House Price Prediction Project

Here is a comprehensive report detailing the actual approach, models, and results.

1. Project Overview

The project is a **supervised machine learning regression problem**. The objective is to build a model that accurately predicts the **SalePrice** of residential houses based on various features such as location, size, and quality.

2. Technical Approach

The project followed a standard data science pipeline using Python libraries including **NumPy, Pandas, Matplotlib, Seaborn**, and **Scikit-learn**.

A. Initial Data Inspection

- **Dataset Size**: The training data contains **1,460 rows and 81 columns**.
- **Feature Types**: The data includes a mix of numerical (integers/floats) and categorical (object) features.
- **Missing Values**: Significant missing data was identified in features like `Alley`, `PoolQC`, `Fence`, and `MiscFeature`.

B. Preprocessing and Feature Engineering

- **Target Transformation**: The `SalePrice` was found to be right-skewed. To handle this, a **Log Transformation** was applied using `np.log1p()` to normalize its distribution for better model performance.
- **Handling Missing Values**:
    - Numerical features were imputed with the **median**.
    - Categorical features were imputed with the **mode** (most frequent value).
- **Encoding**: Categorical variables were converted into numerical format using **One-Hot Encoding** (`pd.get_dummies`).
- **Feature Scaling**: **StandardScaler** was used to ensure all numerical features were on a similar scale.

3. Models Evaluated

The notebook explicitly evaluates and compares four different regression models:

1. **Linear Regression**: Used as the baseline model.
2. **Ridge Regression**: A regularized model that adds a penalty to prevent overfitting.
3. **Lasso Regression**: A regularized model that can perform feature selection by shrinking coefficients to zero.
4. **Decision Tree Regressor**: A non-linear model that splits data into branches based on feature values.

4. Model Performance Comparison

The models were compared using the **Root Mean Squared Error (RMSE)** on the log-transformed prices. Lower values indicate better performance.

Model	Cross-Validation RMSE (Log)
Lasso Regression	**0.1388** (Best Performance)
Ridge Regression	**0.1413**
Linear Regression	**0.1553**
Decision Tree	**0.2078**

5. Final Results and Optimization

- **Best Model**: **Lasso Regression** was selected as the final model because it achieved the lowest RMSE.
- **Final Prediction**: After training on the full dataset, the model made predictions on the test set. These log-predictions were converted back to actual prices using the inverse transformation `np.expm1()`.
- **Submission**: The final results were saved to a file named `lasso_house_price_submission.csv`.

6. Key Insights

- **Log Transformation** was critical because the target variable was initially skewed, which often violates the assumptions of linear models.
- **Regularization (Lasso/Ridge)** significantly improved performance over basic Linear Regression by handling the high number of features and preventing overfitting.
- The **Decision Tree** performed the worst, suggesting that for this specific dataset size and feature set, linear models with regularization are more robust than simple tree-based models.