Feature Engineering Assignment

House Price Prediction Dataset

Submitted By

Anik Das

Student ID: 2025EM1100026

Submitted To

BITS Pilani

M.Tech in Data Science & Engineering

Course: Feature Engineering

Table of Contents

1. Introduction

This report documents my work on the Feature Engineering assignment using the Ames Housing dataset. The main goal was to apply various feature engineering techniques to prepare the data for machine learning models that predict house prices.

Feature engineering is one of the most important steps in building effective predictive models. Raw data often needs to be transformed, combined, and refined to extract meaningful patterns that machine learning algorithms can use. Throughout this assignment, I've applied multiple techniques including handling missing values, transforming skewed distributions, encoding categorical variables, creating interaction features, and reducing dimensionality.

2. Dataset Overview

2.1 Initial Data Loading

The Ames Housing dataset contains detailed information about residential properties in Ames, Iowa. I loaded the data and performed an initial exploration to understand its structure and characteristics.

> **Dataset Statistics:**
> • Total Records: 1,460 houses
> • Total Features: 81 columns (before engineering)
> • Target Variable: SalePrice (house sale price in dollars)
> • Feature Types: Mix of numerical and categorical variables

2.2 Dataset Structure

The dataset includes various types of features describing different aspects of the properties:

- **Physical characteristics:** Square footage, number of rooms, lot size, basement area
- **Quality ratings:** Overall quality, kitchen quality, garage quality
- **Temporal features:** Year built, year remodeled, year sold
- **Categorical features:** Neighborhood, building type, roof style, heating type

- **Condition ratings:** Various condition assessments on ordinal scales

3. Data Cleaning & Preprocessing

3.1 Missing Value Analysis

Before doing any feature engineering, I needed to handle missing values. Some missing values are actually meaningful (like missing PoolQC means no pool), while others need to be imputed.

> **Missing Data Summary:**
> • PoolQC: 99.5% missing (mostly means no pool)
> • MiscFeature: 96.3% missing
> • Alley: 93.8% missing
> • Fence: 80.8% missing
> • FireplaceQu: 47.3% missing
> • LotFrontage: 17.7% missing (needs imputation)
> • Several garage-related features: ~5% missing

3.2 Imputation Strategy

I used different strategies depending on the feature type and meaning:

- **Categorical features:** Missing values filled with 'None' when absence is meaningful (e.g., no pool, no garage)
- **Numerical features:** Used median imputation for LotFrontage and other continuous variables
- **Garage features:** Filled with 0 (for year) or 'None' (for quality) when garage doesn't exist

Result: After imputation, the dataset has zero missing values and is ready for feature engineering.

4. Feature Engineering

4.1 Numeric Feature Transformation

Many numerical features in housing data are right-skewed (most houses are in a lower price range, with a few very expensive ones). I applied log transformation to reduce this skewness, which helps machine learning models work better.

I identified features with absolute skewness > 0.5 and applied log(x + 1) transformation to them. The "+1" prevents issues when the value is 0.

> **Transformed Features:**
> Applied log transformation to 59 highly skewed numerical features, including:
> • LotFrontage, LotArea
> • MasVnrArea, BsmtFinSF1, BsmtFinSF2
> • Various square footage measurements
> • SalePrice (target variable)

4.2 Categorical Encoding

Machine learning models need numerical inputs, so I converted all categorical variables to numbers using appropriate encoding techniques:

Ordinal Encoding

For features with natural ordering (like quality ratings: Poor < Fair < Good < Excellent), I used ordinal encoding to preserve the order. This included features like OverallQual, KitchenQual, ExterQual, etc.

One-Hot Encoding

For nominal features without natural ordering (like Neighborhood, RoofStyle, HouseStyle), I used one-hot encoding. This creates binary columns for each category.

Result: After encoding, the dataset expanded from 81 to over 300 features, representing all categorical information in numerical form.

4.3 Polynomial Features

I created interaction features between certain numerical variables to capture non-linear relationships. For example, the interaction between OverallQual and GrLivArea might be important because quality matters more for larger houses.

I selected key features like OverallQual, GrLivArea, GarageCars, and TotalBsmtSF, and generated second-degree polynomial features (including interactions).

4.4 Text-Based Feature Representation

I created composite text features by combining multiple categorical variables. The idea is that properties are naturally described using combinations of characteristics rather than individual attributes.

For example, instead of treating Neighborhood, BldgType, and HouseStyle separately, I combined them into a single text description like "CollgCr_1Fam_2Story" which represents a specific property profile.

> **Text Features Created:**
> • property_description: Combines Neighborhood, BldgType, HouseStyle
> • quality_summary: Combines OverallQual, ExterQual, KitchenQual
> • area_summary: Combines GrLivArea, TotalBsmtSF, GarageArea categories

5. Dimensionality Reduction

5.1 PCA Implementation

After all the feature engineering (especially one-hot encoding), I ended up with a very high-dimensional dataset. To reduce this while retaining most of the information, I applied Principal Component Analysis (PCA).

PCA transforms the features into a smaller set of uncorrelated components that capture the maximum variance in the data. This helps reduce overfitting and computational cost.

5.2 Variance Analysis

I configured PCA to retain 95% of the total variance, which is a common threshold that balances dimensionality reduction with information preservation.

PCA Results:
• Original dimensions: 300+ features
• Reduced dimensions: Approximately 150 components
• Variance retained: 95%
• This represents a ~50% reduction in dimensionality while keeping 95% of the information

6. Student Random Feature

As per the assignment requirements, I generated a random feature based on my student ID (2025EM1100026). I extracted the numeric part (2025 and 1100026), combined them, and used this as a seed for reproducibility.

The random feature was generated using a uniform distribution between 0 and 100, and was included in the dataset as 'student_random_feature'. This feature was also processed through all the feature engineering steps.

> **Random Feature Details:**
> • Seed: 20251100026 (derived from student ID)
> • Distribution: Uniform(0, 100)
> • Added as: student_random_feature column
> • Processed through: scaling, PCA transformation

7. Results & Insights

The complete feature engineering pipeline transformed the raw Ames Housing dataset into a machine learning-ready format. Here are the key outcomes:

Final Dataset Characteristics:
• Clean data with zero missing values
• All features normalized/transformed appropriately

• High-dimensional feature space reduced while retaining 95% variance
• Ready for supervised learning models

Key Insights:

- Housing data is naturally right-skewed; log transformation significantly improved distribution normality
- Quality ratings (OverallQual, KitchenQual) are among the most important features for price prediction
- Location (Neighborhood) plays a crucial role, as shown by the variance in different areas
- Interaction features between quality and size metrics capture important non-linear relationships
- PCA effectively reduced dimensionality from 300+ to ~150 components while preserving most information

8. Conclusion

This assignment provided practical experience in applying various feature engineering techniques to real-world data. The Ames Housing dataset presented typical challenges found in data science projects: missing values, skewed distributions, mixed feature types, and high dimensionality.

By systematically applying data cleaning, transformation, encoding, feature creation, and dimensionality reduction techniques, I successfully prepared the dataset for machine learning models. The engineered features should help models better capture the relationships between house characteristics and sale prices.

The most important takeaway from this project is that thoughtful feature engineering, guided by domain understanding of real estate data, can significantly improve model performance compared to using raw features directly.

Student Declaration

I, Anik Das (Student ID: 2025EM1100026), declare that this assignment is my own work and has been completed as part of the Feature Engineering course requirements for M.Tech in Data Science & Engineering at BITS Pilani.