

# Ames House Price Prediction

Kaggle Data Science Project - Advanced Regression



# What is the problem?

- How much are the residential homes in Ames, Iowa given all aspects/attributes of the properties?
- Useful house price prediction for setting data-driven budget for buyers
- Insightful tool for setting more reasonable prices when selling properties
- Build regression models to leverage attributes of properties to predict price



# Dataset Description

Ames House Dataset with 79 Attributes and Sale Prices of 1,460 Properties

## Feature Examples:



### Exterior Features

- Exterior material quality
- Type of foundation
- Masonry veneer type



### Interior Features

- Heat condition
- Central air conditioning
- Kitchen quality



### Location Features

- Zoning classification
- Slope of property
- Physical location within city limits

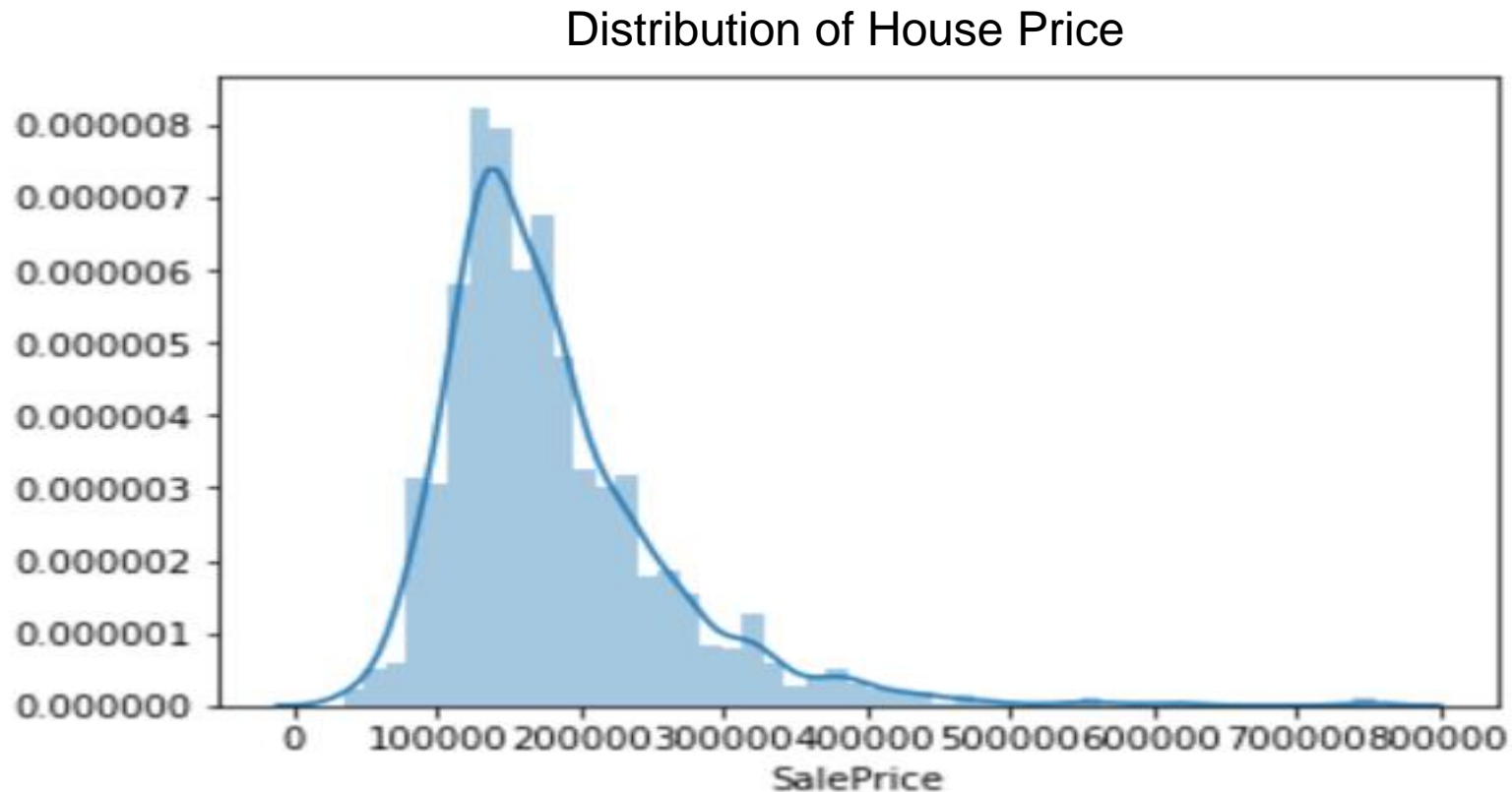


# Exploratory Data Analysis



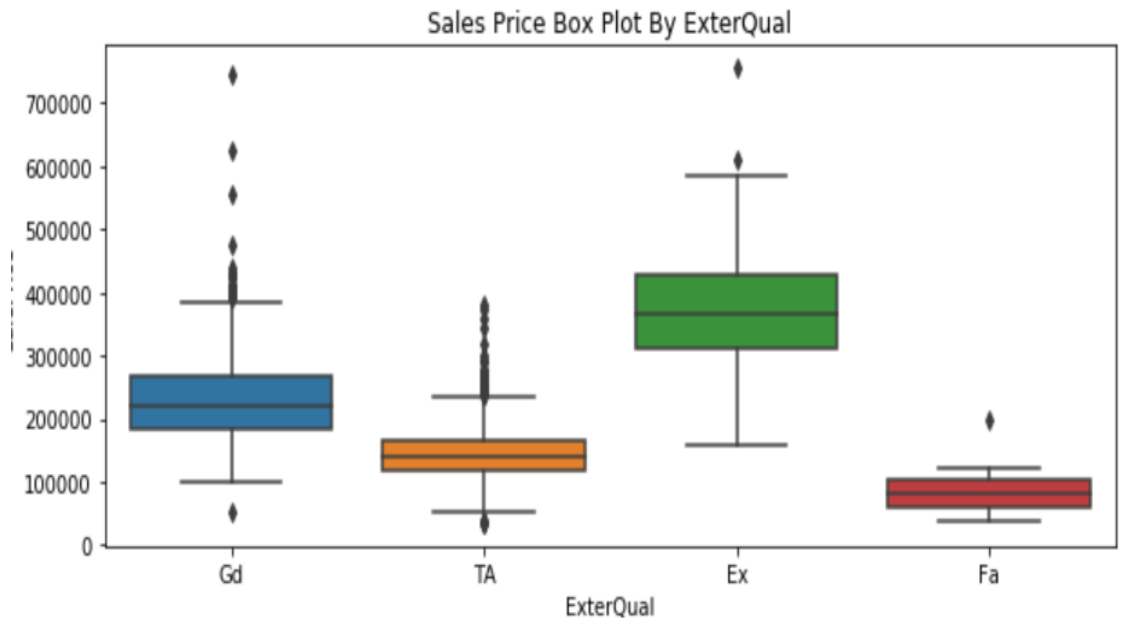
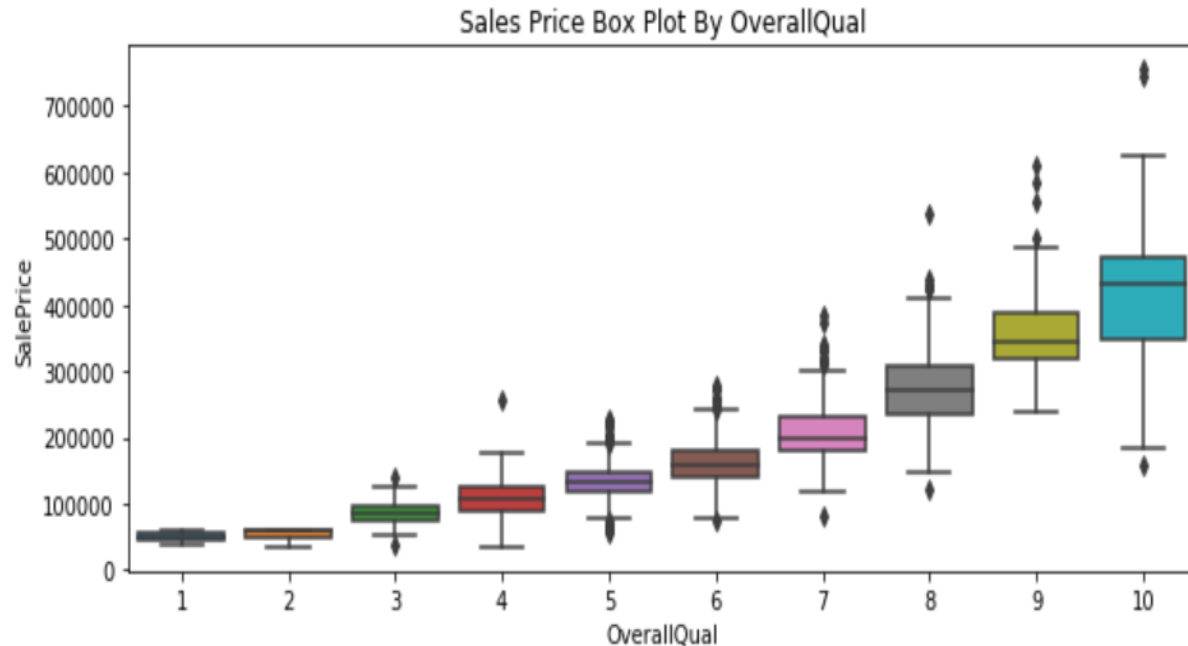
# Price of Properties in Ames, IOWA

- Average house sale price is around \$ 180,000 in Ames.
- There are a few properties over \$500,000. But majority of houses are priced under \$214,000.



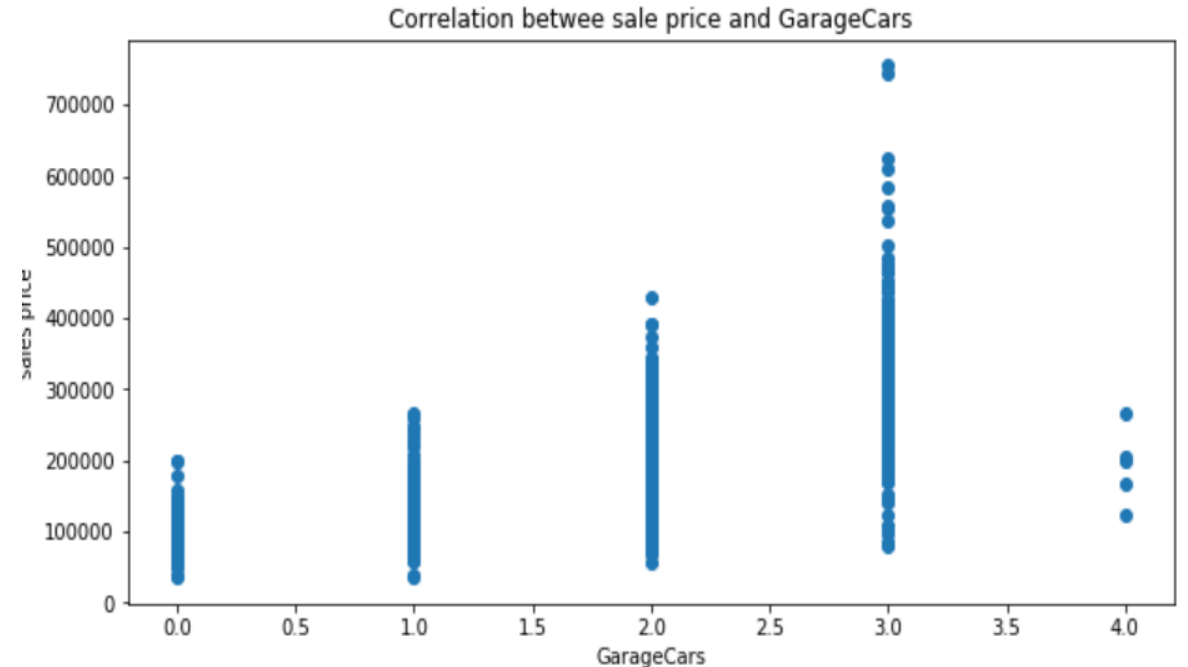
# Attributes Most Correlated with Sale Price

- **Top Significant Categorical Attributes Discovered via ANOVA Test:**
  - **Overall Quality:** The higher the quality score, the higher the sale price.
  - **Exterior Material:** Properties with excellent or good quality tend to be more expensive.
  - **Other significant attributes:** kitchen quality, Height of the basement, Interior finish of the garage, Masonry veneer type, Type of foundation, etc.



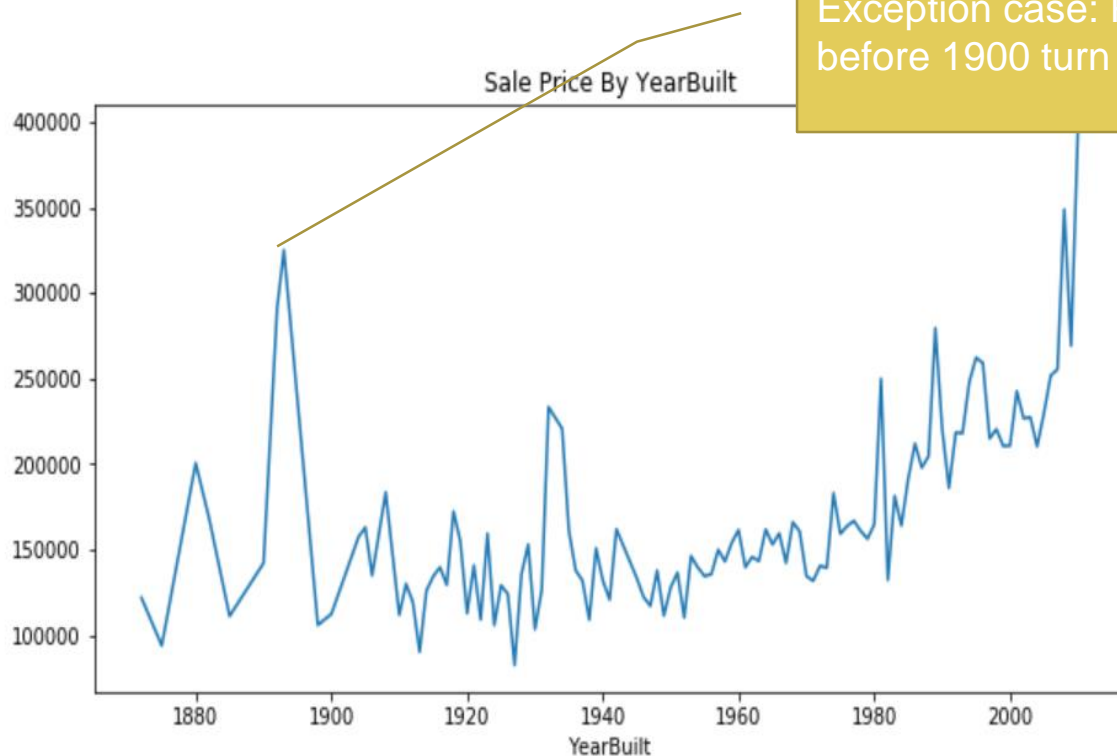
# Attributes Most Correlated with Sale Price

- **Top Correlated Attributes Discovered via Correlation Analysis (R Squared & Scatter Plot) :**
  - Above ground living Area in Square Feet
  - Size of Garage in Car Capacity & Square Feet
  - Total Square Feet of Basement Area
  - First Floor Area in Square Feet



# How Timing Impact Property Price

- In general, the more recent the property was built or remodeled, the more expensive it would be.
- The properties sold during second half of the years were more expensive on average than properties sold in the beginning of the years.



Exception case: Properties built right before 1900 turn out to be expensive.





# **In-Depth Analysis & Modeling**



# Build Base Regression Models

- **Feature Engineering:**
  - Log Transformation on Skewed Features
  - Standard Scaling on all Features
- **Machine Learning Algorithms Used:**
  - Ridge (L2-Norm)
  - Lasso (L1-Norm)
  - Elastic Net (Combination of L1 & L2 Norm)
  - Gradient Boosting Regressor
  - XG Boosting Regressor
  - Light GBM Regressor
- **Model Optimization:**
  - Randomized Search & Grid Search for Hyperparameter Tuning
  - Optimization Metric: RMSE
- **Model Validation:**
  - Validation Metric: RMSE, MAPE & R Squared

Based on the randomized search 156 is the best alpha to use for the Ridget regression.

```
X_train, X_test, y_train, y_test = func_output_training_testing_datasets(cleaned_transformed_data, test_size=0.25)
random_grid={'alpha': np.linspace(start=1, stop=500, num=30)}
estimator=Ridge()
estimator_name='Ridge'
coefficient_df, intercept, rmse_test, mape_test, R_Squared_test=model_sklearn_model_randomizd_search_fitting_output(random_grid,
X_train, y_train, X_test, y_test, estimator, estimator_name)
```

Fitting 3 folds for each of 10 candidates, totalling 30 fits

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 30 out of 30 | elapsed: 5.5s finished
```

Here's the best parameter from the randomized search: {'alpha': 155.86206896551724}  
RMSE on the training set: 0.39393841330416673  
RSquared on the training set: 0.8435023666199838

RMSE on the testing set: 0.37291726075507337  
RSquared on the testing set: 0.8616394722601429  
Mape on the testing set: 0.11415888013607178

**Randomized Search for Ridge Regression**



# Example Base Model Result

## Light GBM Regressor

- **Performance on the testing set:**
  - R2: 87%
  - MAPE: 11%
  - RMSE: 0.36
- Predicted sale price is close to the actual price.



**Note:** RMSE here is calculated based on predicted price and actual price after transformation & scaling.

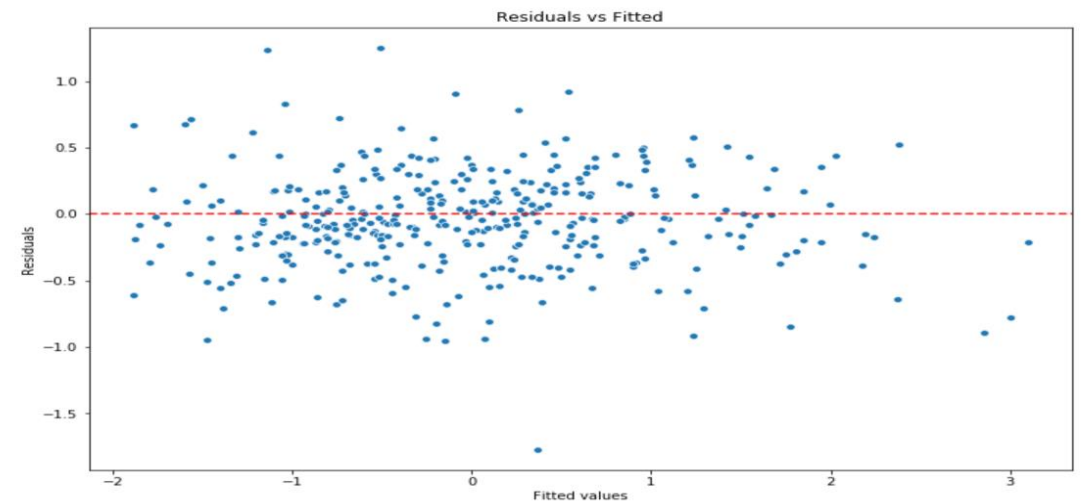
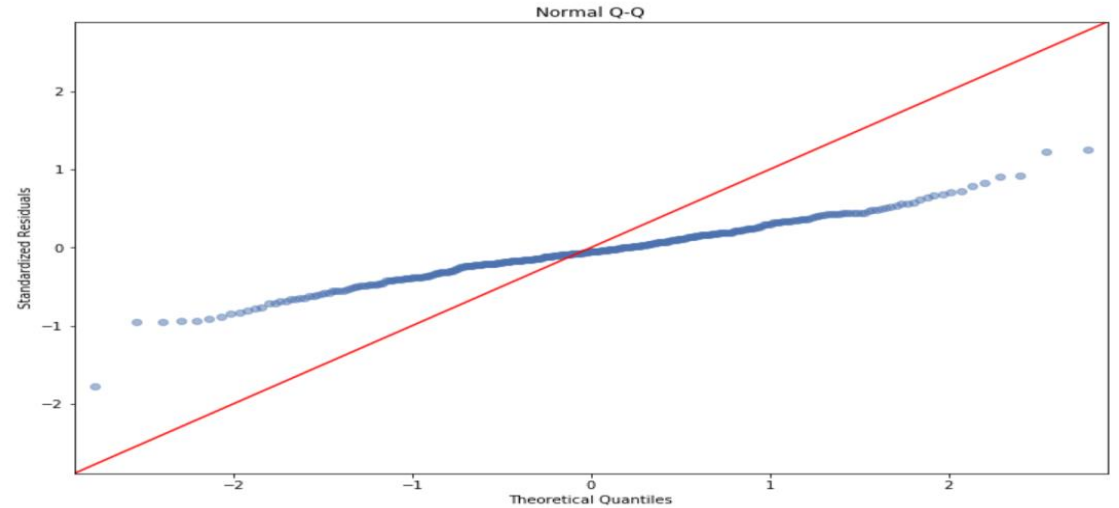


# Example Base Model Result

## Light GBM Regressor

- **Residual Analysis:**

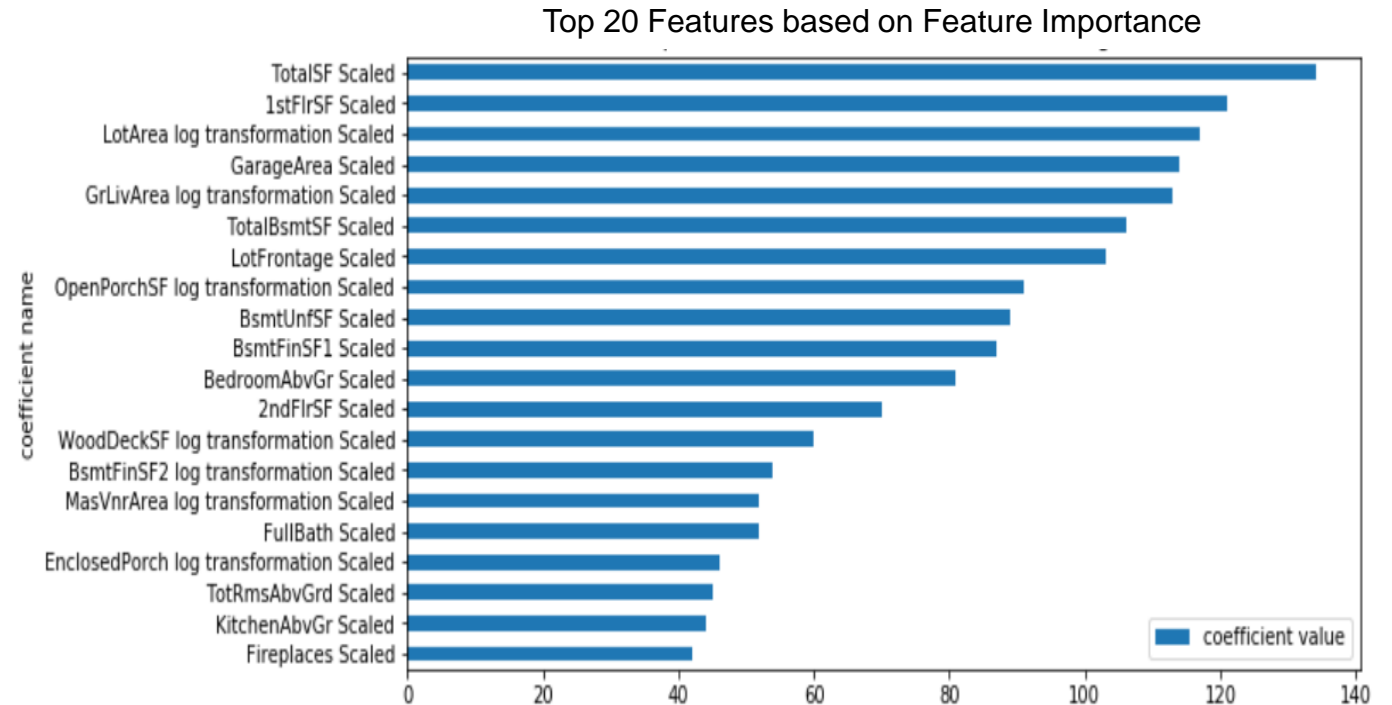
- The model residuals are independently distributed.
- By looking at the QQ plot, the residuals are not normally distributed.



# Example Base Model Result

## Light GBM Regressor

- **Top 20 features:**
  - All the top 20 features for Light GBM are numerical variables.
  - Most of the features are closely related to the area (in square feet).



# Model Accuracy Improvement

- **Pick the most accurate base model:**  
Light GBM here has the lowest RMSE.
- **Stack all the base models by taking the average of the predictions from the base models:**  
Stacked model predictions are more accurate than any single base model.

	model name	RMSE	Actual RMSE	Mape
0	Ridge	0.372920	30706.133666	0.431474
1	Lasso	0.375859	30490.569355	0.434702
2	Elastic Net	0.375070	30627.151324	0.433006
3	Gradient Boosting	0.374879	35395.717683	0.432431
4	Xg Boost	0.376148	39531.299038	0.415521
5	Light gbm	0.361896	34834.651064	0.434717

Mape before transformation and scaling: 0.42875826657812477  
MSE before transformation and scaling: 31285.720650390864



**Note:** RMSE here is calculated based on predicted price and actual price after transformation & scaling. Actual RMSE is calculated based on predicted price and actual price before transformation & scaling.

Performance of the stacked model

# Prediction Application

- **Prediction Tool Demo:**

Use important features as input to output the estimate for sale price.

House Price Prediction Application Demo created via IPyWidget

Input value for features such as first floor area in square feet

basement_...		1750.00
firstfloor_s...		2334.00
abovegrou...		2744.00
size_garag...		2.00
size_garag...		937.00

House price prediction: 162852.82770019144

Output prediction for house price

