House Price Prediction Competition

Mallooki | Purdue applied econometrics and machine learning course project | Public work sample brief

This public brief summarizes an applied econometrics and machine learning project on house price prediction. The project asks which modeling choices improve out-of-sample sale price prediction in a dataset with missing values, categorical variables, and nonlinear housing-quality relationships.

OBJECTIVE

- Predict sale prices for a held-out housing dataset using a training set with numeric, categorical, and missing-value problems.
- Compare modeling choices through out-of-sample validation and use the results to improve the final prediction pipeline.

DATA PREPARATION

- Split numeric and categorical variables, then used median imputation for numeric fields and most-frequent imputation for categorical fields.
- Scaled numeric variables and one-hot encoded categorical variables so tree-based models could use the full feature set cleanly.
- Used a log transformation for sale price during training, then transformed predictions back to dollar values for interpretation.

FEATURE ENGINEERING

- Built 47 engineered features, including total square footage, house age, remodel age, total bathrooms, porch space, quality interactions, binary amenity indicators, and neighborhood price tiers.
- Used training-set median sale prices by neighborhood to create a data-driven neighborhood quality encoding.
- Added interaction terms for quality, living area, basement size, garage area, and neighborhood quality.

MODELING AND VALIDATION

- Tested interpretable linear baselines, including OLS with stepwise selection plus Ridge and Lasso grids.
- Moved toward LightGBM and XGBoost after validation favored tree-based gradient boosting for nonlinear relationships.
- Used hyperparameter search, stratified cross-validation, and a 50/50 ensemble of LightGBM and XGBoost.

LIMITS

- The public summary should avoid unsupported metric claims.
- The provided files do not include final score outputs or raw data for independent reproduction.
- The validation design should be described carefully because some tuning and encoding choices happen before fold evaluation.

REPRODUCIBILITY

- The code keeps a fixed random seed and documents package versions, which makes the modeling workflow easier to inspect and rerun when the original data are available.

SKILLS SHOWN

- Python, pandas, NumPy, scikit-learn, LightGBM, XGBoost, imputation, encoding, feature engineering, model tuning, cross-validation, and reproducible modeling documentation.

PUBLIC NOTE

- This is a public brief based on the project. The original class files remain private for privacy reasons.