Challenge 2: Predicting Housing Prices in Ames, Iowa

Group 13

Our video: https://youtu.be/5pASPfdt2_4

Abstract

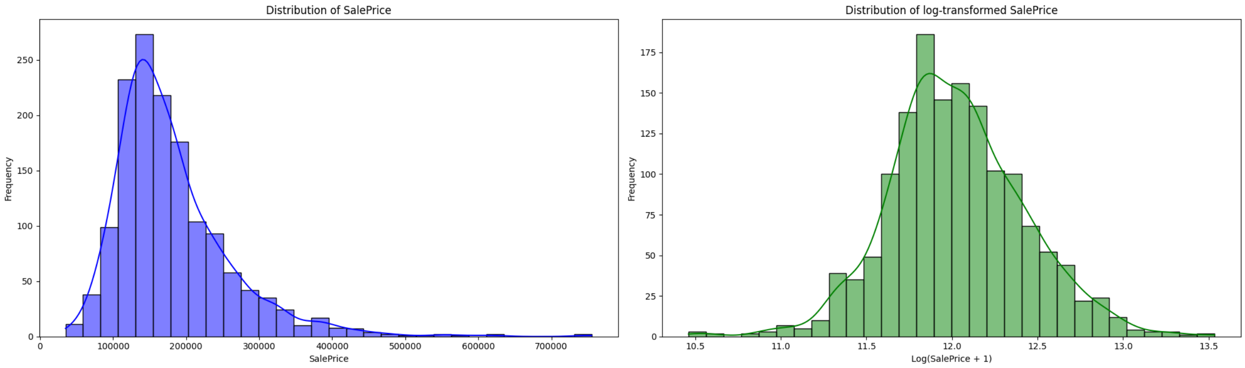
This paper presents a data-driven approach to predict residential housing prices in Ames, Iowa, utilizing the Ames Housing dataset. The dataset contains 79 explanatory variables, covering numerous aspects of residential properties. The primary goal of this study is to forecast the final sale prices of these homes through the application of advanced regression techniques. This paper discusses the data preprocessing, feature engineering, model selection, and performance evaluation.

Introduction

The Ames Housing dataset, meticulously compiled by Dean De Cock, serves as a modernized and extended counterpart to the well-known Boston Housing dataset. The competition challenges data scientists to predict the sale prices of homes based on a diverse set of features. This dataset provides an excellent opportunity to practice creative feature engineering and apply advanced regression techniques, including Random Forest and Gradient Boosting.

Methodology

1. **Data visualization:** Distribution of “SalePrice”



Distribution of SalePrice

Distribution of log-transformed SalePrice

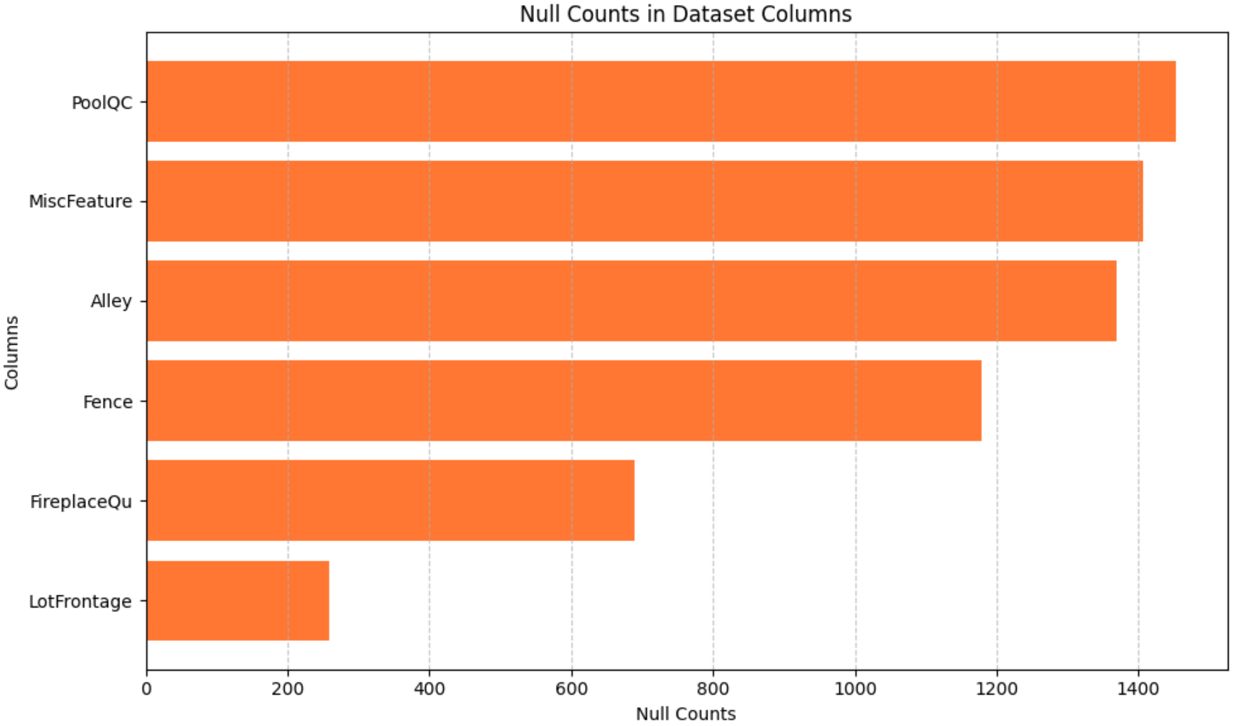
2. Data Preprocessing

The preprocessing phase involved handling missing data, encoding categorical features, and selecting the most relevant features through Recursive Feature Elimination (RFE). Additionally, a crucial step included imputing missing values and scaling the data to ensure compatibility with the chosen regression models.

- **Handle Missing Value:**

  For numerical missing data, we fill these cells with the median of those columns; for categorical (non-number) type data, we replace them with the mode of their columns.

  1. **Visualize missing data and drop high missing rate columns**



Columns such as 'MiscFeature','PoolQC','Fence','Alley', "FireplaceQu" have very high proportions of missing values, therefore, these columns are dropped. Besides, column 'Id' is also dropped.

- **Encode Categorical Columns**

  To realize it, we import LabelEncoder in sklearn.preprocessing and encode categorical columns properly.

- **Impute Missing Value and Normalize Value**

  This is very important, both of them raise compatibility of data and help us build a model successfully.
  We do these functions by importing SimpleImputer from sklearn.impute and StandardScaler from sklearn.preprocessing.

- **Select Feature**

  We realize it with support of **RFE (Recursive Feature Elimination)** provided in sklearn.feature_selection. And finally, we select these 50 columns as our feature columns:

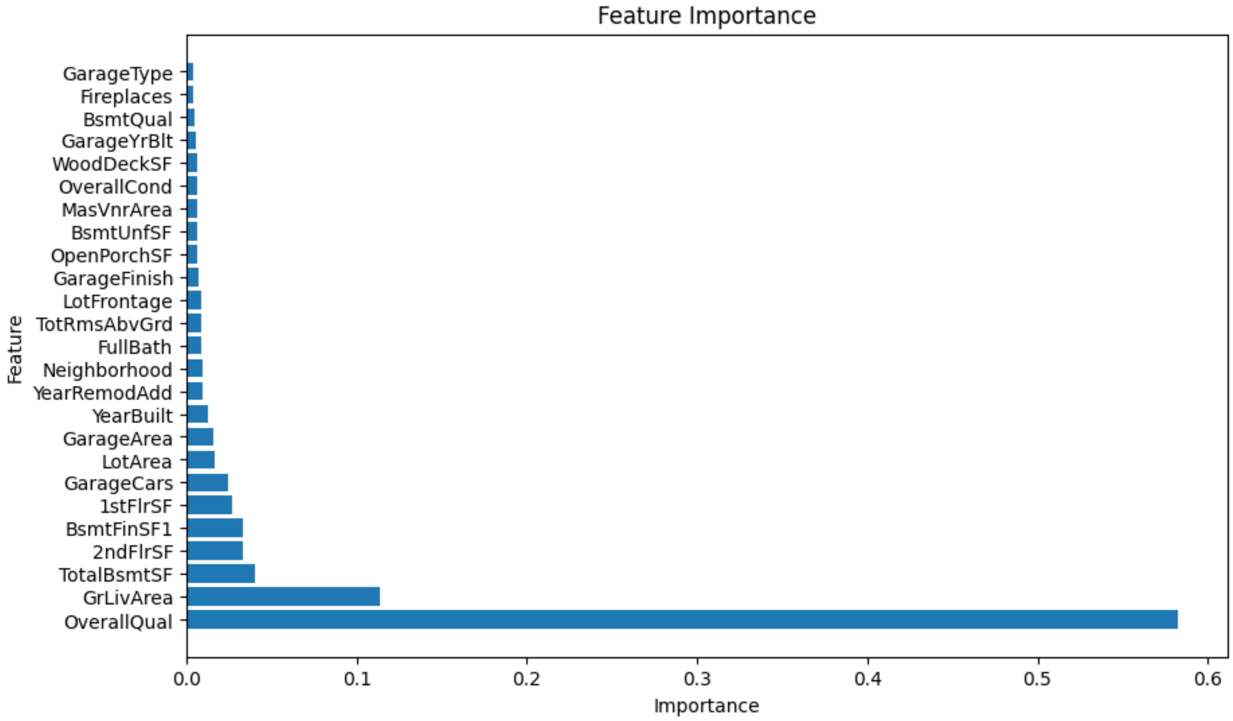
  'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'LotShape', 'LandContour', 'LotConfig', 'Neighborhood', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea', 'ExterQual', 'BsmtQual', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF','TotalBsmtSF', 'HeatingQC', 'CentralAir', '1stFlrSF', '2ndFlrSF', 'GrLivArea', 'BsmtFullBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenQual', 'TotRmsAbvGrd', 'Fireplaces', 'GarageType', 'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', 'ScreenPorch', 'MoSold', 'YrSold', 'SaleType', 'SaleCondition'

3. Model Implementation, Selection and Evaluation

(1) **Random Forest Regressor**: A powerful ensemble method that leverages a multitude of decision trees to make predictions. It is known for its robustness and capability to handle complex relationships in data.

- Implementation:
  - Create a Random Forest Regressor by importing RandomForestRegressor from sklearn.ensemble.
  - Set parameters:
    n_estimators=100, random_state=42.
  - Analyze feature importance: We use the function “feature_importances_” of RandomForestRegressor.

- **Results and discussion:**



  - From the graph, we found that "OverallQual" is the most significant property among them, its importance is Incomparable with others.
  - Second important column is "GrLivArea", its importance is greater than other columns obviously.
  - The rest of features come without any distinctive differences.
  - Unimportant features such as GarageType, Fireplace etc. can be dropped for better accuracy.

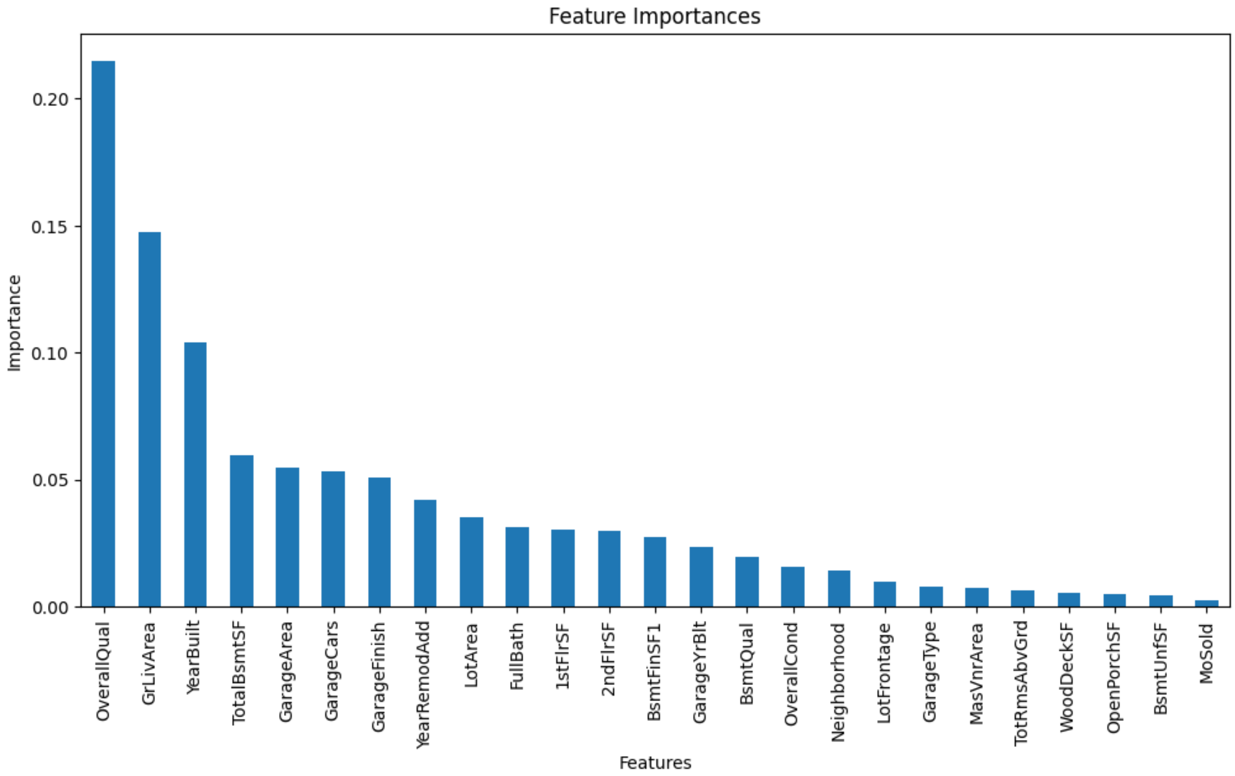
(2)Gradient Boosting Regressor:

Another ensemble technique that builds a strong predictive model by combining the predictions of multiple weaker models. Gradient Boosting is particularly effective in capturing nuanced patterns in the data.

- **Implementation:**
  - Create a Gradient Boosting Regressor by importing GradientBoostingRegressor from sklearn.ensemble.
  - Set parameters n_estimators=3000, learning_rate=0.03, max_depth=5, max_features='sqrt', min_samples_leaf=10, min_samples_split=4, loss='huber', random_state=42.
  - Use the function "feature_importances_" of RandomForestRegressor. With matplotlib.pyplot, we get this graph below:

- **Results and discussion:**

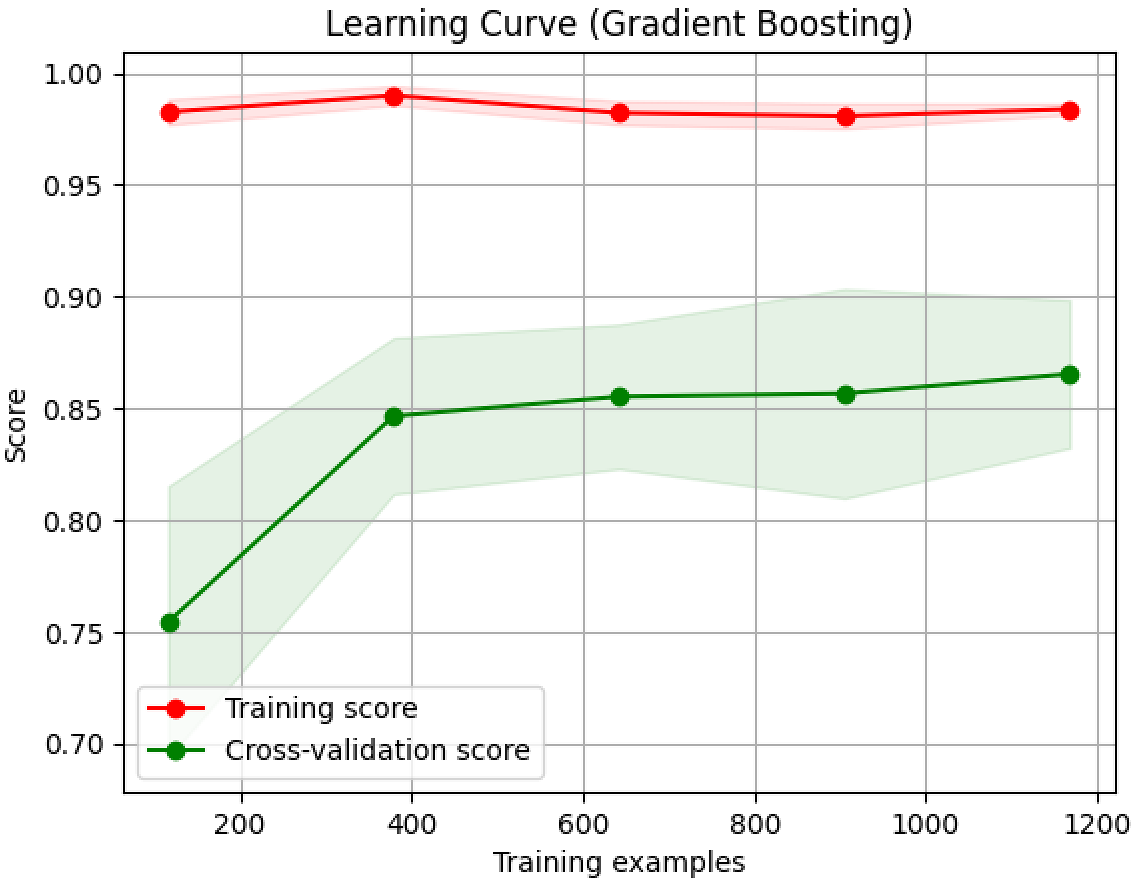
(1)Feature importance bar plot



- From the graph, we find that the ranking of features imortance is similar between Gradient Boosting and Random Forest models. The most important feature is “OverallQual”, second is “GrLivArea”.
- However, different from RandfomForest, in this model, “YearBuilt” show its importance, being more essential than other features.

(2) Learning Curve:

The learning curve demonstrated that the models do not suffer from overfitting, indicating that they are well-generalized for making predictions on unseen data.



4. Prediction Evaluation

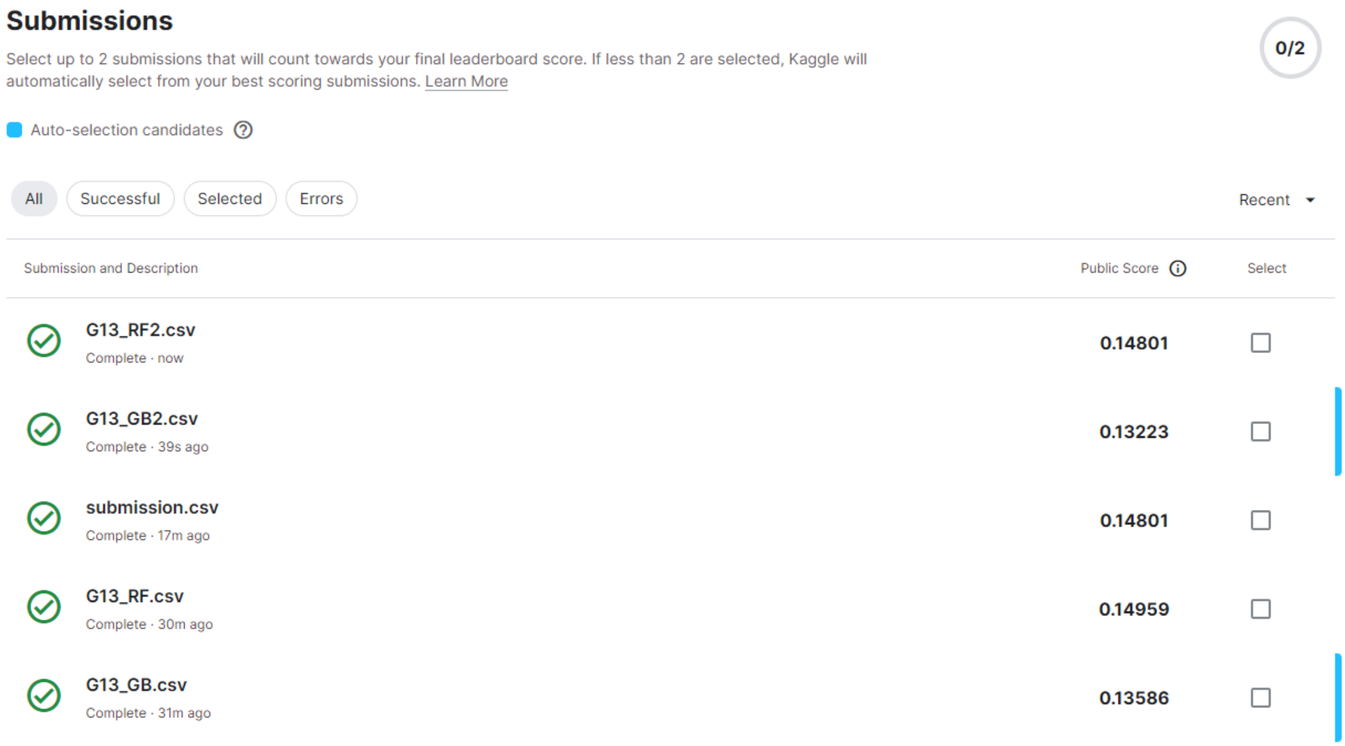
Prediction evaluation was performed using Root-Mean-Squared-Error (RMSE) on a logarithmic scale. This transformation ensured that errors in predicting both expensive and affordable houses were treated equally, emphasizing the overall predictive accuracy. Learning curves were analyzed to assess the models' performance on training and cross-validation datasets.

Results

The key findings of this study are as follows:

1. Both Random Forest and Gradient Boosting Regression demonstrated strong predictive performance.
2. Feature importance analysis revealed which variables had the most substantial impact on housing price predictions. This information can guide future studies and practical applications in the real estate industry.
3. After executing our program, we get two csv files.

One is result predicted by **Random Forest Regression** model, and it got score **0.14801** on Kaggle.
And the other file is result of **Gradient Boosting Regression** model, and it got **0.13223**. From the result of scores, we conclude that **perhaps Gradient Boosting is more suitable for this challenge.**



Submissions

Select up to 2 submissions that will count towards your final leaderboard score. If less than 2 are selected, Kaggle will automatically select from your best scoring submissions. Learn More

Auto-selection candidates

Submission and Description	Public Score	Select
✓ G13_RF2.csv Complete · now	0.14801	□
✓ G13_GB2.csv Complete · 39s ago	0.13223	□
✓ submission.csv Complete · 17m ago	0.14801	□
✓ G13_RF.csv Complete · 30m ago	0.14959	□
✓ G13_GB.csv Complete · 31m ago	0.13586	□

Conclusion

Predicting housing prices is a multifaceted task that requires the application of advanced regression techniques and creative feature engineering. In this study, Random Forest and Gradient Boosting Regressors were able to successfully forecast residential housing prices in Ames, Iowa. The results underscore the importance of feature selection and model evaluation in data science endeavors. Furthermore, the methodologies employed in this analysis can serve as a blueprint for similar predictive tasks in the realm of real estate and beyond.