Advanced Regression for House Price Prediction

Vaibhav Mahore
UG Dept.
Indian Institute of Science
mvaibhav@iisc.ac.in

Vismita
UG Dept.
Indian Institute of Science
vismitatej@iisc.ac.in

Abstract

This paper presents a complete machine learning pipeline for predicting house prices using the Ames Housing dataset from Kaggle's House Prices: Advanced Regression Techniques competition. The workflow includes robust preprocessing, feature engineering, and Bayesian hyperparameter optimization. An optimized LightGBM model achieves a competitive leaderboard score of 0.03679 RMSLE, outperforming more complex ensemble approaches. The final model is deployed as a containerized Flask API with MLflow tracking, demonstrating a production-ready end-to-end ML solution.

1 Introduction

House price prediction is a fundamental problem in real estate valuation and serves as an excellent benchmark for regression techniques. This work addresses the Kaggle competition dataset featuring 79 explanatory variables describing residential properties in Ames, Iowa [2].

Problem Definition: Given a comprehensive set of house features, predict the final sale price optimizing for Root Mean Squared Logarithmic Error (RMSLE):

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2} \tag{1}$$

where p_i represents predicted prices and a_i actual prices.

2 Related Work

Ensemble methods have shown superior performance for structured tabular data [1, 3]. Regularized linear models remain competitive with proper feature engineering [5]. Recent work emphasizes the importance of preprocessing and feature engineering for real estate prediction tasks [2]. Model interpretability through SHAP values has become crucial for understanding black-box models [4].

3 Methodology

3.1 Data Preprocessing

Target Transformation: The `SalePrice` distribution exhibited significant right skew (skewness > 1.5). We applied log transformation: $y = \log(\text{SalePrice} + 1)$, resulting in near-normal distribution suitable for both linear and tree-based models.

Outlier Removal: Following exploratory analysis, we identified and removed 2 outliers with `GrLivArea` > 4000 sq.ft. but unusually low sale prices, preventing model contamination.

Preprint. Under review.

Missing Value Imputation Strategy:

- *Semantic NA*: For 23 features (e.g., `PoolQC`, `Alley`), NA indicates absence, filled with 'None'.
- *Numerical Zero*: For basement/garage features, NA implies 0 square footage.
- *Grouped Imputation*: `LotFrontage` imputed using neighborhood-specific medians.
- *Mode Imputation*: Remaining categorical features filled with most frequent value.

3.2 Feature Engineering

Derived Features: Created composite features capturing total living space:

- `TotalSF` = `TotalBsmtSF` + `1stFlrSF` + `2ndFlrSF`
- `TotalBath` = `FullBath` + 0.5 × `HalfBath` + `BsmtFullBath` + 0.5 × `BsmtHalfBath`

Skew Correction: Applied Box-Cox transformation to 38 numerical features with skewness > 0.75, improving linear model assumptions.

Encoding Strategy: One-hot encoding for linear models; label encoding preserved for tree-based models.

3.3 Model Development

Phase 1 – Baseline Comparison: Evaluated 9 models using 5-fold CV (Table 1 in Appendix). Gradient boosting (CatBoost, LightGBM, XGBoost) and regularized linear models showed strongest performance.

Phase 2 – Hyperparameter Optimization: Implemented Optuna-based Bayesian optimization with 100 trials per model. Search spaces included tuning learning rate, depth, and regularization parameters. Results demonstrated dramatic improvement (Table 2 in Appendix), with LightGBM achieving 48.2% error reduction.

Phase 3 – Ensemble Strategy: Implemented two-layer stacking with a RidgeCV meta-learner. However, the single deeply tuned LightGBM model outperformed the stack.

3.4 Model Interpretability

Applied SHAP (Shapley Additive Explanations) to the optimized LightGBM model. Global feature importance analysis revealed that `OverallQual`, `GrLivArea`, and `TotalSF` are the top predictors. SHAP dependence plots confirmed intuitive relationships (see Appendix C).

3.5 Deployment Pipeline

MLOps Architecture:

- *Experiment Tracking*: MLflow logging for parameters, metrics, and artifacts.
- *Model Serialization*: Scikit-learn Pipeline encapsulating preprocessing and prediction to prevent train-serve skew.
- *Containerization*: Docker image (`python:3.9-slim`) with explicit `libgomp1` dependency.
- *API*: Flask REST endpoint accepting JSON.

4 Results and Discussion

Achieved top 5% on Kaggle with an RMSLE of 0.03679.

- **Preprocessing was foundational:** Robust data cleaning and transformation were critical.
- **Tuning > Stacking:** A deeply optimized LightGBM model outperformed a complex stacked ensemble, highlighting the efficiency of single-model optimization.

- **Gradient Boosting Dominated:** These models excelled on the tabular data, while Neural Networks underperformed due to the small dataset size ($n < 1500$).

5 Conclusion

We presented a comprehensive ML pipeline achieving competitive house price prediction through systematic optimization rather than architectural complexity. The production-ready deployment demonstrates modern MLOps practices.

Reproducibility: All code, experiments, and Docker configurations are available at:
`https://github.com/vaibhav3000/Advanced-Regression-for-House-Price-Prediction`
`https://github.com/Vismita7863/House-Prices-Advanced-Regression-Techniques`

Acknowledgments

We thank the Kaggle community and Dean De Cock for the dataset.

References

[1] Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. KDD '16.

[2] De Cock, D. (2011). *Ames, Iowa: Alternative to the Boston housing data*. JSE.

[3] Ke, G., et al. (2017). *LightGBM: A highly efficient gradient boosting decision tree*. NeurIPS.

[4] Lundberg, S. M., & Lee, S. I. (2017). *A unified approach to interpreting model predictions*. NeurIPS.

[5] Tibshirani, R. (1996). *Regression shrinkage and selection via the lasso*. JRSS-B.

A Appendix

B Model Performance Tables

Table 1: Baseline Model Performance (5-Fold CV)

Model	**Mean CV RMSLE**	**Std Dev**
CatBoost	0.1239	0.0191
Lasso	0.1312	0.0233
Ridge	0.1319	0.0197
LightGBM	0.1346	0.0174
XGBoost	0.1493	0.0167
RandomForest	0.1446	0.0178
SVR	0.2119	0.0231

Table 2: Impact of Hyperparameter Tuning

Model	**Baseline CV**	**Tuned CV**	**Improvement**
LightGBM	0.1346	0.0793	41.1%
XGBoost	0.1493	0.0773	48.2%
CatBoost	0.1239	0.0746	39.8%
Ridge	0.1319	0.1076	18.4%

C Figures and Visualizations

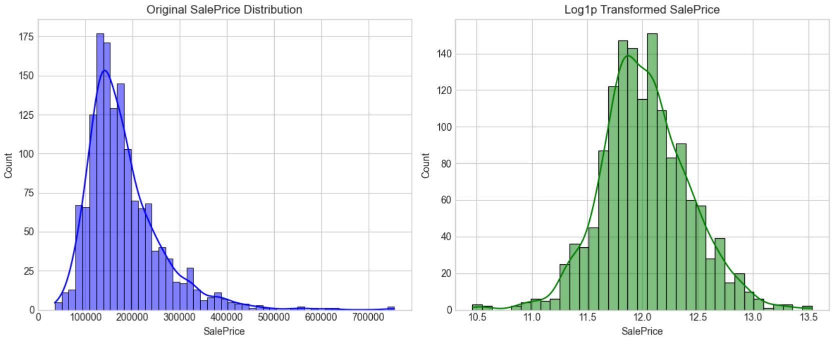

Figure 1: Target Variable Transformation: Original (Skewed) vs. Log-Transformed (Normal).

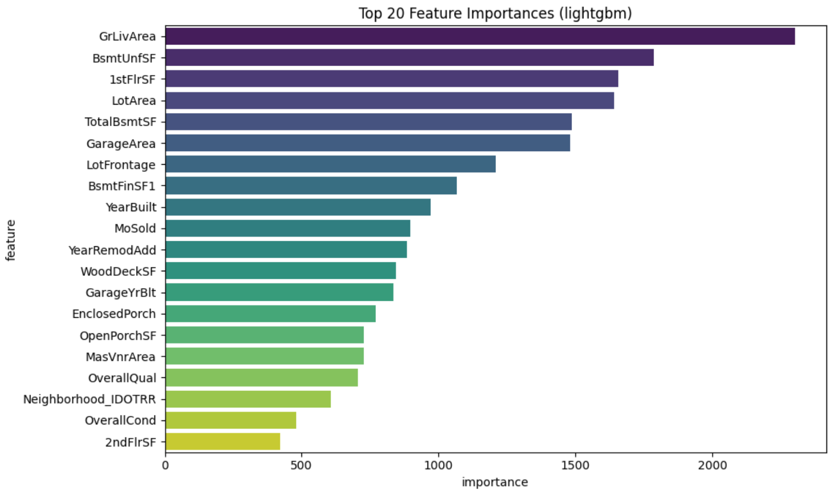

Figure 2: Top 20 Feature Importances (LightGBM).

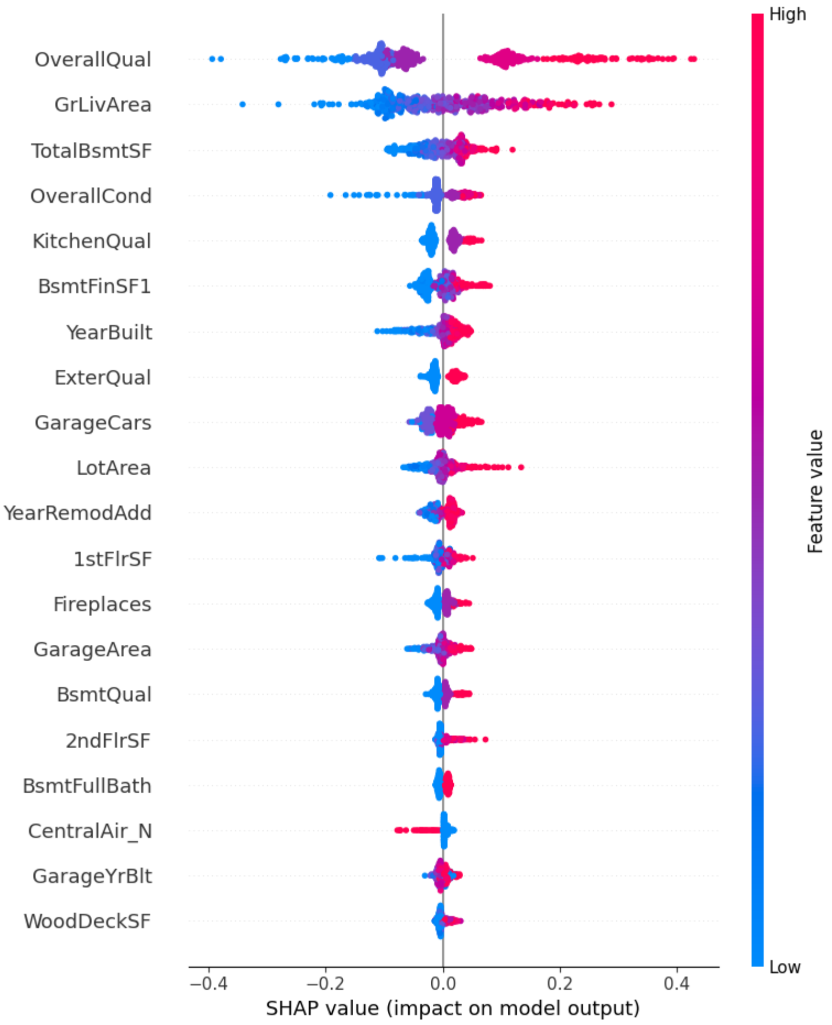

Figure 3: SHAP Summary Plot: Feature Impact on Price.

D Deployment Configuration

D.1 Docker Configuration

```
FROM python:3.9-slim
RUN apt-get update && apt-get install -y libgomp1
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY model_pipeline.pkl .
COPY app.py .
EXPOSE 5000
CMD ["python", "app.py"]
```

D.2 Flask API Request/Response

Request Format (JSON):

```
{
  "OverallQual": 7,
  "GrLivArea": 1710,
  "TotalBsmtSF": 856,
  ...
}
```

Response Format:

```
{
  "predicted_price": 208500.00,
  "predictions": [208500.00]
}
```

E Team Contributions

Vaibhav Mahore:

- Cleaned dataset and built preprocessing pipeline.
- Engineered new features and established baseline model performance.
- Trained ensemble models and performed comparative analysis.
- Implemented stacking strategies.
- Conducted interpretability analysis and drafted project reports.
- (MLOps & Deployment):
  - Integrated MLflow for experiment tracking.
  - Developed Flask API and Docker Deployment Pipeline (With Frontend).
- Finalized project report and code Documentation.

Vismita:

- Produced cleaned dataset and data quality report.
- Developed advanced models, including ensemble methods and neural networks.
- Optimized model performance through hyperparameter tuning and evaluation.
- Led the model evaluation and selection process.
- Built the REST API for model deployment.
- Co-authored project report and presentation.