# **Report of Machine Learning Regression Models for Predicting House Sale Prices** By Huihao Xing, Wenyu Chen, Joshua Lim

## **Problem Overview**

The objective of this project is to predict house sale prices based on various features provided in the dataset. The dataset encompasses 81 columns, including the target variable "SalePrice." The goal is to construct a regression model that accurately predicts house prices.

## **Exploratory Data Analysis (EDA)**

### **Data Loading and Inspection**

The analysis began by loading the dataset using the pandas library. The initial dataset contains 81 columns, including the target variable "SalePrice." A quick examination of the data revealed missing values in some columns. A heatmap was used to identify missing values in certain columns and to explore relationships between variables.

### **Data Cleaning and Preprocessing**

Preprocessing focused on handling missing values. Columns with a high percentage of missing values were dropped, and categorical columns were one-hot encoded to prepare the data for machine learning models.

Key Steps:

* Dropped columns with high missing values (missing percentage greater than 45%).
* One-hot encoded categorical columns.

### **Data Visualization**

Exploratory data visualization techniques, pair plots, and correlation matrices were employed to identify significant features.

Key Insights:

* Features like "OverallQual," "GrLivArea," and "GarageCars" exhibit strong correlations with "SalePrice."

## **Model Building**

### **Feature Selection**

After preprocessing, the following features were selected for model training: "GarageArea," "GarageCars," "TotRmsAbvGrd," "GarageYrBlt," "Fireplaces," "BsmtFullBath," "GrLivArea," "TotalBsmtSF," "1stFlrSF," "MasVnrArea," "YearRemodAdd," "YearBuilt," and "OverallQual."

### **Model Training**

Linear regression functioned as our starting point, laying the foundation for comparison within our analysis. In addressing overfitting concerns, lasso regression played a crucial role, leveraging its regularization mechanism by introducing a penalty term to the linear regression objective function. Our focus extended to the exploration of ensemble methods within the context of highly correlated data. Consequently, we proceeded to train five regression models on the dataset:

* Linear Regression
* Lasso Regression
* Bagging Regressor
* Random Forest Regressor
* Gradient Boosting Regressor

## **Model Evaluation**

Models’ performance was evaluated based on R-squared, Mean Absolute Error (MAE), and error ratio.

**1. Performance of Each Model:**

Linear Regression:

* R-squared: 0.77
* MAE (test): 26,531.13
* Error Ratio (Linear Regression): 0.14075809804329203

Lasso Regression:

* R-squared: 0.77
* MAE (test): 26,531.08
* Error Ratio (Lasso Regression): 0.1407578105151361

Bagging Regressor:

* MAE (test): 35,704.56
* MAE (train): 33,246.77
* Error Ratio (Bagging Regressor): 0.18942673978148636

Random Forest Regressor:

* MAE (test): 22,851.70
* MAE (train): 12,805.19
* Error Ratio (Random Forest Regressor): 0.12123726982217185

Gradient Boosting Regressor:

* MAE (test): 21,607.79
* MAE (train): 9,045.27
* Error Ratio (Gradient Boosting Regressor): 0.11463780151343929

### **2. Why Ensemble Models Outperform Except Bagging:**

* Bagging might only be effective if the dataset exhibits significant variability or if the underlying relationships in the data are not captured well by the chosen base models.

### **3. Impact of Feature Selection on Model Performance:**

* Feature selection based on correlation graphs and domain knowledge has likely contributed to better model performance.
* However, overfitting in some ensemble models suggests that the selected features may not be perfectly generalizable.
* Further refinement of feature selection and model hyperparameters could enhance overall performance.

## **Model Selection Based on A Surprising Result**

Best Performing Model: Random Forest Model

The Random Forest Model outperformed other models, demonstrating the lower Mean Absolute Error on the test dataset (22,851.70) and achieving the highest Kaggle score of 0.15769.

This result contrasts with initial our prediction that the Gradient Boosting Model would perform better because of its lowest MAE in the test dataset (21,607.79), and lower rate in error ratio. While the Gradient Boosting Model showed the lowest error ratio during the initial analysis, the Kaggle results did not align with expectations. Consequently, a shift was made to the Random Forest Regressor based on its superior performance on the Kaggle score. Further optimization and hyperparameter tuning can be explored to enhance model performance.

**Code, Libraries, and Data Sources**

Code sources:

* Kaggle notebook (Heatmap and Scatter plots between 'SalePrice' and correlated variables )
* Code Source:https://www.kaggle.com/code/pmarcelino/comprehensive-data-exploration-with-python

Libraries sources:

* Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn.

Data Sources:

* Kaggle Dataset: 'train.csv' and 'test.csv' from House Prices - Advanced Regression Techniques Competition
* Data Source: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data