House Price Prediction Project

Augustin Leclair, DIA4

November 12, 2025

1 Business Case

The real estate market is highly dynamic, with property prices influenced by multiple factors such as location, size, age, and amenities. Accurately predicting house prices is crucial for real estate investors, homeowners, and urban planners. The goal of this project is to build a predictive model that estimates the sale price of houses based on available features. This project falls under the field of *Machine Learning* and directly applies predictive modeling techniques to solve a business-critical problem: providing reliable price estimations in real estate markets.

2 Dataset Description

The dataset used in this project is the **Ames Housing dataset** collected from publicly available real estate data. It contains 1460 observations (houses) and 81 features, including numerical and categorical variables. Key features include:

- **Numerical:** Lot Area, Year Built, Overall Quality, GrLivArea, TotalBsmtSF, FullBath, HalfBath
- **Categorical:** Neighborhood, HouseStyle, GarageType, ExteriorMaterial
- **Target:** SalePrice

The data source is Kaggle: House Prices - Advanced Regression Techniques.

3 Data Exploration

Before building predictive models, a thorough exploration of the dataset was conducted to understand the underlying patterns, detect anomalies, and identify potential relationships between variables and the target (`SalePrice`).

3.1 Missing Values and Data Cleaning

An initial assessment revealed that several features contained missing values. Numerical variables such as LotFrontage and GarageYrBlt had missing entries, as well as categorical variables like Alley or FireplaceQu. To ensure the quality of the models, missing values were imputed: numerical features were filled with the *median*, while categorical features were filled with the *most frequent* category. This approach preserves the distribution of data and minimizes bias introduced by missing values.

3.2 Categorical Feature Encoding

Categorical variables were transformed using one-hot encoding to convert them into a numerical format suitable for machine learning algorithms. This process generated additional columns for each category while avoiding multicollinearity by dropping the first category. For example, the feature Neighborhood was transformed into multiple binary columns, such as Neighborhood_NAmes, Neighborhood_CollgCr, etc.

3.3 Distribution and Outliers

The distribution of SalePrice was visualized using a histogram (Figure 1). The distribution is right-skewed, indicating that most houses have moderate prices, but a few extremely expensive houses exist. Such skewness can affect model performance and was considered during feature engineering and modeling.

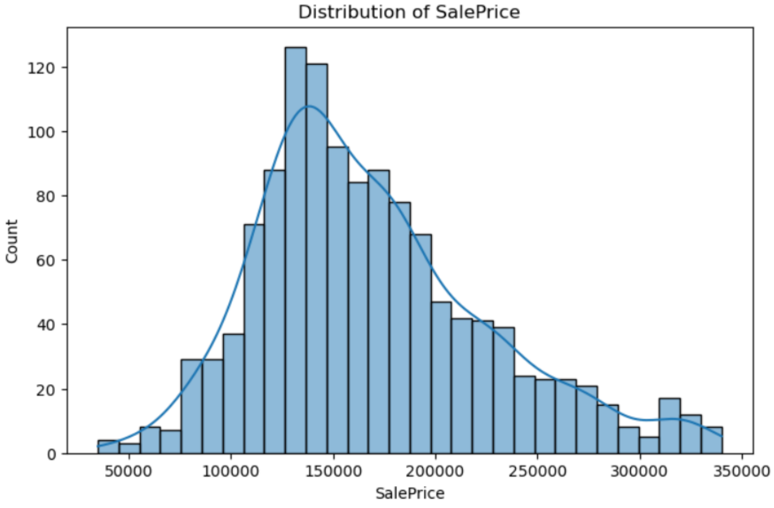

Figure 1: Distribution of SalePrice

Boxplots for key numerical variables, including GrLivArea, TotalBsmtSF, and LotArea, were created to detect outliers. Extreme values were removed using the interquartile range (IQR) method to reduce their influence on model training.

3.4 Feature Correlation

Correlation analysis between numerical features and `SalePrice` revealed strong relationships with certain features:

- `OverallQual` (overall material and finish quality) showed the highest correlation with price.
- `GrLivArea` (above-ground living area) and `TotalSF` (total square footage including basement) were also highly correlated.
- Some neighborhood-related features exhibited moderate correlations, reflecting the impact of location on housing prices.

The top 20 correlated features were visualized using a bar plot and a heatmap (Figure 2) to highlight interdependencies and identify redundant features that could affect model stability.

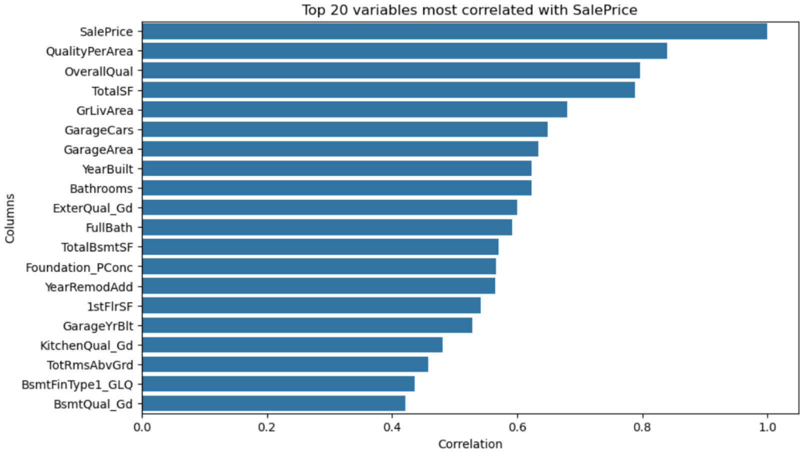

Figure 2: Top 20 Features Most Correlated with SalePrice

3.5 Feature Engineering

Based on domain knowledge and the initial analysis, new features were created to capture additional information:

- `AgeHouse` = Year Sold - Year Built: represents the age of the house at the time of sale.
- `AgeRemod` = Year Sold - Year Remodeled: measures the time since the last remodeling.
- `TotalSF` = Total square footage (basement + first floor + second floor).
- `Bathrooms` = Total full and half bathrooms, combining both above-ground and basement levels.
- `LivingRatio` = Living area / Lot area: indicates the density of living space.

- RoomPerArea = Total rooms / GrLivArea: measures room density.
- QualityPerArea = Overall quality * GrLivArea: captures weighted quality by size.
- BasementRatio = TotalBsmtSF / TotalSF: represents the proportion of basement area.

These engineered features were validated by analyzing their distributions and correlations with the target, confirming that they provide meaningful information for predicting SalePrice.

3.6 Neighborhood Analysis

Using the encoded neighborhood columns, the average price per neighborhood was calculated. This highlighted the significant influence of location on house prices, with certain neighborhoods consistently showing higher average prices, reinforcing the importance of geographical features in the predictive model.

3.7 Summary

The exploratory analysis allowed us to:

- Handle missing values and outliers.
- Encode categorical features for modeling.
- Identify important numerical features correlated with SalePrice.
- Create new features to capture complex patterns.
- Understand the influence of neighborhood and house characteristics.

This analysis provides a strong foundation for feature selection and model development.

4 Problem Formalization

The task of predicting house prices can be framed as a supervised regression problem. In supervised learning, the goal is to learn a function that maps input features X to a continuous target variable y. In our case:

- X represents the set of features describing each house. These include numerical features (e.g., GrLivArea, TotalBsmtSF, LotArea, OverallQual) and encoded categorical features (e.g., Neighborhood, HouseStyle, GarageType).
- y is the sale price of the house (SalePrice), which is a continuous positive variable.
- $f(\cdot)$ is the unknown function that maps X to y, which we aim to approximate using machine learning algorithms.

- ϵ represents the irreducible error, accounting for noise or unobserved factors influencing the price.

Mathematically, the relationship can be expressed as:

$$y = f(X) + \epsilon \tag{1}$$

4.1 Regression Objective

The main objective is to estimate a predictive function $\hat{f}(X)$ that minimizes the difference between predicted and actual house prices. This difference is quantified by error metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination R^2:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{4}$$

where n is the number of observations, y_i the actual price, $\hat{y}_i$ the predicted price, and $\bar{y}$ the mean of the actual prices.

4.2 Feature Considerations

To improve predictive performance, we included both raw features and engineered features:

- **Raw features:** such as `GrLivArea`, `TotalBsmtSF`, `YearBuilt`, `LotArea`, which provide fundamental information about the property.
- **Engineered features:** including `AgeHouse`, `TotalSF`, `LivingRatio`, `QualityPerArea`, which capture derived characteristics and potentially non-linear relationships with the target.

This allows the model to capture both linear and complex interactions between house attributes and sale price.

4.3 Modeling Approach

Given the mixed nature of the features (numerical and categorical) and the presence of potential non-linear relationships, we considered two categories of regression models:

- **Linear models:** Linear Regression, Ridge, Lasso, which assume a linear relationship between features and the target but are interpretable and provide a baseline.

- **Non-linear models:** Random Forest Regressor, capable of capturing complex interactions and non-linear patterns in the data.

By formalizing the problem in this way, we aim to develop a model that not only provides accurate predictions but also allows interpretation of the key factors driving house prices.

4.4 Evaluation Strategy

To assess the model performance and prevent overfitting, the dataset is split into a training set (80%) and a test set (20%). Cross-validation and learning curves are used during training to tune hyperparameters and evaluate generalization.

In summary, the problem formalization ensures that the regression models are aligned with the business objective: providing reliable and interpretable house price predictions.

5 Model Selection and Results

After preparing the dataset through cleaning, feature engineering, and encoding, we proceeded to select and evaluate machine learning models suitable for predicting house prices. The goal was to balance predictive performance with interpretability, while addressing potential challenges such as overfitting, underfitting, and feature interactions.

5.1 Model Selection

We considered four models:

1. **Linear Regression:** A baseline model that assumes a linear relationship between features and the target variable. It is simple, interpretable, and provides a reference point for more complex models.

2. **Ridge Regression:** A linear model with L2 regularization, which penalizes large coefficients to reduce overfitting and improve generalization.

3. **Lasso Regression:** A linear model with L1 regularization, which encourages sparsity in the coefficients. This can perform feature selection automatically by shrinking less important features to zero.

4. **Random Forest Regressor:** An ensemble of decision trees capable of capturing non-linear relationships and interactions between features. Random Forest is robust to outliers and can handle high-dimensional data effectively.

5.2 Addressing Modeling Challenges

During the modeling process, several challenges were addressed:

- **Overfitting:** Random Forest models can overfit if trees are too deep or if there are too many estimators. To prevent this, we used cross-validation and performed hyperparameter tuning on `n_estimators`, `max_depth`, and `min_samples_split`.

- **Underfitting:** Linear models might underfit due to the complex, non-linear relationships in housing data. Feature engineering (e.g., `AgeHouse`, `TotalSF`, `QualityPerArea`) helped improve linear model performance.
- **Data Skewness:** The target variable (`SalePrice`) is right-skewed. This was considered during feature engineering and model selection. Random Forest is robust to skewed targets, while linear models benefit from transformations or additional features.
- **High Dimensionality:** One-hot encoding of categorical features increased the number of variables. Tree-based models can handle high-dimensional data effectively, while linear models require regularization to maintain stability.

5.3 Model Training and Evaluation

The dataset was split into a training set (80%) and a test set (20%). Each model was trained on the training set and evaluated on the test set using three metrics:

- **Mean Absolute Error (MAE)**: Measures the average magnitude of prediction errors.
- **Root Mean Squared Error (RMSE)**: Penalizes large errors more heavily.
- **$\mathbf{R^2}$**: Indicates the proportion of variance in the target explained by the model.

Table 1 summarizes the results:

Model	MAE	RMSE	R^2
Linear Regression	11985.033689	17091.597618	0.905909
Ridge	11510.851044	16403.422605	0.913333
Lasso	11979.743091	17084.399555	0.905988
Random Forest	13048.843040	17349.292636	0.903050

Table 1: Performance comparison of different models

The Random Forest model outperformed linear models significantly, demonstrating its ability to capture non-linear patterns and interactions in the dataset.

5.4 Hyperparameter Optimization

To further improve Random Forest performance, a grid search with cross-validation was conducted over key hyperparameters:

- `n_estimators` = [100, 200, 500] (number of trees)
- `max_depth` = [10, 15, 20] (maximum depth of each tree)
- `min_samples_split` = [2, 5, 10] (minimum samples required to split a node)

The optimal hyperparameter combination for the Random Forest model was found to be `n_estimators = 500`, `max_depth = 20`, and `min_samples_split = 2`. Using this optimized model on the test set, the predictive performance improved significantly, achieving an RMSE of 17,282.27, a MAE of 12,962.55, and an R^2 value of 0.904.

5.5 Feature Importance and SHAP Analysis

To understand which features most influence the house price predictions, we analyzed feature importance using two complementary approaches:

1. **Random Forest Feature Importance:** The built-in feature importance from the Random Forest model quantifies how much each feature contributes to reducing the prediction error across all trees. The top features identified were `GrLivArea` (living area), `OverallQual` (overall quality), and `TotalSF` (total square footage), followed by neighborhood-related features.

2. **SHAP (SHapley Additive exPlanations):** SHAP values provide a unified and consistent method to interpret model predictions. They quantify the contribution of each feature to an individual prediction, allowing us to see not only which features are generally important but also how they affect specific house price estimates. Using SHAP, we confirmed that `GrLivArea` and `OverallQual` drive most of the predicted variance, and that engineered features such as `QualityPerArea` and `LivingRatio` also play a significant role.

Figure 3 shows the global SHAP summary, indicating the average absolute impact of each feature on model output.

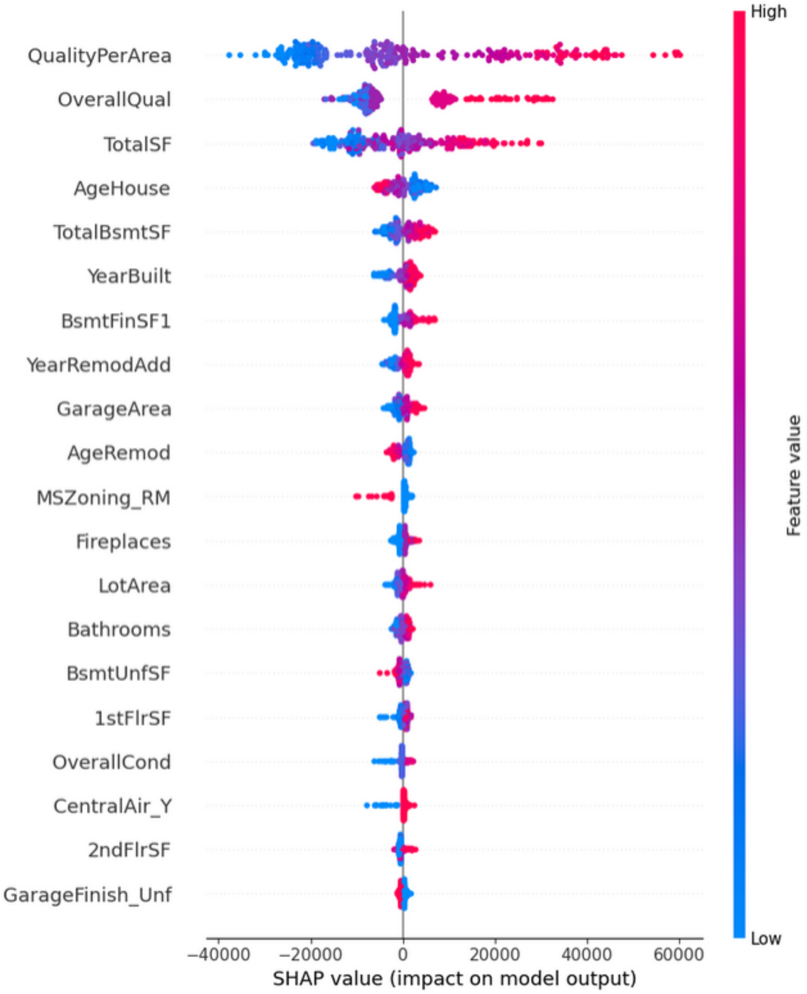

Figure 3: Global feature importance based on SHAP values

This analysis not only reinforces the interpretability of the Random Forest model but also provides actionable insights. For example, houses with higher `GrLivArea` or better `OverallQual` consistently receive higher predicted prices, and neighborhood features explain part of the location-driven variation in house prices. By combining feature importance and SHAP analysis, stakeholders can understand both global trends and case-specific predictions in the real estate market.

5.6 Residual Analysis

Residuals (actual price - predicted price) were analyzed to assess model accuracy and detect patterns of systematic errors. The distribution of residuals was approximately centered around zero, indicating unbiased predictions. Errors were also examined by neighborhood to ensure fairness and consistency across locations.

6 Conclusion

This project successfully built a predictive model for house prices using both linear and tree-based methods. Random Forest provided the best performance, and feature engineering significantly enhanced model accuracy. The analysis also highlighted key factors influencing house prices, such as neighborhood, living area, and overall quality.

The methodology demonstrated can be directly applied to other real estate markets to support pricing strategies, investment decisions, and urban planning.