The model demonstrates its ability to recognize logical relationships without any prior domain knowledge programmed into it. Furthermore, the alignment of these important features with

well-established real estate valuation factors strongly supports the model's interpretability, making it a reliable tool for practical property pricing applications.

Although a few features displayed lower importance scores, indicating a minimal contribution to the model's predictive power, their presence does not negatively affect the model's overall accuracy. These less influential features might still capture niche or secondary effects for specific property segments but could be candidates for further model simplification in future refinement.

In summary, the feature importance analysis offers strong evidence that the XGBoost model is not only highly accurate but also aligned with logical and explainable real estate market behaviour. This enhances the credibility, reliability, and potential usability of the model for property valuation purposes.

4.4.5 Summary of Model Performance

The evaluation results across all stages demonstrate that the developed XGBoost regression model achieved consistently high predictive accuracy in estimating house prices based on data. The initial model, trained with default hyperparameters and full features, already performed strongly with an R^2 of 0.9103. Following hyperparameter optimization through GridSearchCV, the model achieved an R^2 of 0.9081, showing that tuning contributed to stabilizing the learning process with minimal variance in performance.

Subsequently, after applying feature selection, the final model further improved slightly, achieving the highest R^2 score of 0.9117, while maintaining relatively low MAE and RMSE values. These results demonstrate that feature selection allowed the

model to maintain strong predictive power while reducing unnecessary complexity in the feature space. The small differences across the stages confirm that the model is both highly stable and generalizable, capable of accurately predicting house prices across a wide range of property types and price segments.

Moreover, the residual analysis revealed that prediction errors were symmetrically distributed around zero, with no significant patterns of bias. The feature importance analysis further confirmed that key real estate factors such as total living area, overall quality, above-ground living area, and total bathrooms were the primary contributors driving the model's predictive success. Collectively, these findings validate that the combination of extensive data preprocessing, careful feature engineering, systematic hyperparameter tuning, and feature selection successfully produced a highly interpretable and effective machine learning model for property price estimation.

4.5 Web Application Deployment Results and Evaluation

4.5.1 Model Export and Backend Deployment

Following the successful development and training of the XGBoost regression model, the trained model was exported from Google Colab using Python's joblib library. The model file was saved in pickle format as xgb_model.pkl, allowing it to be seamlessly loaded later for deployment purposes. This process of model serialization ensures that the entire trained model, including its learned parameters and structure, is preserved and can be reused without the need for retraining.

For deployment, the Flask web framework was utilized as the backend platform to create a lightweight and flexible web server capable of hosting the machine learning

model. Flask allows easy integration of Python-based models with web interfaces, enabling users to interact with the trained model via a browser-based user interface. The backend code was developed entirely within Visual Studio Code (VS Code) on a local machine, ensuring full control over the deployment environment and enabling local testing and debugging during the development phase.

4.5.2 Web Interface Design

The web application interface was designed using standard HTML, CSS, and Flask templating to create a simple yet functional user interface that allows users to input house features and receive instant price predictions. As shown in Figure 4.8, the main webpage includes input fields for each of the selected features used by the trained model, such as overall quality, living area, garage capacity, total basement area, first floor area, year built, number of bathrooms, total rooms, number of fireplaces, and finished basement area.

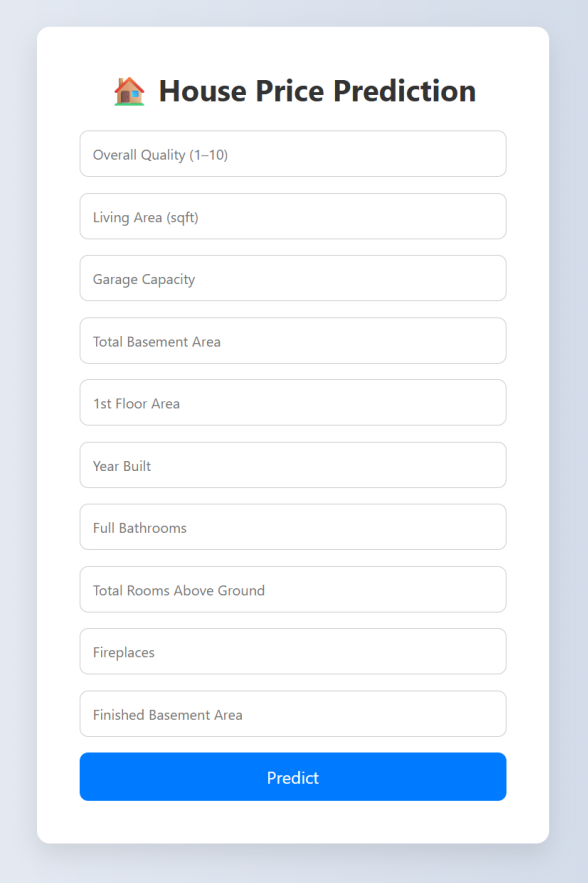


Figure 4.13 Webpage Interface

The web interface collects these input values from the user and submits them to the Flask backend server via HTTP POST requests. The server then processes the inputs, loads the previously saved xgb_model.pkl model, performs the prediction, and returns the predicted house price to the user.

The webpage's design was customized using a separate CSS file (style.css) to improve visual appearance, applying a clean and modern layout with a soft color palette, rounded form containers, and clear typography for ease of use. This interface ensures that even users with no technical background can easily interact with the system.

4.5.3 System Output and Functionality Demonstration

Upon entering the required property details and submitting the form, the web application returns the estimated house price predicted by the machine learning model. As shown in **Figure 4.9**, once the prediction is complete, the output is displayed directly on the webpage with both USD and converted MYR currency values to enhance user readability.

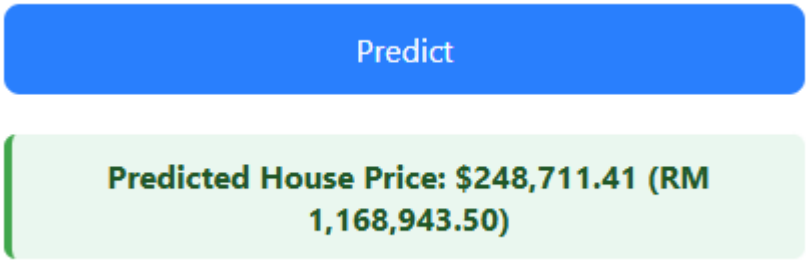


Figure 4.14 System Output

The deployment was successfully tested in a local environment, confirming that the integration between the trained model and the web interface functions correctly. Predictions are generated in real-time, with the system processing input features and returning the predicted price within seconds.

4.5.4 System Demonstration Summary

The successful deployment of the house price prediction model into a functional web application demonstrates the full applicability of machine learning models for real-world use. Users are able to interact with the model through an intuitive web-based interface, entering property features and receiving instant, data-driven price estimates.

The system provides significant benefits by allowing prospective homeowners, property agents, or financial institutions to quickly assess estimated property values based on actual property characteristics. Although this initial version was deployed locally, the

same system architecture could be extended and hosted on cloud platforms to enable remote access, scalability, and multi-user functionality.

While the current deployment performs well for single-user local testing, future improvements could involve connecting the system to external databases, automating data collection for real-time market updates, and extending the model to include additional economic indicators such as interest rates or inflation for even more comprehensive property valuation assessments.

4.6 Comparison with previous work

To evaluate the significance of the results obtained in this study, a comparative analysis was performed against prior research works discussed in the literature review of this project. Comparing results with previous studies allows us to assess the relative improvements achieved, validate the effectiveness of the adopted methodology, and highlight the contributions made by this research.

In previous studies, various machine learning models were applied for house price prediction, including Random Forest, Decision Trees, and Gradient Boosting models. For example, Ouyang (2023) employed a Random Forest model and reported an R^2 score of 0.69, indicating moderate predictive capability but limited ability to fully capture non-linear relationships. Similarly, Varma (2024) applied a Decision Tree model and achieved an R^2 of approximately 0.70, though Decision Trees are often prone to overfitting, particularly on complex datasets with many features. Both of these models, while straightforward and interpretable, demonstrated limited accuracy in explaining price variance in real-world property markets.

More recent studies such as Chuhan (2024) and Kumar (2024) explored more advanced ensemble models, including Gradient Boosting and XGBoost algorithms. Chuhan (2024), using XGBoost, achieved an R^2 score of approximately 0.90, while Kumar (2024), applying Gradient Boosting, reported a slightly higher R^2 of approximately 0.91. These studies demonstrated that ensemble boosting algorithms offer substantial improvements over basic tree-based methods by effectively capturing complex non-linear interactions and improving generalization.

The model developed in this current study achieved a final R^2 score of 0.9117, which is highly competitive and slightly exceeds some of the prior works reviewed. This confirms that the integration of comprehensive preprocessing, extensive feature engineering, systematic hyperparameter tuning, and proper feature selection contributed positively to model performance. Unlike earlier studies that primarily focused on raw datasets with limited preprocessing, this study introduced new engineered features such as TotalSF and TotalBath, which better captured combined effects of multiple physical property characteristics, leading to richer model inputs.

Additionally, while many previous works lacked practical deployment components, this project extended its contributions beyond model development by successfully integrating the trained model into a fully functional Flask-based web application, demonstrating its practical usability for end users in a simulated real-world environment.

Another important distinction in this work is the rigorous evaluation across multiple stages: starting from initial model training, through hyperparameter optimization, followed by feature selection refinement, and finally deployment testing. This systematic evaluation ensures that the reported results are stable, robust, and not overly dependent on any single configuration, further strengthening confidence in the reliability of the

model. The summary of R^2 scores from prior studies compared to this research is presented in Table 4.1.

Study	Model Used	R^2 Score	Notes
Ouyang (2023)	Random Forest	0.69	Lower Accuracy
Varma (2024)	Decision Tree	0.70	Overfitting Prone
Chuhan (2024)	XGBoost	0.90	Strong Performance
Kumar (2024)	Gradient Boosting	0.91	High Accuracy
Current Study	XGBoost + Feature Engineering + Deployment	0.9117	High Stability

Table 4.1 Summary of R^2 scores

In summary, when compared to prior research, this study demonstrates not only high predictive accuracy but also improved model robustness, interpretability, and practical deploying ability. The findings validate that a carefully designed end-to-end machine learning pipeline incorporating detailed data preparation, optimized model training, and real-world deployment can achieve state-of-the-art results in property price prediction.

4.7 Tools and Modern software used

Throughout the development of this project, several modern tools and software platforms were utilized to support data processing, model development, analysis, and deployment. Google Colab was employed as the primary environment for data preprocessing, feature engineering, model training, and hyperparameter tuning, benefiting from its integrated Python environment and scalable computational resources.

For machine learning model development, the XGBoost library was utilized alongside essential Python packages such as pandas for data manipulation, scikit-learn for

preprocessing and evaluation metrics, and matplotlib and seaborn for data visualization and analysis. The model serialization was performed using Python's joblib library to export the trained model for deployment.

The web application deployment was carried out using the Flask web framework, developed entirely within Visual Studio Code (VS Code). HTML, CSS, and Flask's Jinja2 templating system were used to build the user interface, while Python was used for backend integration with the machine learning model. Local deployment and testing were performed on a Windows-based system.

4.8 Limitations and Challenges

While the developed house price prediction model achieved high accuracy and stability, certain limitations were encountered during the project. One key challenge was the limited size and diversity of the available dataset, which may restrict the model's ability to generalize to different geographical regions or property markets with distinct characteristics not represented in the training data.

Another limitation arises from the exclusion of certain external economic factors, such as interest rates, inflation, or neighborhood-level market dynamics, which can also significantly influence property prices. The current model relies solely on structural and property-specific features without considering broader macroeconomic indicators.

Additionally, while the model was successfully deployed on a local Flask-based web application, deployment was limited to a controlled environment without live cloud hosting or integration with real-time data sources. Future enhancements could involve by deploying the system (webpage) on cloud platforms, connecting to live market data feeds,

and extending the model to handle dynamic market fluctuations for improved practical utility.

Finally, despite achieving strong predictive performance, all machine learning models inherently carry some degree of prediction uncertainty, especially for unique or highly customized properties that deviate from typical market norms. Continuous data updating and model retraining would be essential to maintain long-term predictive reliability.

4.9 Chapter summary

This chapter presented the results and detailed evaluation of the house price prediction model developed using housing data. The project involved extensive data preprocessing, including missing value imputation, ordinal encoding, and feature engineering, which produced a clean and highly informative dataset. The XGBoost algorithm was employed as the predictive model, with hyperparameter tuning and feature selection contributing to its improved stability and accuracy.

The final model achieved strong predictive performance with an R^2 score of 0.9117. Visual evaluations, including residual analysis and feature importance rankings, confirmed that the model captured key property attributes such as total living space, overall quality, and total bathrooms as primary determinants of house price. The consistent and unbiased error distributions demonstrated the model's robustness and generalization capability.

In addition to model development, the project successfully integrated the trained model into a Flask-based web application, allowing users to interactively input property features and receive instant price predictions. Comparative analysis with prior studies highlighted the superiority of the developed approach, both in terms of accuracy and practical deployment readiness.

CHAPTER 5 : CONCLUSION

5.1 Introduction

This project successfully developed a house price prediction system using Extreme Gradient Boosting (XGBoost), addressing the limitations of conventional valuation approaches by modeling complex, non-linear relationships within housing data. Through extensive data preprocessing, including handling missing values, feature engineering, and encoding, the dataset was optimized for machine learning. Important composite features such as Total Living Area, Overall Quality, and Total Bathrooms significantly enhanced the model's predictive capabilities.

After hyperparameter tuning using GridSearchCV, the XGBoost model achieved a strong R^2 score of 0.9117, explaining over 91% of price variance in the dataset. The model was integrated into a user-friendly web-based platform developed with Flask, HTML, CSS, and Python, allowing users to input property details and receive real-time price predictions. This practical deployment demonstrated the system’s potential for real-world application.

Despite its success, the study recognizes certain limitations, including the restricted regional scope of the dataset and the exclusion of dynamic economic factors such as interest rates and inflation. Nevertheless, the research contributes significantly to real estate analytics by delivering an accurate, interpretable, and deployable machine learning solution that can benefit homeowners, investors, and policymakers.

5.2 Future Recommendation

While the developed system has demonstrated strong predictive performance and practical usability, several enhancements can be pursued to further improve its robustness. Incorporating macroeconomic indicators such as inflation rates, interest rates, employment levels, and housing policies would allow the model to dynamically adapt to evolving market conditions, enhancing its long-term forecasting capabilities.

Expanding the dataset to include more diverse geographic regions and property types would improve the model's generalization across different markets, making it applicable to national and international real estate contexts. Deploying the system on scalable cloud platforms like Google Cloud or AWS would also enable multi-user access, real-time data integration, and improved scalability for commercial applications.

Additional future improvements include integrating explainable AI techniques such as SHAP values to enhance model transparency and user trust. Establishing dynamic retraining pipelines would allow continuous model updates as new data becomes available, ensuring sustained accuracy. Furthermore, incorporating new data modalities like satellite imagery, property photos, and GIS data could provide deeper insights into property quality and neighbourhood desirability, strengthening the model's predictive capabilities for real-world applications.

REFERENCES

[1] Yeee, G. F., Sufahani, S. F., Wahab, M. H. A., & Idrus, S. Z. S. (2021). Factors influence the housing price in Kuala Lumpur by using AHP. *Journal of Physics: Conference Series*, *1793*(1). https://doi.org/10.1088/1742-6596/1793/1/012027

[2] Chen, T., & Guestrin, C. (n.d.). *XGBoost: A Scalable Tree Boosting System*. https://doi.org/10.1145/2939672.2939785

[3] Nagassou, M., Mwangi, R. W., Nyarige, E., Nagassou, M., Mwangi, R. W., & Nyarige, E. (2023). A Hybrid Ensemble Learning Approach Utilizing Light Gradient Boosting Machine and Category Boosting Model for Lifestyle-Based Prediction of Type-II Diabetes Mellitus. *Journal of Data Analysis and Information Processing*, *11*(4), 480–511. https://doi.org/10.4236/JDAIP.2023.114025

[4] Rane, N. L., Mallick, S. K., Kaya, Ö., & Rane, J. (2024). Applications of machine learning in healthcare, finance, agriculture, retail, manufacturing, energy, and transportation: A review. *Applied Machine Learning and Deep Learning: Architectures and Techniques*. https://doi.org/10.70593/978-81-981271-4-3_6

[5] Mathauer, I., & Oranje, M. (2023). Machine learning in health financing: benefits, risks and regulatory needs. *Bulletin of the World Health Organization*, *102*(3), 216. https://doi.org/10.2471/BLT.23.290333

[6] Huang, C., Clayton, E. A., Matyunina, L. v., McDonald, L. D. E., Benigno, B. B., Vannberg, F., & McDonald, J. F. (2018). Machine learning predicts individual cancer patient responses to therapeutic drugs with high accuracy. *Scientific Reports*, *8*(1), 1–8. https://doi.org/10.1038/S41598-018-34753-5;TECHMETA=38,39,45,61,91;SUBJMETA=114,2785,631,67,69;KWRD=CANCER+GENOMICS,GENOME+INFORMATICS

[7] Hernandez Aros, L., Bustamante Molano, L. X., Gutierrez-Portela, F., Moreno Hernandez, J. J., & Rodríguez Barrero, M. S. (2024). Financial fraud detection through the application of machine learning techniques: a literature review. *Humanities and Social Sciences Communications 2024 11:1*, *11*(1), 1–22. https://doi.org/10.1057/s41599-024-03606-0

[8] Xia, Z., Sun, A., Xu, J., Peng, Y., Ma, R., & Cheng, M. (2022). *Contemporary Recommendation Systems on Big Data and Their Applications: A Survey*. https://doi.org/10.1109/access.2024.3517492

[9] Hong, T., Wang, Z., Luo, X., & Zhang, W. (2020). State-of-the-art on research and applications of machine learning in the building life cycle. *Energy and Buildings*, *212*. https://doi.org/10.1016/j.enbuild.2020.109831

[10] Zhang, S., Chen, W., Xu, J., & Xie, T. (2024). Use of interpretable machine learning approaches for quantificationally understanding the performance of steel fiber-reinforced recycled aggregate concrete: From the perspective of compressive strength and splitting tensile strength. *Engineering Applications of Artificial Intelligence*, *137*, 109170. https://doi.org/10.1016/J.ENGAPPAI.2024.109170

[11] Mitchell, T. M. (2010). CHAPTER 1 GENERATIVE AND DISCRIMINATIVE CLASSIFIERS : NAIVE BAYES AND LOGISTIC REGRESSION Learning Classifiers based on Bayes Rule. *Machine Learning*, *1*(Pt 1-2), 1–17. https://doi.org/10.1093/bioinformatics/btq112

[12] Mexis, K., Xenios, S., & Kokosis, A. (2023). On the systematic development of large-scale kinetics using stability criteria and high-throughput analysis of curated dynamics from genome-scale models. *Computer Aided Chemical Engineering*, *52*, 2729–2734. https://doi.org/10.1016/B978-0-443-15274-0.50434-0

[13] Gaikwad, D. P., & Thool, R. C. (2015). Intrusion detection system using Bagging with Partial Decision Tree base classifier. *Procedia Computer Science*, *49*(1), 92–98. https://doi.org/10.1016/j.procs.2015.04.231

[14] Zhou, Z. H. (2021). Machine Learning. *Machine Learning*, 1–458. https://doi.org/10.1007/978-981-15-1967-3

[15] Rijcken, E., Zervanou, K., Mosteiro, P., Scheepers, F., Spruit, M., & Kaymak, U. (2025). Machine learning vs. rule-based methods for document classification of electronic health records within mental health care—A systematic literature review. *Natural Language Processing Journal*, *10*, 100129. https://doi.org/10.1016/J.NLP.2025.100129

[16] Li, Z., Du, X., Xu, A., Wu, T., & Cao, Y. (2025). Explaining tree ensembles through single decision trees. *Information Fusion*, *123*. https://doi.org/10.1016/J.INFFUS.2025.103244

[17] Sharma, H., Harsora, H., & Ogunleye, B. (2024). An Optimal House Price Prediction Algorithm: XGBoost. *Analytics 2024, Vol. 3, Pages 30-45*, *3*(1), 30–45. https://doi.org/10.3390/ANALYTICS3010003

[18] Banjongkan, A., Pongsena, W., Kerdprasop, N., & Kerdprasop, K. (2021). A study of job failure prediction at job submit-state and job start-state in high-performance computing system: Using decision tree algorithms. *Journal of Advances in Information Technology*, *12*(2), 84–92. https://doi.org/10.12720/JAIT.12.2.84-92

[19] Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. E. (2014). *Data Preprocessing for Supervised Learning*. https://www.researchgate.net/publication/228084519

[20] Kam, K. J., Lim, A. S. H., Al-Obaidi, K. M., & Lim, T. S. (2018). Evaluating Housing Needs and Preferences of Generation Y in Malaysia. *Planning Practice and Research*, *33*(2), 172–185. https://doi.org/10.1080/02697459.2018.1427413

[21] Li, A. (2025). House Price Prediction Using Machine Learning: Analysis of XGBoost and Gradient Boosting Models. *Applied and Computational Engineering*, *155*(1), 117–124. https://doi.org/10.54254/2755-2721/2025.GL23414

[22] Zheng, R. (2025). Bayesian Optimization of Lasso and XGBoost Models for Comparative Analysis in Housing Price Prediction. *ITM Web of Conferences*, *73*, 03005. https://doi.org/10.1051/ITMCONF/20257303005

[23] Limbong, H., Lubis, M. A., & Mhd. Furqan. (2025). House Price Prediction Analysis Using Linear Regression and Random Forest Algorithms. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, *4*(3), 1928–1933. https://doi.org/10.59934/JAIEA.V4I3.1047

[24] Mathotaarachchi, K. V., Hasan, R., & Mahmood, S. (2024). Advanced Machine Learning Techniques for Predictive Modeling of Property Prices. *Information 2024, Vol. 15, Page 295*, *15*(6), 295. https://doi.org/10.3390/INFO15060295

[25] Carneiro, T., da Nobrega, R. V. M., Nepomuceno, T., Bian, G. bin, de Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, *6*, 61677–61685. https://doi.org/10.1109/ACCESS.2018.2874767,

[26] Rawool, A. G., Rogye, D. v, Rane, S. G., Vinayk, D. R., & Bharadi, A. (2021). House Price Prediction Using Machine Learning. *IRE Journals |*, *4*.

APPENDIX A
GANTT CHART

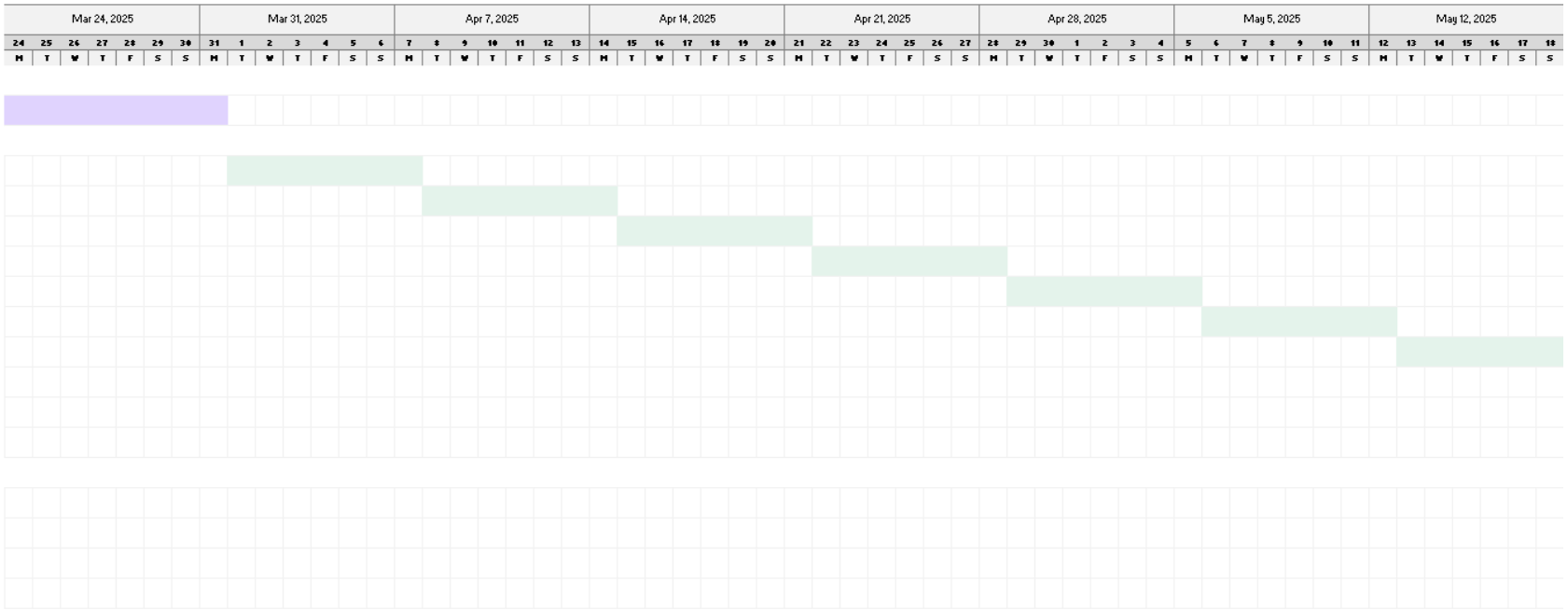
FINAL YEAR PROJECT II

TITLE : PREDICTION OF HOUSE PRICE WITH EXTREME GRADIENT BOOSTING.

Project start: Mon, 3/24/2025

Display week: 1

TASK	MILESTONES	PROGRES!	START	END
Project Planning and Design				
Dataset Finalization & Preprocessing Review	Yes	100%	24/03/2025	3/31/25
Execution				
Feature Engineering & Variable Selection	Yes	100%	01/04/2025	4/7/25
Model Development - Initial Training (Baseline Model)	Yes	100%	08/04/2025	4/14/25
Hyperparameter Tuning & Model Optimization	Yes	100%	15/04/2025	4/21/25
Model Evaluation & Performance Validation	Yes	100%	22/04/2025	4/28/25
Feature Importance Analysis & Model Interpretation	Yes	100%	29/04/2025	5/5/25
Backend Development (Flask API + Model Integration)	Yes	100%	06/05/2025	5/12/25
Frontend Development (Web Interface Design)	Yes	100%	13/05/2025	5/26/25
Full System Integration (Backend + Frontend Connection)	Yes	100%	27/05/2025	6/2/25
System Testing, Debugging & Improvement	Yes	100%	03/06/2025	6/9/25
Deployment Testing (Cloud or Local Deployment Trial)	Yes	100%	10/06/2025	6/16/25
Evaluation				
Writing Results, Discussions & Analysis Chapters	Yes	100%	17/06/2025	6/23/25
Draft Full Report Writing	Yes	100%	17/06/2025	6/23/25
Final Report Editing, Formatting & Submission	Yes	100%	22/06/2025	6/27/25
Preparation of viva	Yes	100%	23/06/2025	7/14/25



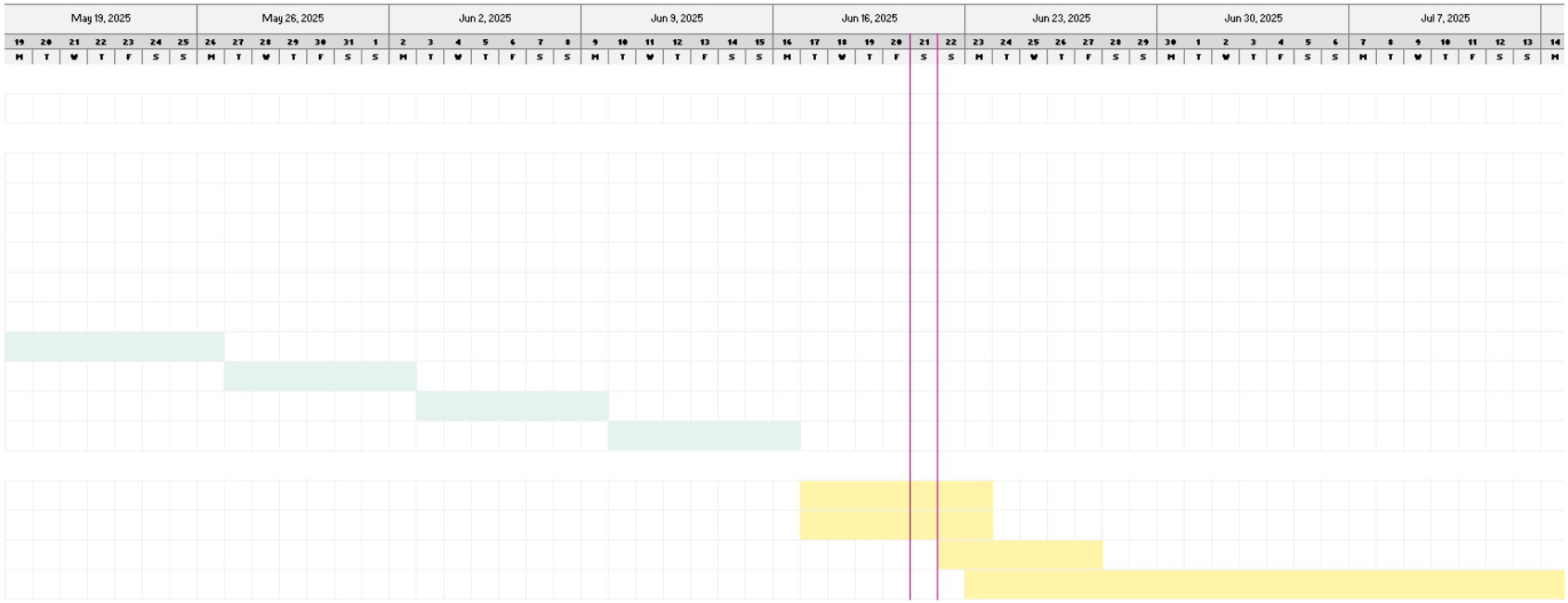
FINAL YEAR PROJECT II

TITLE : PREDICTION OF HOUSE PRICE WITH EXTREME GRADIENT BOOSTING.

Project start: Mon, 3/24/2025

Display week: 9

TASK	MILESTONES	PROGRES!	START	END
Project Planning and Design				
Dataset Finalization & Preprocessing Review	Yes	100%	24/03/2025	3/31/25
Execution				
Feature Engineering & Variable Selection	Yes	100%	01/04/2025	4/7/25
Model Development - Initial Training (Baseline Model)	Yes	100%	08/04/2025	4/14/25
Hyperparameter Tuning & Model Optimization	Yes	100%	15/04/2025	4/21/25
Model Evaluation & Performance Validation	Yes	100%	22/04/2025	4/28/25
Feature Importance Analysis & Model Interpretation	Yes	100%	29/04/2025	5/5/25
Backend Development (Flask API + Model Integration)	Yes	100%	06/05/2025	5/12/25
Frontend Development (Web Interface Design)	Yes	100%	13/05/2025	5/26/25
Full System Integration (Backend + Frontend Connection)	Yes	100%	27/05/2025	6/2/25
System Testing, Debugging & Improvement	Yes	100%	03/06/2025	6/9/25
Deployment Testing (Cloud or Local Deployment Trial)	Yes	100%	10/06/2025	6/16/25
Evaluation				
Writing Results, Discussions & Analysis Chapters	Yes	100%	17/06/2025	6/23/25
Draft Full Report Writing	Yes	100%	17/06/2025	6/23/25
Final Report Editing, Formatting & Submission	Yes	100%	22/06/2025	6/27/25
Preparation of viva	Yes	100%	23/06/2025	7/14/25



APPENDIX B
TURNITIN REPORT

CHAPTER 1 : INTRODUCTION

Document Viewer

Turnitin Originality Report

Processed on: 21-Jun-2025 23:18 +08
ID: 2703404624
Word Count: 1700
Submitted: 1

Chp 1.docx By Dinesh Manivannan

Similarity Index	Similarity by Source	
8%	Internet Sources:	5%
	Publications:	1%
	Student Papers:	4%

include quoted | include bibliography | excluding matches < 5 words | mode: quickview (classic) report | print | download

2% match (Internet from 10-May-2025)
https://americaspg.com/article/download/3401

1% match (student papers from 06-Jul-2022)
Submitted to Universiti Malaysia Perlis on 2022-07-06

1% match (student papers from 16-Nov-2024)
Submitted to Queensland University of Technology on 2024-11-16

1% match (student papers from 20-Nov-2023)
Submitted to University of North Texas on 2023-11-20

1% match (student papers from 04-Apr-2025)
Submitted to University of Sunderland on 2025-04-04

1% match (Internet from 03-Nov-2024)
https://lopez-christian.github.io/2020-06-15-liver-disease-machine-learning-project/?ref=https%3A%2F%2Fgithubhelp.com

1% match (Internet from 08-Jun-2023)
https://www.researchgate.net/publication/336414552_A_Pattern_Recognition_Method_for_Partial_Discharge_Detection_on_Insulated_Overhead_Conductors

1% match (Internet from 02-May-2025)
https://ijirt.org/publishedpaper/IJIRT175898_PAPER.pdf

Turnitin Originality Report

Processed on: 12-Jan-2025 22:09 +08
ID: 2562791482
Word Count: 10478
Submitted: 2

LRPR2 By Dinesh Manivannan

Similarity Index	Similarity by Source	
23%	Internet Sources:	16%
	Publications:	14%
	Student Papers:	13%

include quoted | include bibliography | excluding matches < 5 words | mode: quickview (classic) report | print | download

1% match (Internet from 13-Nov-2024)
http://fastercapital.com

1% match (Internet from 03-Jan-2024)
https://fastercapital.com/keyword/r-squared-value.html

1% match (Internet from 25-Feb-2024)
https://fastercapital.com/keyword/predicted-actual-values.html

1% match (Internet from 20-Aug-2024)
https://www.mdpi.com/2571-9394/6/3/28

1% match (student papers from 14-Sep-2023)
Submitted to University of Southampton on 2023-09-14

1% match (Internet from 07-May-2024)
https://drpress.org/ojs/index.php/HSET/article/download/18525/18063

1% match (student papers from 19-Aug-2024)
Submitted to Kingston University on 2024-08-19

CHAPTER 3: METHODOLOGY

Document Viewer

Turnitin Originality Report

Processed on: 21-Jun-2025 23:20 +08
ID: 2703405264
Word Count: 2486
Submitted: 1

METHODOLOGY fyp2.docx By Dinesh Manivannan

Similarity Index	Similarity by Source	
16%	Internet Sources:	11%
	Publications:	7%
	Student Papers:	12%

include quoted | include bibliography | excluding matches < 5 words | mode: quickview (classic) report | print | download

5% match (Internet from 06-Jan-2023)
https://thedeveloperblog.com/data/data-preprocessing-machine-learning

2% match (student papers from 27-Mar-2024)
Submitted to Liverpool John Moores University on 2024-03-27

2% match (Internet from 05-Nov-2024)
https://machinelearningadda.blogspot.com/2022/09/4-data-preprocessing-in-machine.html

1% match (student papers from 14-Feb-2025)
Submitted to University of South Wales - Pontypridd and Cardiff on 2025-02-14

1% match (publications)
R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced Artificial Intelligence (ICAAAI-2024)", CRC Press, 2025

1% match (Internet from 14-May-2025)
https://codingisland.net/machine-learning/supervised-vs-unsupervised-learning/

1% match (Internet from 21-May-2024)
https://ir.mallareddyecw.com/id/eprint/474/1/B21.pdf

1% match (Internet from 05-May-2025)
https://www.ijraset.com/research-paper/medicine-recommendation-system-using-ml

1% match (student papers from 18-Jan-2024)
Submitted to Brunel University on 2024-01-18

Turnitin Originality Report

Processed on: 21-Jun-2025 19:59 +08
ID: 2703349005
Word Count: 4680
Submitted: 1

chp 4.docx By Dinesh Manivannan

Document Viewer

Similarity Index	Similarity by Source	
3%	Internet Sources:	1%
	Publications:	3%
	Student Papers:	2%

include quoted | include bibliography | excluding matches < 5 words | mode: quickview (classic) report | print | download

1% match (publications)
Poonam Nandal, Mamta Dahiya, Meeta Singh, Arvind Dagur, Brijesh Kumar. "Progressive Computational Intelligence, Information Technology and Networking", CRC Press, 2025

1% match (Li Xue, Runyu Jing, Nanya Zhong, Xiaoyu Nie, Yitong Du, Jiesi Luo, Kama Huang. "Machine learning to guide the use of plasma technology for antibiotic degradation", Journal of Hazardous Materials, 2024)
Li Xue, Runyu Jing, Nanya Zhong, Xiaoyu Nie, Yitong Du, Jiesi Luo, Kama Huang. "Machine learning to guide the use of plasma technology for antibiotic degradation", Journal of Hazardous Materials, 2024

<1% match (student papers from 09-Jun-2024)
Submitted to AUT University on 2024-06-09

<1% match (publications)
Dinesh Goyal, Bhanu Pratap, Sandeep Gupta, Saurabh Raj, Rekha Rani Agrawal, Indra Kishor. "Recent Advances in Sciences, Engineering, Information Technology & Management - Proceedings of the 6th International Conference "Convergence2024" Recent Advances in Sciences, Engineering, Information Technology & Management, April 24–25, 2024, Jaipur, India", CRC Press, 2025

<1% match (student papers from 23-Apr-2025)
Submitted to UCL on 2025-04-23

<1% match (student papers from 06-May-2025)
Submitted to Arab Open University on 2025-05-06

<1% match (student papers from 02-Aug-2023)
Submitted to KCA University on 2023-08-02

<1% match (Internet from 25-May-2025)

CHAPTER 5: CONCLUSION

CONCLUSION Introduction This research has successfully fulfilled its stated objectives of developing a robust house price prediction system using advanced machine learning techniques, specifically Extreme Gradient Boosting (XGBoost). The project addressed the identified problem of limited accuracy in conventional property valuation methods by constructing a predictive model capable of capturing complex, non-linear relationships among various real estate features. Comprehensive data preprocessing, including missing value handling, feature engineering, and encoding, ensured the dataset was highly suitable for machine learning training. The integration of important composite features such as Total Living Area (TotalSF), Overall Quality (OverallQual), and Total Bathrooms (TotalBath) further improved the model's ability to understand real-world housing market dynamics. Following extensive hyperparameter optimization using GridSearchCV, the final XGBoost model achieved an impressive R^2 score of 0.9117, demonstrating its high predictive accuracy. The model successfully explains over 91% of the variance in house prices based on available features, confirming its reliability and generalization capability. The deployment of this predictive model into a functioning web-based interface further demonstrates the practical applicability of this research. Using Flask, HTML, CSS, and Python backend development tools, a user-friendly web application was created, enabling real-time predictions based on user input. Despite the positive outcomes, this study acknowledges several limitations. The dataset used was limited to specific regional data from Kaggle and lacked dynamic economic variables such as interest rates or inflation. The model deployment remained within a controlled local environment without full-scale cloud deployment or integration with real-time data sources. In summary, this project makes a significant contribution to the field of real estate analytics by delivering a highly accurate, practical, and deployable house price prediction system. It demonstrates that tree-based ensemble models, combined with effective data preprocessing, careful feature selection, and advanced tuning techniques, can significantly outperform traditional valuation approaches. The developed system provides a valuable tool for homeowners, buyers, property investors, and policymakers seeking reliable, data-driven property valuations. Future Recommendation While the developed system demonstrates strong predictive accuracy and practical usability, several opportunities for future enhancement are identified to further improve its robustness and applicability. One important direction is the integration of macroeconomic indicators into the model. Incorporating real-time economic data such as inflation rates, mortgage interest rates, housing policies, employment levels, and other financial factors would allow the model to adapt dynamically to changing market conditions and improve its long-term forecasting ability. Another recommended improvement is to expand the dataset to cover a broader range of geographic regions and diverse property types. Including more locations with varying market characteristics will enable the model to generalize better across different regions and ensure that the predictions remain relevant beyond the initially studied market. This would increase the model's applicability for both national and international real estate markets. Furthermore, deploying the system on scalable cloud platforms such as Google Cloud or AWS would allow for multi-user access, real-time operation, and continuous integration of new data sources. Cloud deployment would also enhance system scalability, security, and availability for potential real-world commercial applications. The incorporation of explainable artificial intelligence (XAI) techniques is another valuable recommendation. By implementing interpretability methods such as SHAP (SHapley Additive Explanations) values, the model's predictions can be made more transparent to end-users, offering clearer insights into how different features influence the predicted house prices. Additionally, establishing dynamic retraining pipelines would allow the model to automatically update itself as new data becomes available. This continuous learning mechanism would ensure that the system remains accurate over time, adapting to evolving market trends without the need for manual retraining. Lastly, future research may explore the incorporation of additional data modalities, such as image-based features derived from satellite imagery, street view data, or property interior photos. Combining such visual data with existing numerical features could further enhance the model's predictive performance by capturing aspects of property quality not fully reflected in tabular data. This could also provide valuable supplementary insights into neighborhood desirability and market perceptions. By addressing these future enhancements, the developed house price prediction system has the potential to evolve into a more powerful, adaptive, and widely applicable tool, capable of serving both academic research and practical real-world property valuation needs in a rapidly changing real estate market.

APPENDIX C

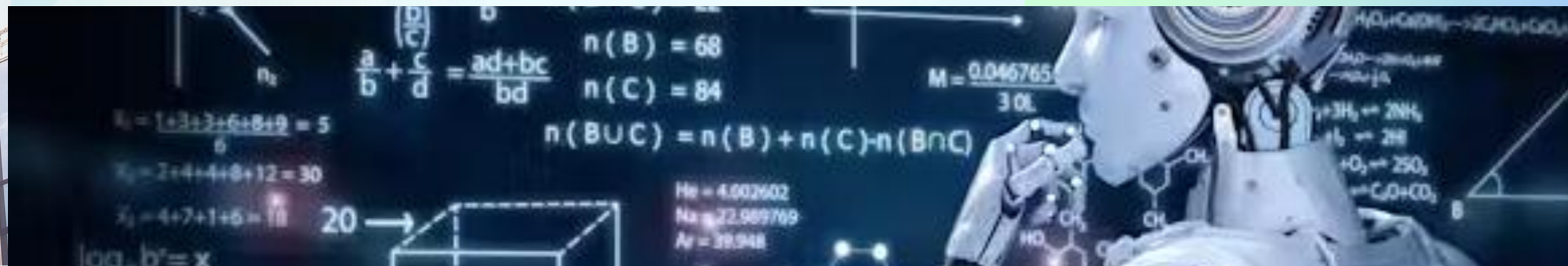
Feature Code	Actual Feature Name	Explanation
f0	TotalSF	Total square footage (basement + above ground)
f1	OverallQual	Overall material and finish quality
f2	YearBuilt	Year the house was built
f3	YearRemodAdd	Year of last remodelling
f4	ExterQual	Exterior quality
f5	BsmtQual	Basement quality
f6	BsmtCond	Basement condition
f7	BsmtFinSF1	Finished basement area (1st portion)
f8	GrLivArea	Above ground living area (main house living space)
f9	KitchenQual	Kitchen quality
f10	TotRmsAbvGrd	Total rooms above ground
f11	Fireplaces	Number of fireplaces
f12	FireplaceQu	Fireplace quality
f13	GarageYrBlt	Garage year built
f14	GarageFinish	Garage finish quality
f15	GarageCars	Garage capacity (number of cars)
f16	GarageArea	Garage area
f17	GarageQual	Garage quality
f18	TotalBath	Total number of bathrooms (engineered feature)
f19	LotArea	Lot area (total land area)
f20	LotFrontage	Frontage width

PREDICTION OF HOUSE PRICES WITH EXTREME GRADIENT BOOSTING.

PREDICTION OF HOUSE PRICES WITH EXTREME GRADIENT BOOSTING.

NAME: DINESH A/L MANIVANNAN (211021094)
SUPERVISED BY ASSOC. PROF. DR. ZAHEREEL ISHWAR ABDUL KHALIB

FYP: FYP2



Limitations

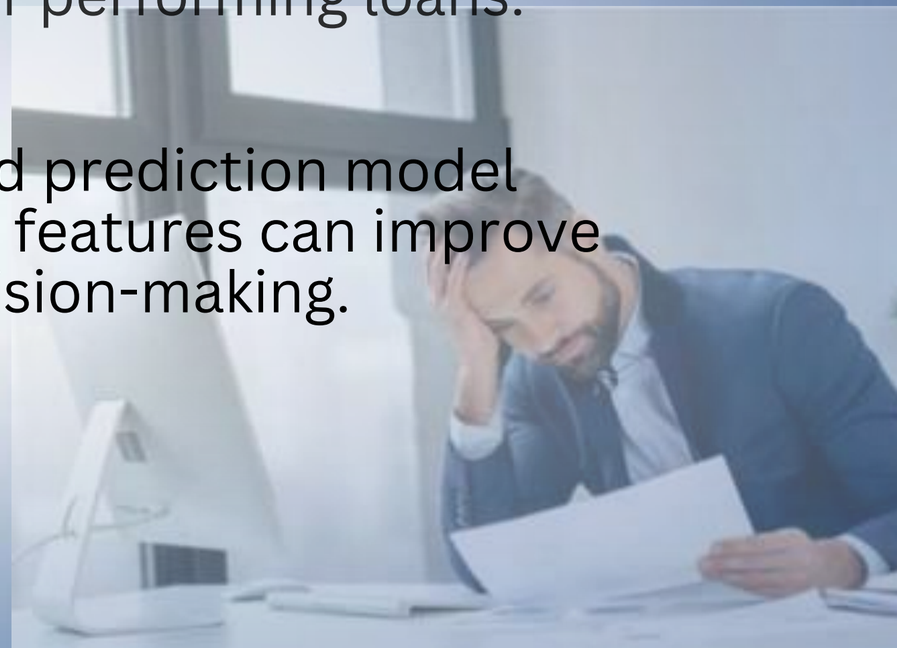
Problem Statement

Objectives

- Overvaluing and undervaluing can cause financial difficulties for buyers and market imbalances for developers.
- Important factors are often overlooked, leading to inaccurate predictions.
- Overpricing forces low- and middle-income families to take unsustainable loans, increasing household debt and non-performing loans.

SOLUTION??

A near-accurate ML-based prediction model incorporating overlooked features can improve pricing accuracy and decision-making.



Problem Statement

Objectives

Scope

- To design a tree-based machine learning model for predicting house prices using real world dataset.
- To develop a webpage that allows user to input feature details and get house price prediction.
- To evaluate the performance of model using standard metrics which and MAE, MSE, RMSE and R squared.

Objectives

Scope

Limitations

- The project aims for high accuracy and clarity by using algorithm such as Gradient Boosting.
- Assess performance using metrics such as R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE).
- Attempts to understand the significance of characteristics such as neighborhood quality and access to conveniences that determine property values.

Scope

Limitations

Problem Statement

- Model accuracy depends on data quality. Incomplete or outdated data may impact performance.
- Availability of relevant datasets with comprehensive features can be a constraint.
- Tree-based models may overfit and struggle with generalization.

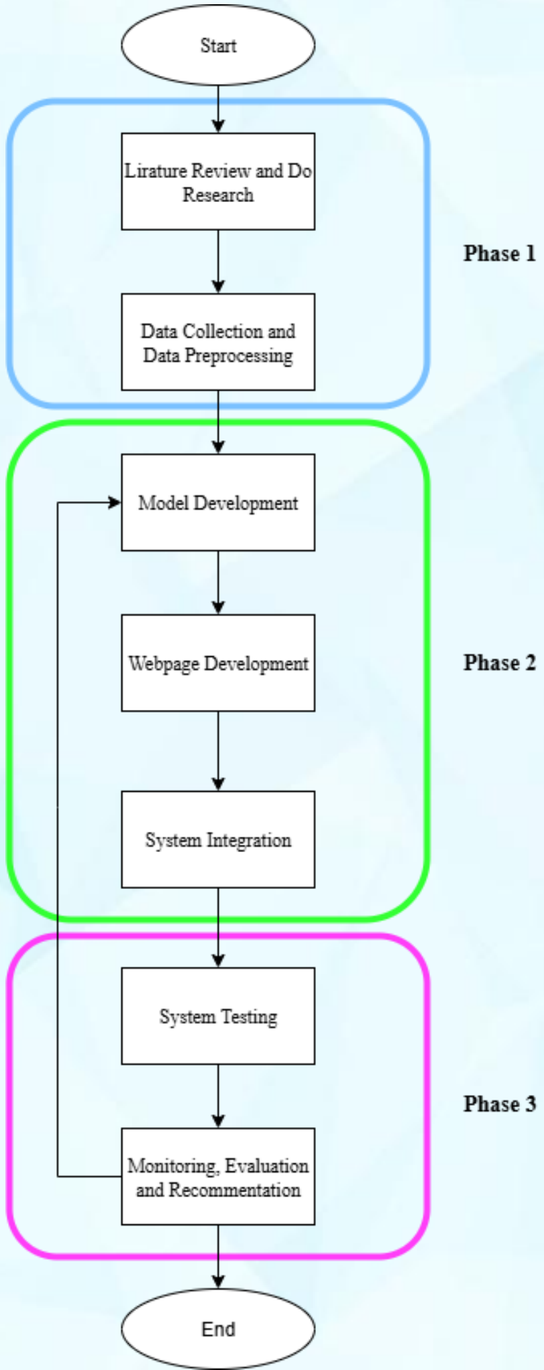
SOTA - PART A

Previous paper	Model	Attributes/Factors	Limitations / Gaps
Kalidass, J., et(2024). HOUSE PRICE PREDICTION USING MACHINE LEARNING.	Random Forest, Gradient Boosting	Property size, neighbourhood quality, proximity to schools and work centres	Dataset preprocessing and feature engineering not emphasized.
Li, Y. (2023). Analysis of Real Estate Predictions Based on Different Models.	Decision Tree, Extreme Gradient Boosting, Random Forest	Square footage, proximity to amenities, market trends, year of construction	The absence of temporal data integration leads to inaccurate forecasting when long-term trends shift.
Mao, M. (2024). A Comparative Study of Random Forest Regression for Predicting House Prices Using.	Random Forest Regression	Property type, neighbourhood characteristics, transport access, historical prices	The model uses static factors and lacks integration of dynamic factors like policy changes or new infrastructure projects.
Quang, T., Minh, N., Hy, D., & Bo, M. (2020). Housing Price Prediction via Improved Machine Learning Techniques.	Random Forest, Extreme Gradient Boosting	Proximity to business hubs, crime rates, public infrastructure, social factors	Improved techniques but lack cross-region adaptability; no detailed explainability analysis.
YAVUZ ÖZALP, A., & AKINCI, H. (2023). Comparison of tree-based machine learning algorithms in price prediction of residential real estate.	Decision Tree, Random Forest, Extra Trees, Gradient Boosting	Proximity to public transport, land area, crime rates, infrastructure	Overlapping features (e.g., public transport and infrastructure) may introduce multicollinearity, affecting prediction reliability.

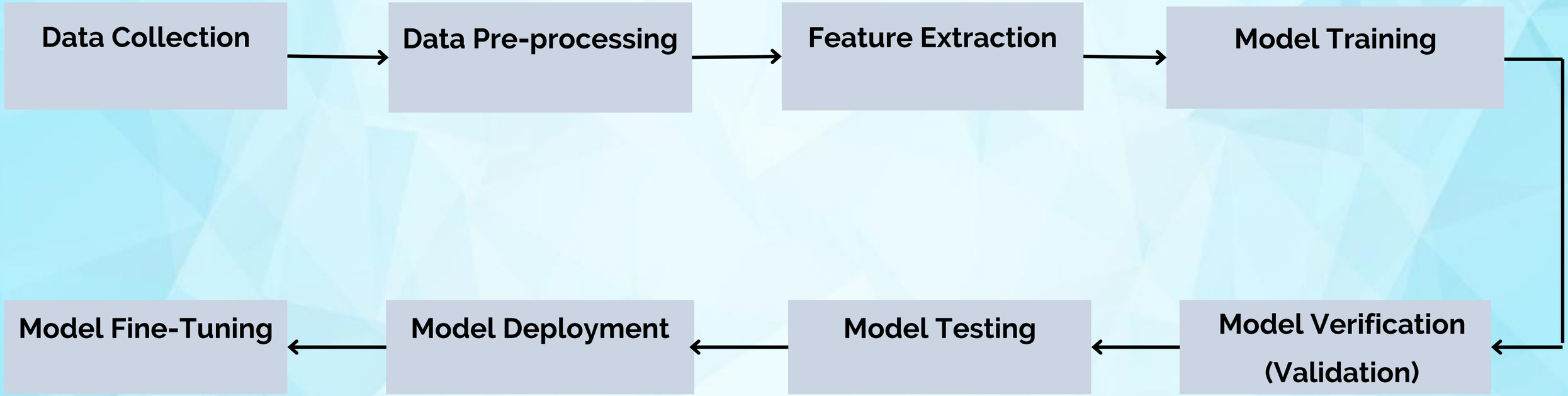
SOTA - PART B

Previous paper	Model	Attributes/Factors	Limitations / Gaps
Kumar, Bv., & Professor, A. (2020). House Price Prediction using Garadient Boost Regression Model	Gradient Boosting	Neighbourhood, lot size, historical pricing data, economic indicators	Performance limited to Gradient Boost regression; no comparative study with other tree based models.
Adetunji, A. B., Akande, O. N., Ajala, F. A., Oyewo, O., Akande, Y. F., & Oluwadara, G. (2022). House Price Prediction using Random Forest Machine Learning Technique	Random Forest Regression	Distance to city centre, property type, land size, social factors	Limited cross-validation techniques; no use of advanced ensemble methods for comparison.
Akash Dagar and Shreya Kapoor. (2020). A Comparative Study on House Price Prediction.	Multivariable Linear Regression, Decision Tree Regression, Random Forest Regression	Area, location, age of property, number of rooms	Lack of scalability for larger datasets; limited focus on hyperparameter optimization for tree-based models.
Chuhan, N. (2024). House price prediction based on different models of machine learning.	Linear Regression, Support Vector Machine (SVM), Random Forest regression, Extreme Gradient Boosting	Size, number of bedrooms and bathrooms, proximity to public transport	No detailed discussion on feature importance or interpretability of results.
Mohd, T., Masrom, S., & Johari, N. (2019). Machine learning housing price prediction in petaling jaya, Selangor, Malaysia.	Linear Regression, Decision Tree, Random Forest, Ridge and Lasso algorithms	Property type, land area, age, location	Study focused only on a specific geographical area (Petaling Jaya, Selangor), limiting wider applicability.

FLOWCHART


Start
Lirature Review and Do Research
Data Collection and Data Preprocessing
Phase 1
Model Development
Webpage Development
System Integration
Phase 2
System Testing
Monitoring, Evaluation and Recommentation
Phase 3
End

MACHINE LEARNING PIPELINE



DATA PRE-PROCESSING

Import Dataset

kaggle

Kaggle allows users to find datasets they want to use in building AI models, publish datasets.

Data Cleaning (minimal)

- Checked for nulls, removed irrelevant or redundant features
- Previewed structure and column types

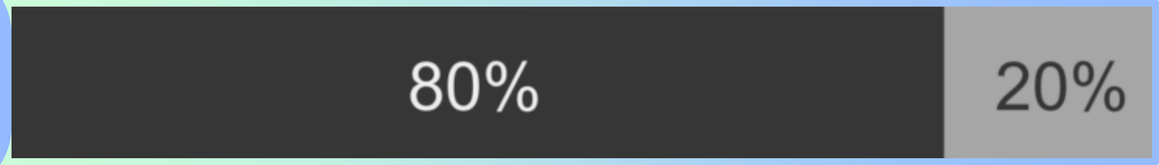
Handling missing values

- By calculating the mean.
- Replace the missing data with mean value of specific column.

Encoding Categorical Data

- Hot-one encoding
- Creating separate binary columns for each category.

Split Dataset



FEATURE EXTRACTION & ENGINEERING

Constructed new features

- TotalSF (Total square footage) = TotalBsmtSF + 1stFlrSF + 2ndFlrSF
- TotalBath = FullBath + 0.5 × HalfBath + BsmtFullBath + 0.5 × BsmtHalfBath

Combined related features

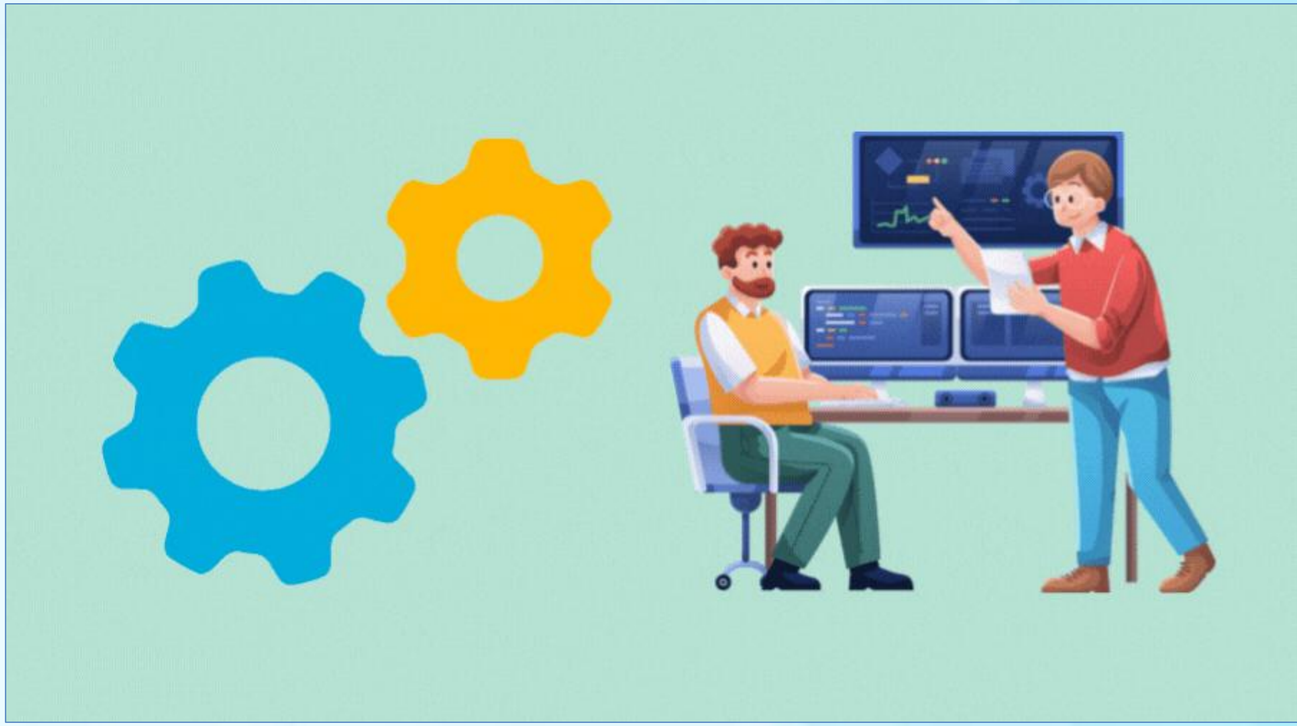
- YearRemodAdd and YearBuilt were used to derive property age
- GarageArea and GarageCars considered jointly to reflect garage utility

Removed irrelevant or redundant columns

- Dropped ID columns and features with near-zero variance or too many missing values

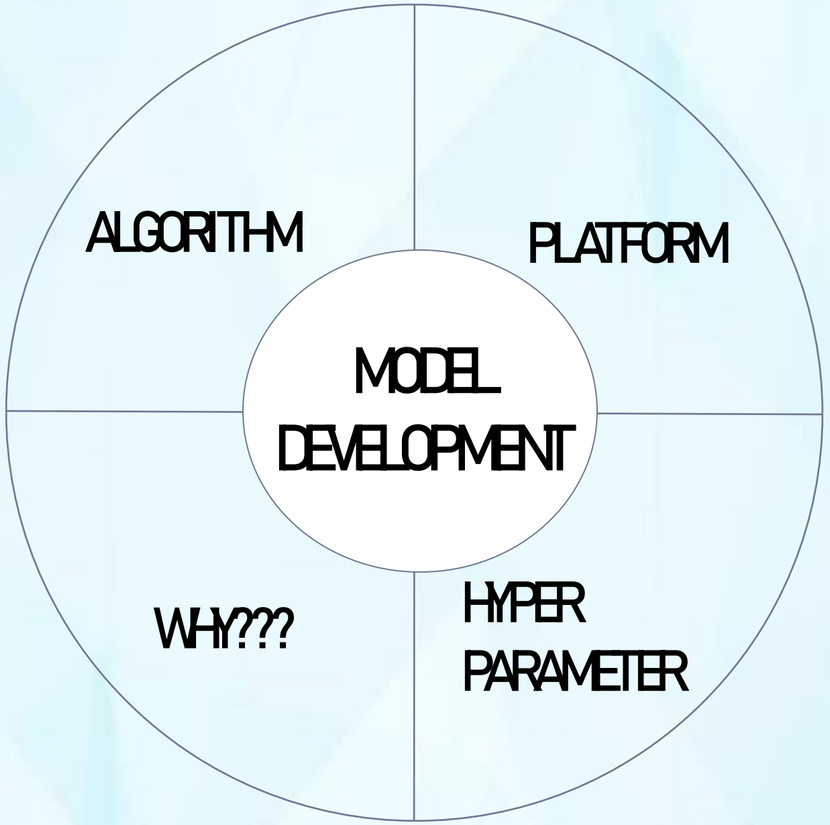
Converted categorical features

- Transformed into numerical form using one-hot encoding before selection



MODEL DEVELOPMENT

- Gradient boosting decision tree that build models sequentially to correct predecessor errors.
- Offers high prediction accuracy, resistance to overfitting, and efficient training.



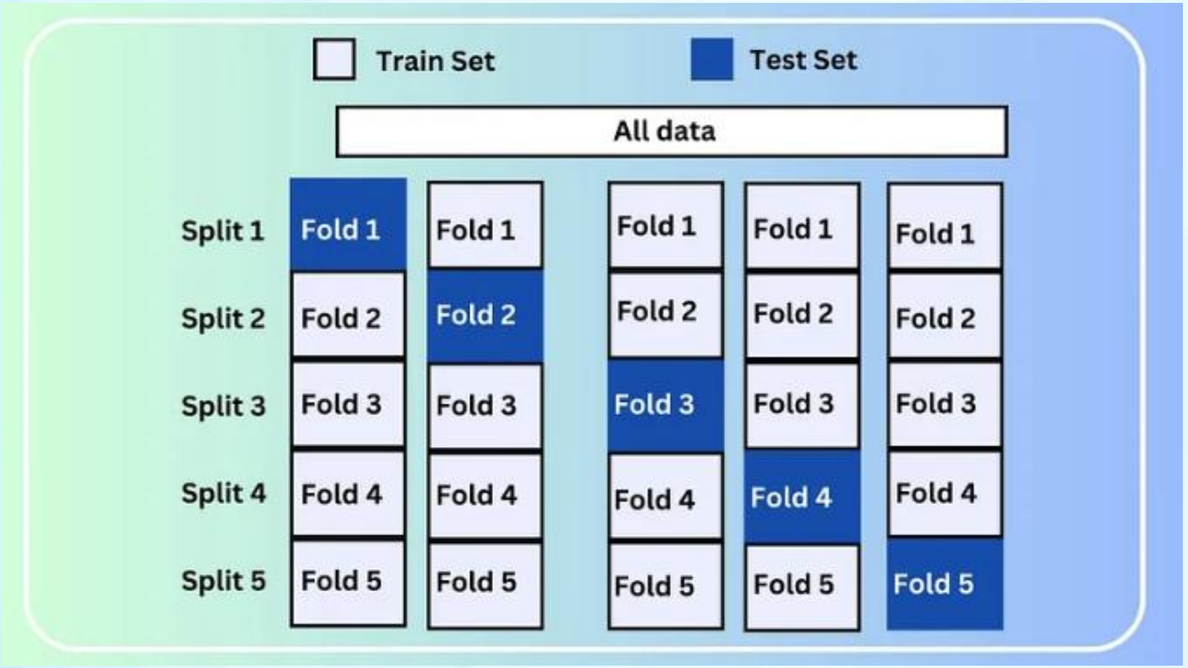
- Google Colab (Python)
- Libraries used: xgboost, scikit-learn, pandas, matplotlib, seaborn

- Handles missing data automatically.
- Supports regularization (L1 & L2) to avoid overfitting.
- Highly optimized for speed and memory.
- Robust to outliers and irrelevant features.

- n_estimators
- learning_rate
- max_depth
- subsample
- random_state

MODEL VERIFICATION

K-Fold Cross Validation



- Split training data into *k* folds.
- Train model on k−1 folds and validate on the remaining one.
- Ensures the model generalizes across different subsets of data.
- Reduces the risk of overfitting and improves reliability of evaluation metrics

MODEL DEPLOYMENT

Backend Development

- Flask-based backend
- Handle user input, interact with the ML model, and provide predictions.

Frontend Development

- Use simple HTML + JavaScript for the frontend to collect user input and display results.
- Outputs prediction results clearly and concisely on the page.

Testing

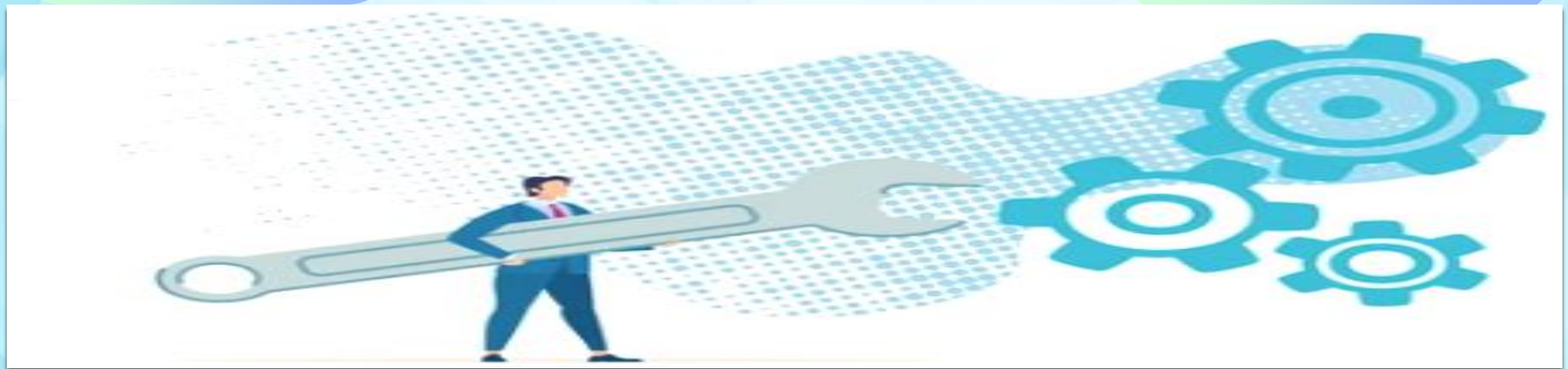
- End-to-end testing conducted locally using a browser interface.
- Verified prediction consistency with validation data samples.

MODEL FUNE-TUNING

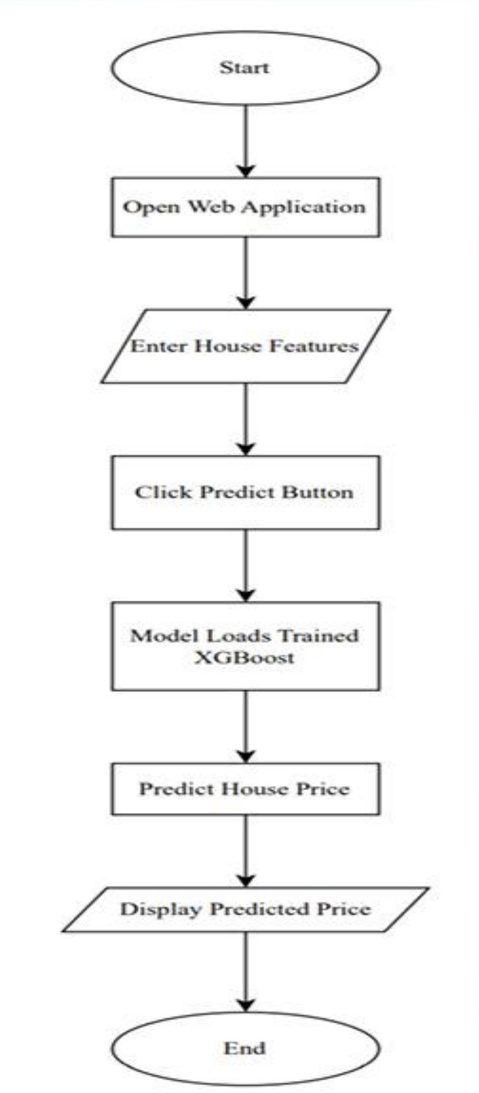
Hyperparameter Tuning

Grid Search

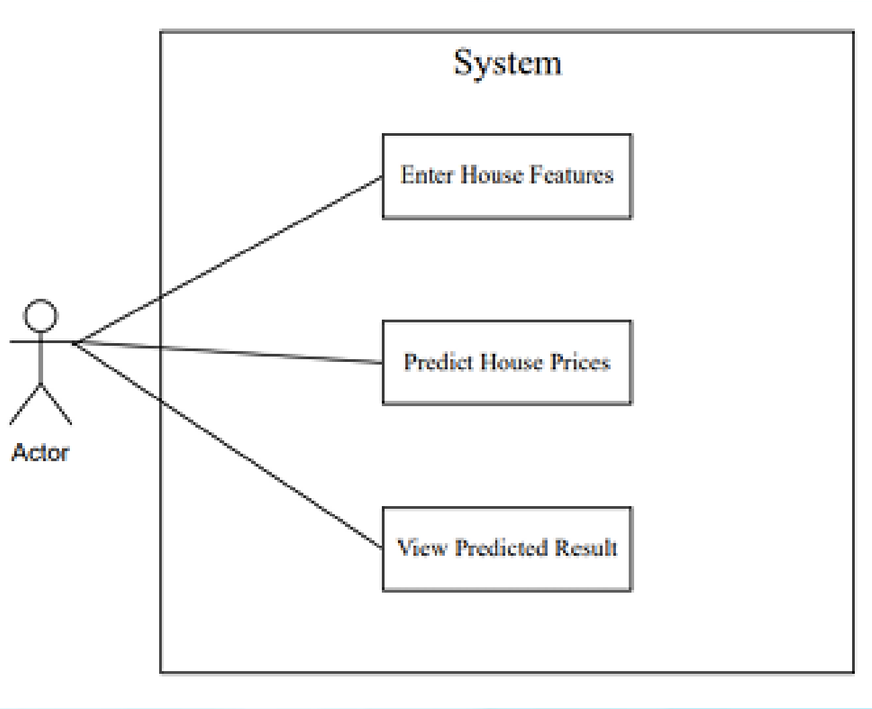
- n_estimators
- max_depth
- learning_rate
- Subsample
- colsample_bytree



FLOWCHART OF WEB-PAGE SYSTEM



USE CASE DIAGRAM



MODEL EVALUATION & PERFORMANCE

Evaluation Metrics Used: MAE, MSE, RMSE, R^2 Score

Dataset Split: 80% Training / 20% Testing

Test Set Results:

- MAE: 16945.41
- MSE: 6.77×10^8
- RMSE: 26030.62
- R^2 Score: 0.9117

Hyperparameter Values:

- n_estimators = [100,200]
- learning_rate = [3,5,7]
- max_depth = [0.05,0.1]
- Subsample = [0.8,1]
- random_state = [0.8,1]

- **Indicates strong predictive power and good generalization to unseen data.**

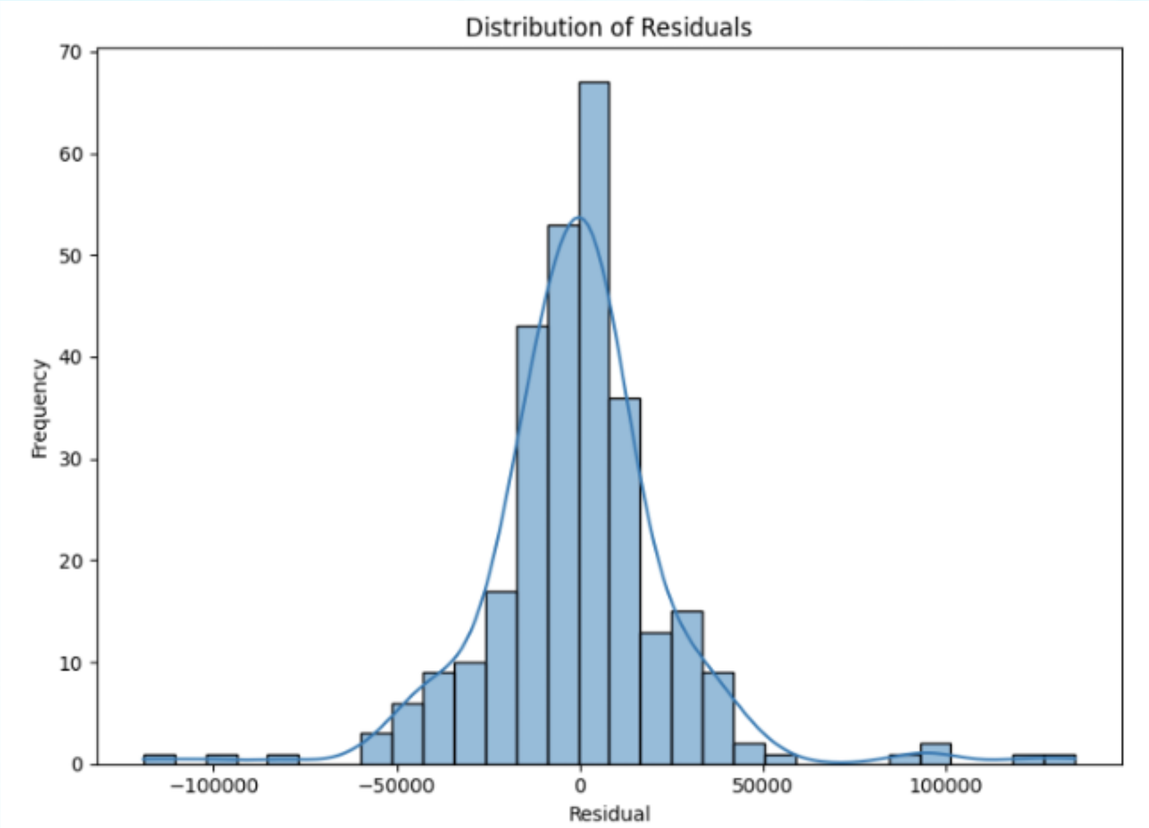
RESULT VISUALIZATION

Actual vs. Predicted Prices

- A red diagonal reference line indicates perfect prediction accuracy.
- Majority of data points cluster tightly along the diagonal, reflecting strong predictive performance.
- Despite a few outliers, the model effectively captures the underlying patterns in the housing market.

RESULT VISUALIZATION

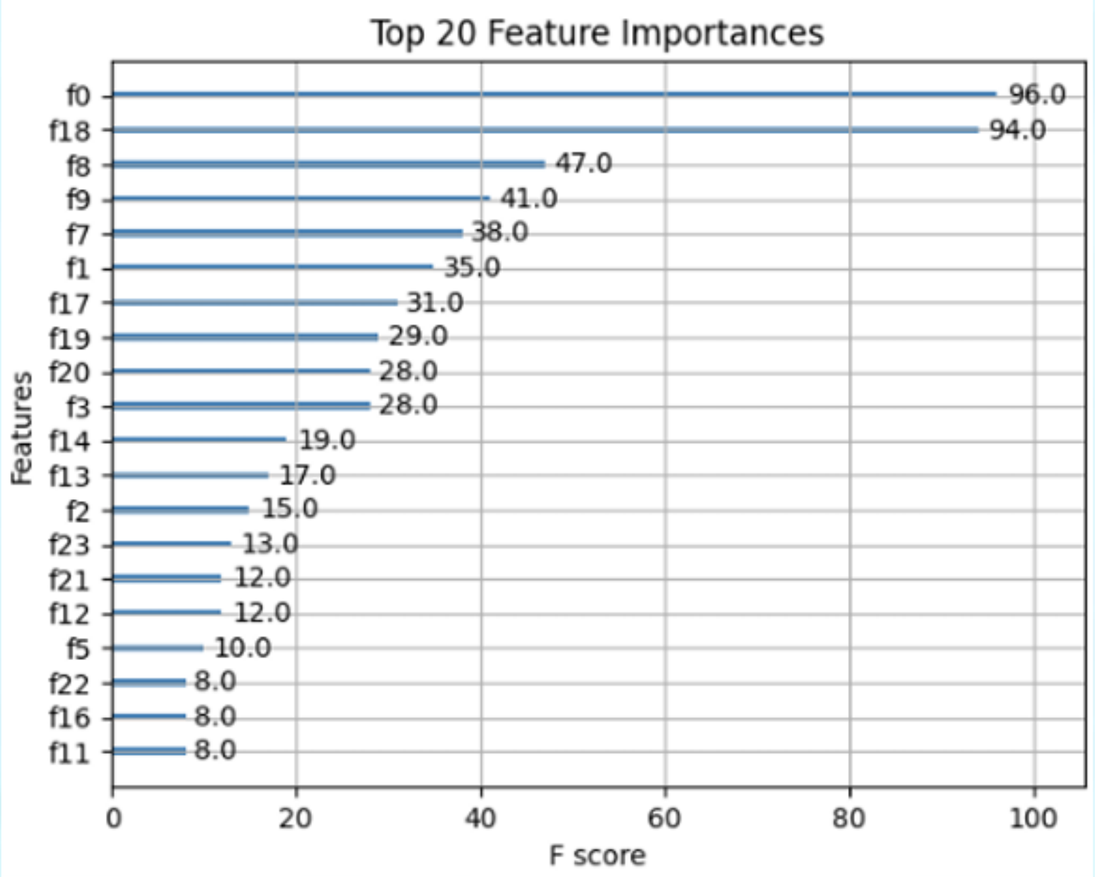
Distribution of Residuals



- This indicates that the model errors are random and unbiased, which is ideal for regression.
- The majority of residuals lie within a narrow range, suggesting high prediction accuracy.
- The distribution's symmetry supports the conclusion that the model neither consistently overestimates nor underestimates prices.

RESULT VISUALIZATION

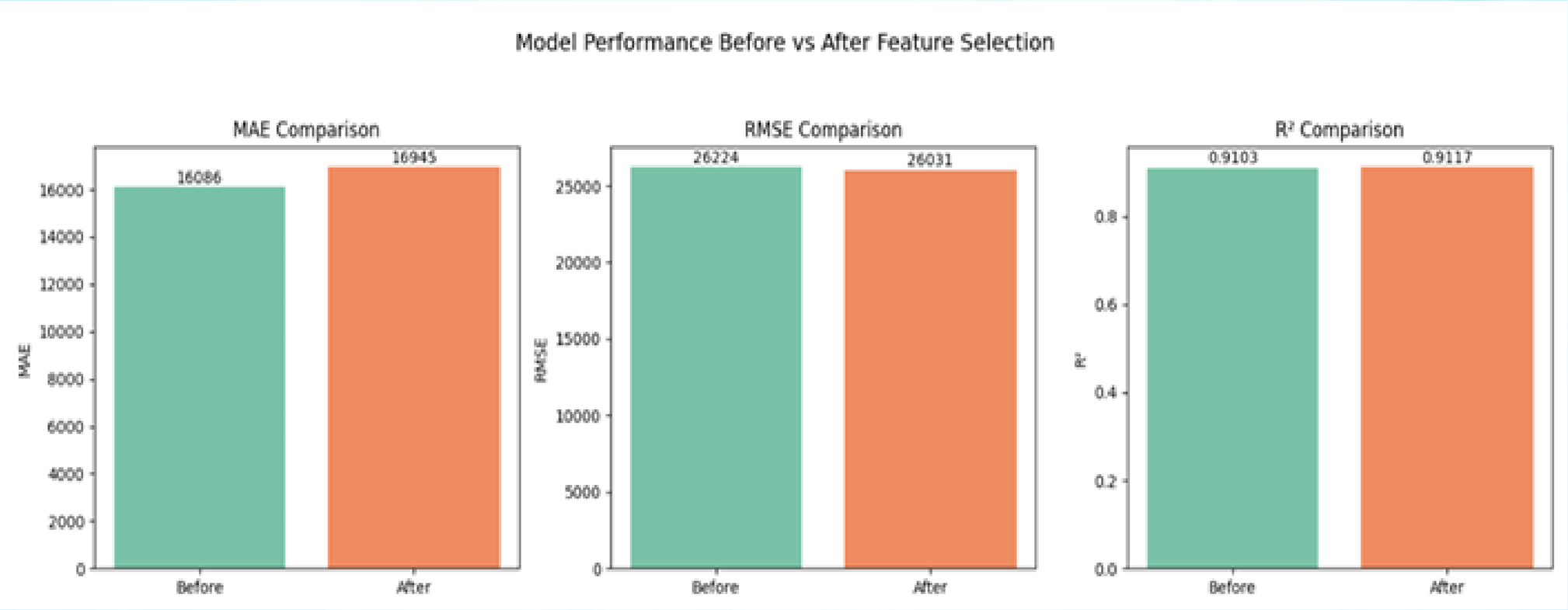
Feature Importance Analysis



- Feature importance confirms that feature engineering was effective.
- The model demonstrates interpretable behaviour without needing domain knowledge.
- Less important features may still contribute to niche cases, and could support future model simplification.

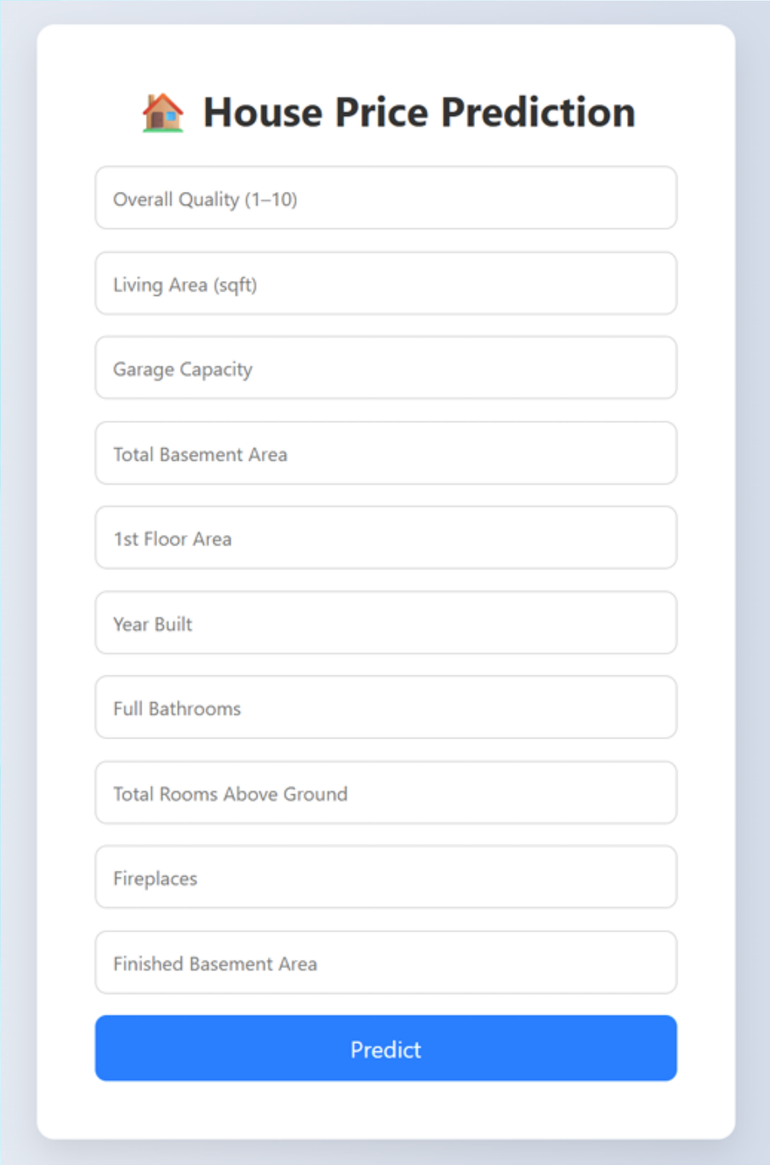
RESULT VISUALIZATION

Performance Comparison Across All Stages

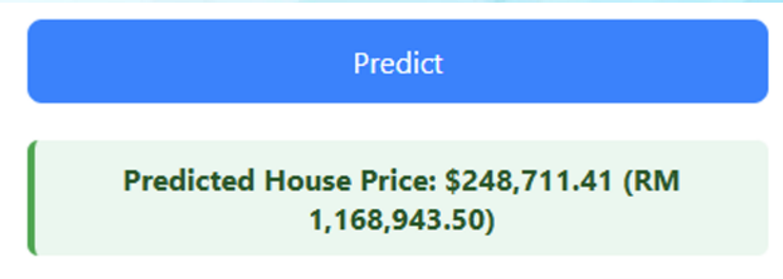


- Each step contributes incremental improvements.
- Final model is more stable, accurate, and generalizable.
- Feature selection helped simplify the model without degrading performance.

WEB-PAGE INTERFACE



SYSTEM OUTPUT



Q & A