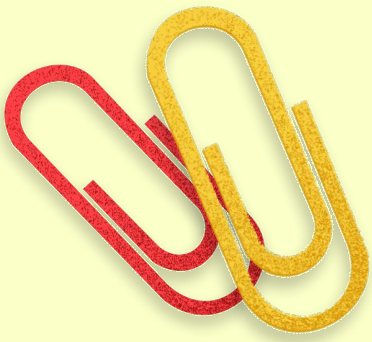
E-CELL
IIT GUWAHATI

UDGAM '26

INTERNFAIR

Your Gateway To Internships

Residential Property Price Prediction Challenge



Problem Statement Release

UDGAM 2.0

E-CELL
IIT GUWAHATI

Table of Content

Problem Statement

Real estate valuation is one of the most critical challenges in the housing market. While most people think the price of a home depends solely on obvious factors like the number of bedrooms, bathrooms, or the presence of a garage, the reality is far more complex. Property prices are influenced by dozens of subtle factors—from the quality of the basement ceiling to the proximity of historical railroads, from the type of roof material to neighborhood-specific characteristics.

Your task is to develop a machine learning model that can accurately predict residential property sale prices based on 79 explanatory variables describing nearly every aspect of homes in a real-world dataset. This is a regression problem where you'll apply advanced data science techniques to understand feature importance, engineer new features, and build robust predictive models.

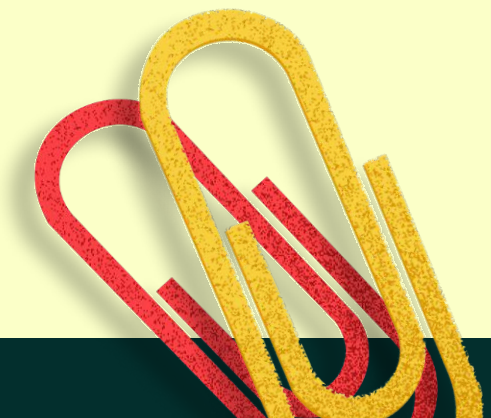
Objective

Given comprehensive information about residential properties, predict the final sale price for each house in the test dataset. Your predictions will be evaluated on how accurately they match actual sale prices, with special consideration given to both expensive and affordable properties.

Learning Outcomes

By solving this problem, you will:

- Perform exploratory data analysis (EDA) on a large, complex real-world dataset
- Identify and handle missing data effectively
- Engineer creative features that improve model performance
- Apply advanced regression techniques including linear regression, regularization methods, decision trees, and ensemble methods
- Evaluate models using appropriate metrics and prevent overfitting
- Build production-ready predictive systems for continuous variables



Dataset Description

The dataset contains residential property information from Ames, Iowa, comprising both a training set (1,460 properties) and a test set (1,459 properties). Each property is described by 79 features capturing architectural, structural, and transactional details.

Files Provided:

- train.csv - Training dataset with 1,460 properties and their corresponding sale prices
- test.csv - Test dataset with 1,459 properties without sale prices
- datadescription.txt - Detailed descriptions of all 79 features
- samplesubmission.csv - Example submission file format

Target Variable

SalePrice: The property's sale price in dollars. This is the continuous variable you need to predict.

Feature Categories:

The 79 features are organized into the following categories:

Property Identification & Classification:

- MSSubClass - Building class (e.g., 2-story, 1946 and newer all brick)
- MSZoning - General zoning classification (Agriculture, Commercial, Floating Village, etc.)
- BldgType - Type of dwelling (Single-family, Townhouse, Duplex, etc.)
- HouseStyle - Architectural style (1-story, 2-story, split-level, etc.)

Lot & Land Features:

- LotFrontage - Linear feet of street connected to property
- LotArea - Lot size in square feet
- LotShape - General shape (Regular, Slightly irregular, Moderately irregular, Irregular)
- LotConfig - Configuration (Inside lot, Corner lot, Cul-de-sac, etc.)
- LandSlope - Slope of property (Gentle, Moderate, Steep)
- LandContour - Flatness of the property (Level, Banked, Hillside, Depression)
- Street - Type of road access (Paved, Gravel)
- Alley - Type of alley access (Gravel, Paved, No Alley)

Neighborhood & Location:

- Neighborhood - Physical location within Ames city limits (25 different neighborhoods)
- Condition1 - Proximity to main road or railroad (Normal, Feedlot, Industrial, etc.)
- Condition2 - Proximity to secondary road or railroad if present
- Utilities - Type of utilities available (All public utilities, No public sewers, etc.)

Structural & Construction Features:

- YearBuilt - Original construction date
- YearRemodAdd - Year of remodel or addition
- RoofStyle - Type of roof (Gable, Hip, Flat, etc.)
- RoofMatl - Roof material (Composition, Metal, Shingles, etc.)
- Exterior1st - Primary exterior covering (Brick, Vinyl siding, Wood siding, etc.)
- Exterior2nd - Secondary exterior covering if more than one material
- MasVnrType - Masonry veneer type (Brick, Stone, Concrete blocks, None)
- MasVnrArea - Masonry veneer area in square feet
- Foundation - Type of foundation (Concrete block, Cinder block, Poured concrete, etc.)

Basement Features:

- BsmtQual - Height of the basement (Excellent 100+, Good 90-99, Average 80-89, Below Average, No Basement)
- BsmtCond - General condition of basement (Good, Average, Fair, Poor, No Basement)
- BsmtExposure - Walkout or garden level basement walls (Good exposure, Average exposure, Minimal exposure, No exposure, No Basement)
- BsmtFinType1 - Quality of basement finished area (Good living quarters, Average living quarters, Below average living quarters, Average rec room, Low quality, Unfinished, No Basement)
- BsmtFinSF1 - Type 1 finished square feet
- BsmtFinType2 - Quality of second finished area if present
- BsmtFinSF2 - Type 2 finished square feet
- BsmtUnfSF - Unfinished square feet of basement area
- TotalBsmtSF - Total square feet of basement area

Floor Features:

- 1stFlrSF - First floor square feet
- 2ndFlrSF - Second floor square feet
- LowQualFinSF - Low quality finished square feet (all floors)
- GrLivArea - Above grade ground living area square feet

Room & Bathroom Features:

- Bedroom - Number of bedrooms above basement level
- Kitchen - Number of kitchens
- KitchenQual - Kitchen quality (Excellent, Good, Average, Fair, Poor)
- TotRmsAbvGrd - Total rooms above grade (does not include bathrooms)
- BsmtFullBath - Basement full bathrooms
- BsmtHalfBath - Basement half bathrooms
- FullBath - Full bathrooms above grade
- HalfBath - Half bathrooms above grade

Heating & Climate Control:

- Heating - Type of heating (Floor furnace, Gas forced warm air furnace, Gas hot water, Gravity furnace, Hot water, Electric, Steam)
- HeatingQC - Heating quality and condition (Excellent, Good, Average, Fair, Poor)
- CentralAir - Central air conditioning (Yes/No)
- Electrical - Electrical system (Standard Circuit Breaker, Fused Box over 200 Amps, Fused Box under 200 Amps, etc.)

Garage Features:

- GarageType - Garage location (Attached, Detached, Built-in, Carport, No Garage)
- GarageYrBlt - Year garage was built
- GarageFinish - Interior finish of garage (Finished, Rough finished, Unfinished, No Garage)
- GarageCars - Size of garage in car capacity
- GarageArea - Size of garage in square feet
- GarageQual - Garage quality (Excellent, Good, Average, Fair, Poor, No Garage)
- GarageCond - Garage condition (Excellent, Good, Average, Fair, Poor, No Garage)
- PavedDrive - Paved driveway (Paved, Partial Pave, Dirt/Gravel)

Functional & Outdoor Features:

- Functional - Home functionality rating (Typical, Minor deductions, Minor deductions 2, Moderate deductions, Major deductions 1, Major deductions 2, Severely deducted, Salvage only)
- WoodDeckSF - Wood deck area in square feet
- OpenPorchSF - Open porch area in square feet
- EnclosedPorch - Enclosed porch area in square feet
- 3SsnPorch - Three season porch area in square feet
- ScreenPorch - Screen porch area in square feet
- Fence - Fence quality (Good privacy, Minimum privacy, Good wood, Minimum wood, No Fence)
- PoolArea - Pool area in square feet
- PoolQC - Pool quality (Excellent, Good, Average, Fair, No Pool)
- MiscFeature - Miscellaneous feature not covered in other categories (Elevator, 2nd kitchen or family room, Shed over 800 sq ft, Tennis court, No miscellaneous feature)
- MiscVal - Value of miscellaneous feature

Exterior & Fireplace Features:

- ExterQual - Exterior material quality (Excellent, Good, Average, Fair, Poor)
- ExterCond - Present condition of exterior material (Excellent, Good, Average, Fair, Poor)
- Fireplaces - Number of fireplaces
- FireplaceQu - Fireplace quality (Excellent, Good, Average, Fair, Poor, No Fireplace)

Transaction Information:

- SaleType - Type of sale (Warranty Deed, Conventional, New, Court Officer Deed, Contract, etc.)
- SaleCondition - Condition of sale (Normal, Abnormal, Adjudged Infirmed, Allocated, Short Sale, Foreclosure)
- MoSold - Month sold (1-12)
- YrSold - Year sold (2006-2010)

Evaluation Metric

Your model's performance will be evaluated using RMSE on a logarithmic scale:
RMSE = sqrt(1/n * sum((ln(yi) - ln(yi_pred))^2))

Where:
- yi is the predicted sale price for property i
- yi_pred is the actual sale price for property i
- n is the number of properties in the test set
- ln denotes the natural logarithm

Why logarithmic scale? Using the logarithm of prices ensures that prediction errors on expensive homes and affordable homes are weighted equally. Without this transformation, a model that predicts expensive houses well but struggles with affordable properties might still appear competitive because larger absolute errors on expensive homes would dominate the metric.

Submission Format

Your predictions must be submitted as a CSV file with exactly two columns in the following format:

Id,SalePrice
1461,169000.1
1462,187724.1233
1463,175221.0
1464,195000.5
...

Requirements:
- Include a header row with column names "Id" and "SalePrice"
- One prediction per line
- Property IDs must match the test set starting from 1461
- Sale prices must be positive numerical values
- The file should contain exactly 1,459 predictions (one for each test property)

Key Challenges & Considerations

1. Missing Data

Some properties have missing values in certain features. You must decide whether to:

- Remove rows or columns with missing data
- Impute missing values using statistical methods
- Create indicator variables for missingness

2. Feature Engineering

Raw features may not directly predict price. Consider:

- Combining related features (e.g., total living area from different components)
- Creating categorical groupings (e.g., age of house from year built)
- Transforming skewed distributions
- Encoding categorical variables appropriately

3. Data Distributions

Different features have different distributions. Some may be normally distributed while others are highly skewed. Appropriate transformations may improve model performance.

4. Multicollinearity

Multiple features may capture similar information. Advanced regression techniques like Ridge Regression (L2) or LASSO (L1) can help address this.

5. Overfitting vs. Underfitting

Balancing model complexity with generalization is crucial. Use techniques like:

- Cross-validation
- Regularization
- Early stopping in ensemble methods
- Validation set evaluation

6. Feature Importance

With 79 features, not all will be equally important. Identifying which features drive price predictions can:

- Simplify your model
- Provide business insights
- Reduce computational cost

Recommended Approaches

Exploratory Data Analysis (EDA):

- Visualize distributions of numerical features
- Analyze correlation between features and target variable
- Identify outliers and missing data patterns
- Create summary statistics by neighborhood or property type

Data Preprocessing:

- Handle missing values systematically
- Encode categorical variables (one-hot encoding, ordinal encoding, target encoding)
- Scale/normalize numerical features if using distance-based models
- Remove or transform outliers if appropriate

Regression Techniques to Explore:

- Linear Regression - Baseline model for understanding feature relationships
- Ridge Regression - Addresses multicollinearity through L2 regularization
- LASSO (L1 Regularization) - Feature selection through regularization
- ElasticNet - Combines Ridge and LASSO penalties
- Random Forest - Ensemble of decision trees capturing non-linear relationships
- Gradient Boosting Machines (XGBoost, LightGBM) - Sequential tree-building for improved performance
- Neural Networks - Deep learning approach for capturing complex patterns
- Ensemble Methods - Combining predictions from multiple models

Model Evaluation:

- Use k-fold cross-validation to assess generalization
- Monitor both training and validation performance to detect overfitting
- Analyze residuals to understand where predictions fail
- Interpret feature importances to validate model behavior

Success Criteria

A successful solution should:

1. Handle all missing data appropriately
2. Engineer at least 5-10 new features with clear rationale
3. Implement at least 3 different regression algorithms
4. Achieve RMSE on log scale in the lower percentiles relative to other submissions
5. Include clear documentation of methodology and assumptions
6. Demonstrate understanding of why certain features are important

Tools & Libraries

Recommended tools for this project:

Python:

- pandas - Data manipulation and analysis
- numpy - Numerical computing
- scikit-learn - Machine learning algorithms
- matplotlib, seaborn - Data visualization
- XGBoost - Gradient boosting

R:

- dplyr - Data manipulation
- ggplot2 - Visualization
- caret - Machine learning framework
- xgboost, randomForest - Regression models

Best Practices

- Keep your data science code organized and reproducible
- Document your assumptions and decisions
- Use version control (Git) for tracking changes
- Create separate notebooks/scripts for different stages of analysis
- Always validate your approach with unseen test data

Success Criteria

A successful solution should:

1. Handle all missing data appropriately
2. Engineer at least 5-10 new features with clear rationale
3. Implement at least 3 different regression algorithms
4. Achieve RMSE on log scale in the lower percentiles relative to other submissions
5. Include clear documentation of methodology and assumptions
6. Demonstrate understanding of why certain features are important

Tools & Libraries

Recommended tools for this project:

Python:

- pandas - Data manipulation and analysis
- numpy - Numerical computing
- scikit-learn - Machine learning algorithms
- matplotlib, seaborn - Data visualization
- XGBoost - Gradient boosting

R:

- dplyr - Data manipulation
- ggplot2 - Visualization
- caret - Machine learning framework
- xgboost, randomForest - Regression models

Best Practices

- Keep your data science code organized and reproducible
- Document your assumptions and decisions
- Use version control (Git) for tracking changes
- Create separate notebooks/scripts for different stages of analysis
- Always validate your approach with unseen test data

Frequently Asked Questions

Q: Can I use pre-trained models or transfer learning?

A: While the focus is on regression techniques for tabular data, you're welcome to explore any legitimate machine learning approach.

Q: Should I use all 79 features?

A: Not necessarily. Feature selection is an important part of the process. Some features may be redundant or add noise.

Q: What if my model performs poorly on the public leaderboard?

A: This is normal. Focus on:

- Checking for data leakage
- Validating your preprocessing steps
- Ensuring your submission format is correct
- Experimenting with different feature engineering approaches

Evaluation Scoring

Your solution will be evaluated based on:

1. Prediction Accuracy (RMSE on log-scale) - 80% weight
2. Code Quality & Documentation Clarity and reproducibility - 10% weight
3. Feature Engineering - Creativity, novel and effective features - 5% weight
4. Model Understanding - Demonstration of knowledge and analysis - 5% weight

Expected Outcomes

Upon completing this challenge, you should be able to:

- Explain the complete machine learning pipeline from data to predictions
- Justify your choice of algorithms and hyperparameters
- Interpret feature importances and model coefficients
- Identify and mitigate common pitfalls like overfitting
- Build and deploy practical regression models for real-world problems

DATASET:

Drive: https://drive.google.com/filed/1uWDwUdDGlwgyu40m18r2rz38eX4o9BAg/view?usp=drive_link