2025

Machine Learning & Artificial Intelligence

PREDICTING HOUSE PRICES USING MACHINE LEARNING

YASIR AHMED

Predicting House Prices Using Machine Learning

1. Introduction

1.1. Background & Motivation

The real estate market, a significant force in the global economy, requires precise house price forecasts for well-informed decision-making. Historically, forecasts were based on macroeconomic variables, professional judgments, and historical patterns. However, these approaches frequently lack precision and miss the complex relationships between property attributes and market fluctuations.

The advent of data-driven approaches has allowed us to analyze massive datasets and uncover previously hidden patterns. Cured. Machine learning techniques enable us to build models that predict house prices more accurately, empowering stakeholders and instilling confidence in the proposed approach. To give a thorough study, this project uses regression and classification techniques to investigate the use of Machine learning in the house price prediction.

1.2. Objective

The primary goals of this project are:

- **Data Analysis**: to investigate and preprocess a real estate dataset to find patterns and prepare it for modelling.
- **Feature Engineering**: to create new features that enhance the prediction ability of the models.
- **Model Implementation**: to anticipate home prices using various regression models and evaluate their efficacy.
- **Evaluation**: Evaluate the models' accuracy using the appropriate metrics.
- **Classification Analysis**: to measure the effectiveness of categorization models and group property prices into specified categories.
- **Visualization**: To present the results in an intelligible and visually striking manner

2. Dataset Description

2.1.Data Source & Overview

The dataset employed in this project is a diverse collection of property information, covering details such as property size, number of bedrooms and bathrooms, amenities, and prices. Sourced from [insert data source if available], it is a balanced mix of numerical and categorical features, making it suitable for various regression and classification tasks.

The dataset employed in this project is a diverse collection of proper information covering a spectrum of details such as property size, number of bedrooms, bathrooms, amenities, and prices. Sourced from [https://www.kaggle.com/datasets/yasserh/housing-prices-dataset?resource=download], Its balanced mix of numerical and categorical features makes it suitable for various regression and classification tasks.

2.2.Feature Explanation

Each row in the dataset represents a unique residential property with multiple features describing its physical attributes and amenities. Below is a detailed explanation of the features that I used in my code:

Target Variable

- **Price**: This represents the final selling price of the house. It is the dependent variable in predictive modelling, and various property attributes influence its value.

Independent Variables

- **Area** (**sq ft.):** This feature represents the property's total square footage, including all covered spaces. More significant properties generally have a higher price due to increased land and construction costs.
- **Bedrooms**: The number of bedrooms available in the house. This is a key factor in determining the house's capacity to accommodate residents. More bedrooms generally lead to higher property prices.
- **Bathrooms**: The total number of bathrooms available, including full and half. More bathrooms can increase the value and desirability of a home, particularly for larger families.
- **Stories** are the number of floors in the house. Multi-story houses may have higher values due to better space utilization and additional rooms.
- **Main Road Access**: A categorical feature that indicates whether the house is adjacent to a main road. Properties with primary road access typically have better connectivity, leading to increased demand and potentially higher prices. This feature has values "Yes" or "No".
- **Parking Spaces**: The number of dedicated parking spots available within the property. Houses with more parking spaces tend to have higher prices, especially in urban areas with limited parking.

2.3.Feature Engineering

Developed several new features to enhance the models' performance.

- **Price per Square Foot**: A standardized measure of price calculated as price/area.
- **Total Bathrooms**: A combined feature that adds the number of bathrooms and stories (bathrooms + stories).
- **Bed-Bath Ratio**: A measure of bedroom-to-bedroom ratio, calculated as the

bedrooms / (bathrooms + 1), to assess the home's livability.

2.4.Data Preprocessing

Before training the models, we cleaned and prepared the dataset to ensure accuracy and consistency. Here's what we did:

- **Handled Missing Values:** We checked for missing values and either removed them if they were insignificant or filled them using the mean or median to maintain data integrity.
- **Encoded Categorical Features:** Since some columns, like "Main Road Access" and "Air Conditioning," contained text-based categories, we converted them into numerical values using one-hot encoding so that the models could process them effectively.
- **Scaled Numerical Features:** We normalized all numerical features using the StandardScaler to ensure they were on the same scale, preventing any one feature from dominating the learning process.
- **Reduced Dimensionality:** To remove redundant information while keeping the most critical data, we applied Principal Component Analysis (PCA), preserving 95% of the dataset's variance.

These steps helped structure the data correctly, improving model performance and making training more efficient.

3. Methodology

3.1.Machine Learning Models

We used different machine learning models to predict house prices accurately, each bringing unique strengths. Here's what we did:

Regression Models

- **Linear Regression**: We started with a straightforward approach, assuming a direct relationship between the features and the house price. This gave us a baseline to compare with more complex models.
- **Decision Tree Regressor**: We then used a decision tree, which splits the data into smaller sections based on specific rules. This makes it good at capturing patterns that a simple linear model might miss.
- **Random Forest Regressor**: To make this prediction more reliable, we combined multiple decision trees. This reduced the risk of overfitting and made the model more robust.
- **Gradient Boosting Regressor**: Then, we took it a step further by using gradient boosting, which learns from the mistakes of previous models and keeps perfecting the predictions.

- **XGBoost Regressor:** We chose XGBoost for its speed and efficiency. It's an advanced version of gradient boosting that works well with large datasets and delivers high accuracy.
- **Stacking Regressor:** Instead of relying on a single model, we combined multiple models using stacking. Leveraging the strengths of different regressors allowed us to make more balanced and accurate predictions.

Using these models, we captured simple and complex relationships in the data, leading to better price predictions.

Classification Model

We trained a Random Forest Classifier to categorize house prices into Low, Medium, and High. This model combines multiple decision trees to improve classification accuracy and capture complex relationships in the data.

3.2.Experimental Setup

To evaluate our models effectively, we followed a structured approach:

- **Train-Test Split**: We divided the dataset into **80% training** and **20% test data** to ensure fair evaluation and prevent overfitting.
- **Evaluation Metrics**:
    - **Regression Models**: To assess accuracy and error, we measured performance using **the R^2 Score**, **Mean Absolute Error (MAE)**, and **Mean Squared Error (MSE)**.
    - **Classification Model**: We evaluated the classifier using **Accuracy**, **Confusion Matrix**, **Precision**, and **Recall** to determine how well it categorized house prices.

Following this setup, we ensured our models were trained, tested, and appropriately validated for reliable predictions.

4. Results and Discussion

4.1.Regression Model Performance

Four of our key metrics were used to evaluate the performance of the regression models: R^2 Score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics assessed the accuracy of the house price predictions. The results are presented in the table below.

Model	R^2 Score	MAE	MSE	RMSE
Linear Regression	0.72	250,000	1.2e+11	346,000

Decision Tree	0.80	180,000	9.0e+10	300,000
Random Forest	0.89	140,000	6.1e+10	247,000
Gradient Boosting	0.92	120,000	5.1e+10	226,000
XGBoost	0.87	145,000	6.5e+10	255,000
Stacking Regressor	0.90	135,000	5.5e+10	234,000

Best Performing Model

The **Gradient Boosting Regressor** performed best, with an **R^2 score of 0.92** and the lowest RMSE of 226,000. This indicates that the model explains 92% of the variance in the data and provides the most accurate predictions.

4.2.Classification Model Performance

The Random Forest Classifier (RFM) was used to categorize house prices into three categories: Low, Medium, and High. The performance metrics are as follows:

Metric	Score
Accuracy	76%
Precision	78%
Recall	76%

Analysis of Results

- **Accuracy**: The model correctly classified **76%** of the houses into their respective price categories, showing a solid overall performance.
- **Precision**: With a precision of **78%**, the model effectively minimized false positives, meaning that most of the predicted price categories were accurate.
- **Recall**: The recall score of **76%** indicates that the model successfully captured **76%** of the houses in each price category, demonstrating its ability to identify relevant cases.

Insights from the Confusion Matrix

- The model accurately classified 23 houses in the Low price category and 55 in the High category, demonstrating strong performance in these groups.
- However, it had difficulty distinguishing Medium-priced homes, misclassifying 14 out of 19 as High. The model may require further tuning or additional features to improve its ability to separate medium and high price ranges.
- There were no misclassifications between the Low and Medium categories, indicating a clear distinction between lower-priced homes.

5. Conclusion & Future Work

5.1.Key Takeaways

- The **Gradient Boosting Regressor** delivered the best performance among the regression models, achieving the **highest R^2 score** and the **lowest RMSE**, making it the most accurate predictor of house prices.
- The **Random Forest Classifier** performed moderately in the classification task, reaching **76% accuracy**. While it effectively categorized **Low —**and **high-price** ranges, it struggled to distinguish **Medium-priced** homes.
- **Feature engineering** and **dimensionality reduction** techniques, such as **PCA**, were key in improving model performance by reducing noise and redundant information.

5.2.Future Enhancements

- **Hyperparameter Tuning:** Fine-tuning the Random Forest Classifier's parameters could improve its ability to differentiate between categories with Medium and High prices.
- **Handling Class Imbalance:** Addressing the imbalance in the Medium price category through techniques like oversampling or class weighting could enhance classification accuracy.
- **Advanced Modeling Techniques**: Exploring more sophisticated models, such as neural networks or enhanced ensemble methods, may help capture more complex relationships in the data.
- **Integration with External Data:** Incorporating external economic indicators, such as interest rates and employment statistics, could provide additional insights and improve prediction accuracy.

6. References

- Scikit-learn Documentation (https://scikit-learn.org/)
- XGBoost Documentation (https://xgboost.readthedocs.io/)
- Bergstra, J., & Bengio, Y. (2012). *Random Search for Hyper-Parameter Optimization*. Journal of Machine Learning Research.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research.
- Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.