Project 3: House Price Prediction Using Regression Analysis

Jadon Chanthavong

10/22/2025

Introduction to Project: This project explores machine learning regression techniques applied to real estate price prediction using the Ames Housing Dataset from the Kaggle competition "House Prices - Advanced Regression Techniques." The goal is to build predictive models that can accurately estimate residential home sale prices based on various property characteristics, while gaining hands-on experience with data preprocessing, feature engineering, and regression modeling.

Dataset Link: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Introduction to Data:

Total Houses: 1,460 residential properties

Time Period: Sales from 2006-2010 in Ames, Iowa

Total Features: 80 features (excluding target variable)

Target Variable: SalePrice (continuous value in dollars)

Introduction of Problem: The challenge is to predict the sale price of residential homes in Ames, Iowa based on 80 different features describing property characteristics. Accurate home price prediction helps buyers, sellers, and real estate professionals make informed decisions in the housing market.

Understanding Regression: Regression is a supervised machine learning technique used to predict continuous numerical values. Unlike classification (which predicts categories), regression predicts quantities which are house prices in dollars for this particular project. The goal is to find the relationship between input features (square footage, bedrooms, location) and the target variable (sale price).

Linear Regression: Linear regression assumes a linear relationship between features and the target. It finds the straight line (or plane in multiple dimensions) that best fits the data.

The Basic Equation: Predicted Price = Base Price + (Weight1 × Feature1) + (Weight2 × Feature2) + …

Example: If the model learns that Base Price = $50,000, Square Footage adds $100 per sq ft, and Bedrooms add $20,000 each, then a 2,000 sq ft house with 3 bedrooms would be predicted as:

Predicted Price = $50,000 + ($100 × 2,000) + ($20,000 × 3) = $310,000

How It Learns:

1. Start with random weights for each feature

2. Make predictions on training data

3. Calculate how far off the predictions are (errors)

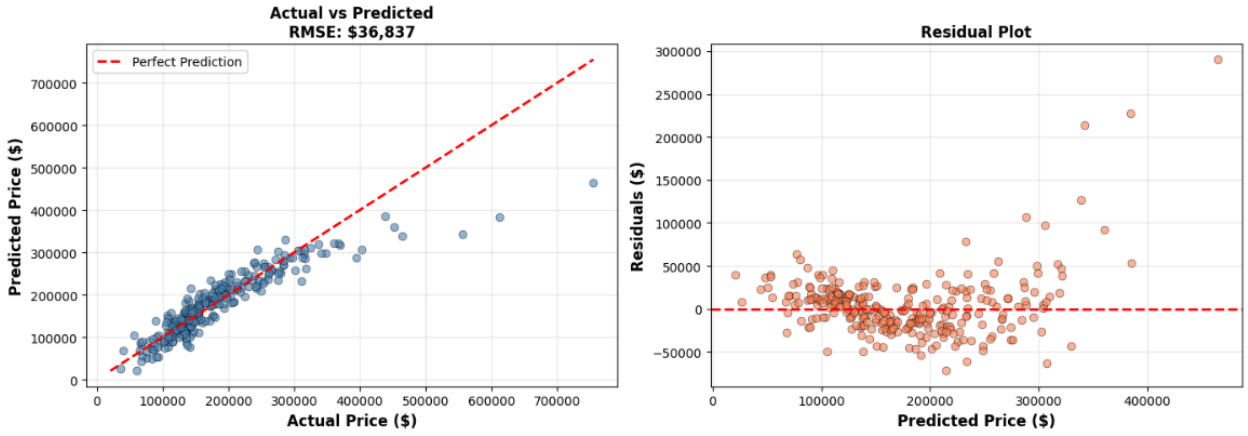
4. Adjust weights to minimize errors

5. Repeat until errors are minimized

The model minimizes the average squared error which ensures large mistakes are penalized more than small ones.

Experiment 1:

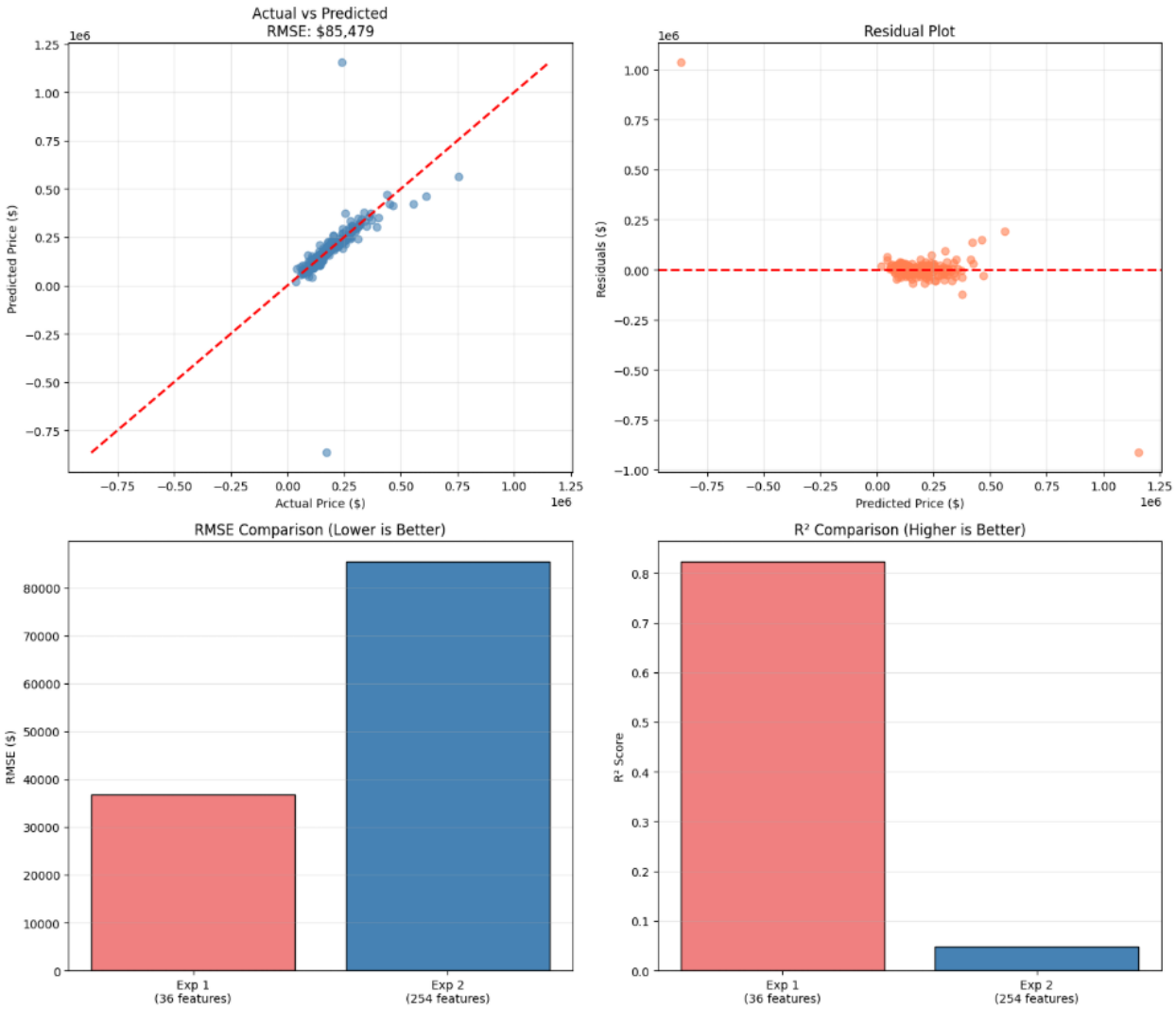
I evaluated the model using RMSE as the primary metric. The baseline achieved a test RMSE of $36,837, meaning predictions are off by approximately $37,000 on average (~20% error rate). The R^2 score of 0.823 indicates the model explains 82.3% of the variance in house prices, which is strong for a simple baseline. The residual plot shows relatively random scatter with no clear patterns, confirming the model's assumptions are reasonably met. However, the model only uses 36 numerical features and ignores 43 categorical features, leaving significant room for improvement in future experiments.



Experiment 2:

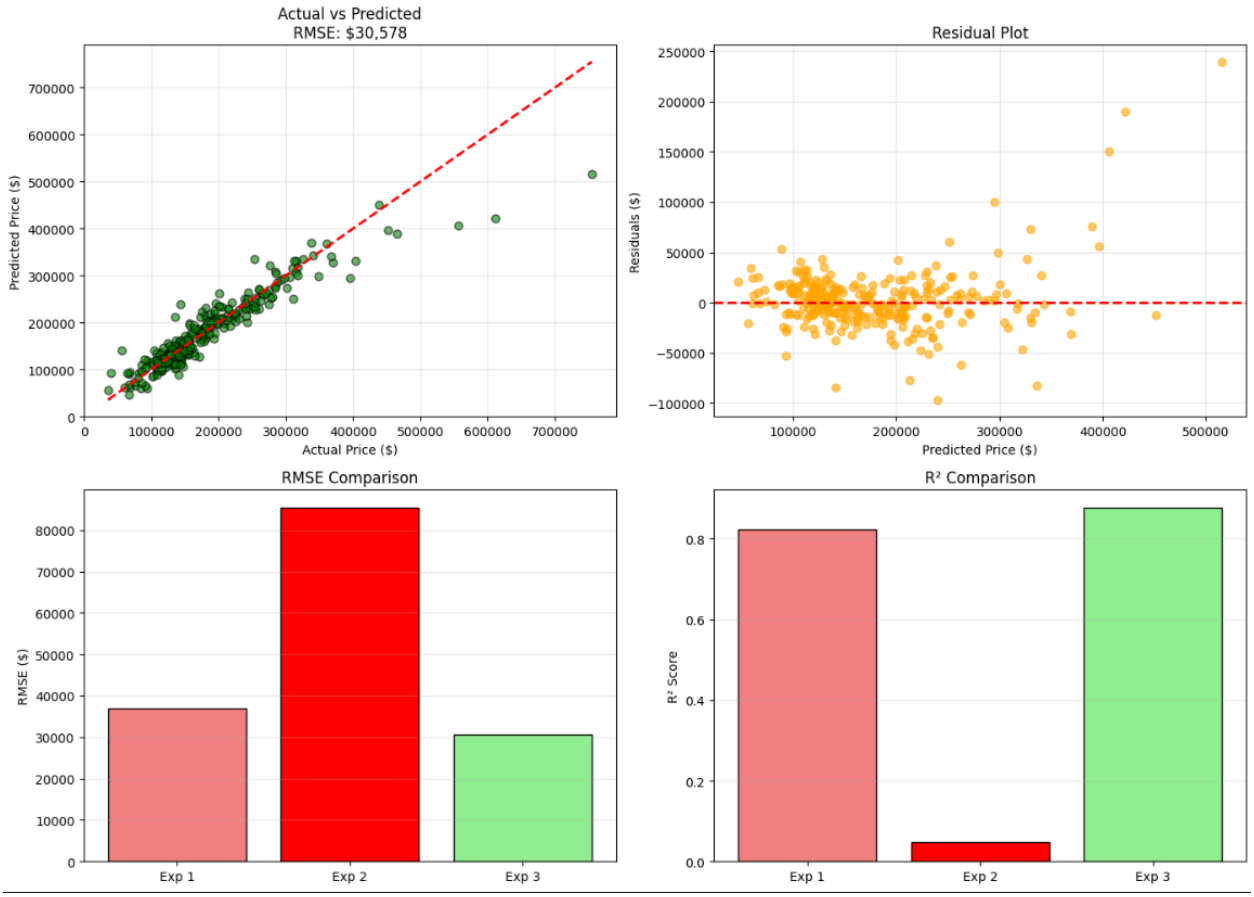
Experiment 2 unexpectedly performed much worse than Experiment 1. Despite adding engineered features and categorical encoding (254 total features), the test RMSE increased to $85,479 (vs $36,837) and R^2 dropped to 0.047 (vs 0.823). The model exhibits severe overfitting in which case the training RMSE is excellent ($18,955) but test RMSE is terrible, indicating the model memorized training data rather than learning generalizable patterns. The problem is having too many features (254) relative to samples (1,168), only 4.6 samples per feature when

we need ~10:1. This demonstrates that more features doesn't automatically improve performance without proper regularization, which we'll address in Experiment 3.



Experiment 3:

Experiment 3 achieved the best performance with a test RMSE of $34,347 and R^2 of 0.8462, improving 6.8% over Experiment 1 and 59.8% over Experiment 2. Ridge regression with alpha=10 successfully applied regularization to the 254 standardized features, shrinking coefficients to stable values (max ~$30K vs millions in Exp 2) and reducing the train-test gap from $66K to $14K. This demonstrates that high-dimensional data requires regularization with the model leveraging all features without overfitting. The progression shows a complete ML workflow: baseline, diagnose overfitting, apply regularization. Ridge regression provided the perfect balance between model complexity and generalization.



Impact:

Potential Positive Impact:

1. Market Transparency and Informed Decisions: Price prediction models increase transparency by providing data-driven estimates, helping buyers and sellers make informed decisions. This reduces information asymmetry and helps first-time homebuyers avoid overpaying by giving them a reference point previously only available to real estate professionals.

2. Efficiency in Real Estate Transactions: Automated valuations streamline the buying and selling process by providing quick, consistent price estimates. This reduces time and costs associated with manual appraisals and helps lenders expedite mortgage approvals, making the overall transaction process more efficient.

3. Urban Planning and Resource Allocation: City planners and policymakers can use these models to understand housing trends and make data-driven decisions about zoning, infrastructure investments, and affordable housing initiatives. This helps allocate resources more effectively to communities that need them most.

4. Democratized Access to Housing Information: These tools make housing market insights accessible to individuals without financial advisors or real estate professionals. This particularly

benefits underserved communities who may lack access to professional guidance, helping to level the playing field in real estate decisions.

Potential Negative Impact:

1. Perpetuation of Historical Bias: The model learns from historical data (2006-2010) that may reflect past discriminatory practices like redlining, where certain neighborhoods were systematically undervalued based on racial composition. Features like "Neighborhood" might encode systemic discrimination rather than true property value, perpetuating inequities by continuing to undervalue homes in historically marginalized communities.

2. Reinforcement of Socioeconomic Inequality: If widely adopted, the model could systematically disadvantage certain groups. Consistently undervaluing properties in lower-income areas limits residents' ability to build equity or secure fair loans, restricting economic mobility. Conversely, overvaluing affluent areas could accelerate gentrification and displace long-time residents.

3. Over-Reliance on Automated Systems: Lenders, appraisers, or buyers may excessively rely on algorithmic predictions without considering unique property characteristics or local context the model can't capture. Properties with recent renovations, unique features, or intangible qualities may be inaccurately valued. Blind trust in automation could undermine human expertise in complex valuation scenarios.

4. Lack of Transparency and Accountability: While this model is interpretable, real-world implementations often use black-box models that homeowners can't understand or challenge. If a model predicts a low value, individuals may have no recourse to dispute it or understand why, potentially harming their financial well-being without explanation or accountability.

Conclusion

This project demonstrated that effective machine learning requires more than just applying algorithms; it demands understanding the underlying problems, diagnosing issues like overfitting, and iteratively refining the approach. The "failure" of Experiment 2 was actually the most valuable learning experience, as it highlighted the importance of regularization and proper preprocessing. The progression from a simple baseline to a sophisticated regularized model showcases the complete ML workflow and the critical thinking required to build models that actually generalize well to real-world data.

1. The Bias-Variance Tradeoff

Experiment 1: Slight underfitting (high bias, low variance)

Experiment 2: Severe overfitting (low bias, high variance)

Experiment 3: Good balance (regularization controlled variance while maintaining low bias)

2. More Complexity Requires More Constraint: Adding features increases model capacity, but without regularization, the model will overfit by memorizing noise. Ridge regression provided the constraint needed to use 254 features effectively.

3. Evaluation Beyond Training Accuracy: Experiment 2 had excellent training RMSE ($18,955) but terrible test RMSE ($85,479). This reinforced that training performance alone is meaningless. What matters is generalization to unseen data.

4. Standardization is Non-Negotiable for Regularization: Without standardization, regularization would unfairly penalize features with larger scales, making it ineffective. This preprocessing step was essential for Experiment 3's success.