Springboard – Data Science Career track

# Capstone Project 2

---

# Predicting Housing Prices

Lisa Lorentzon

May 2021

## INTRODUCTION

When buying/selling a house, buyers/sellers and real estate agents need to be able to plan their budget and financing. This project aimed to build models that identify how features of a property impact its value, to help home buyers/sellers and/or real estate agents estimate the final selling price of a property in Ames, Iowa.

During my modeling I found that a Light Gradient Boosting Model (using hyperparameters num_leaves=10, min_child_samples=30, and 72 features) will give the best predictions.

Implementation details can be found in the notebooks listed below in the following Github repository: https://github.com/lilorent/Springboard/tree/master/capstone2-housing

1. Data Wrangling
2. Exploratory Data Analysis
3. Preprocessing – Base Modeling
4. Advanced Modeling

## DATA ACQUISITION & WRANGLING

The datasets used for this project were compiled by Dean De Cock and have been retrieved from https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview. The original dataset consists of 1460 properties with 79 features listed for each property.

After performing initial check of the data (number of features and data points, features, and data types), I determined that the feature "SalePrice" (the final closing price for each property) would be the target feature to use for the modeling. No data was missing for this feature.

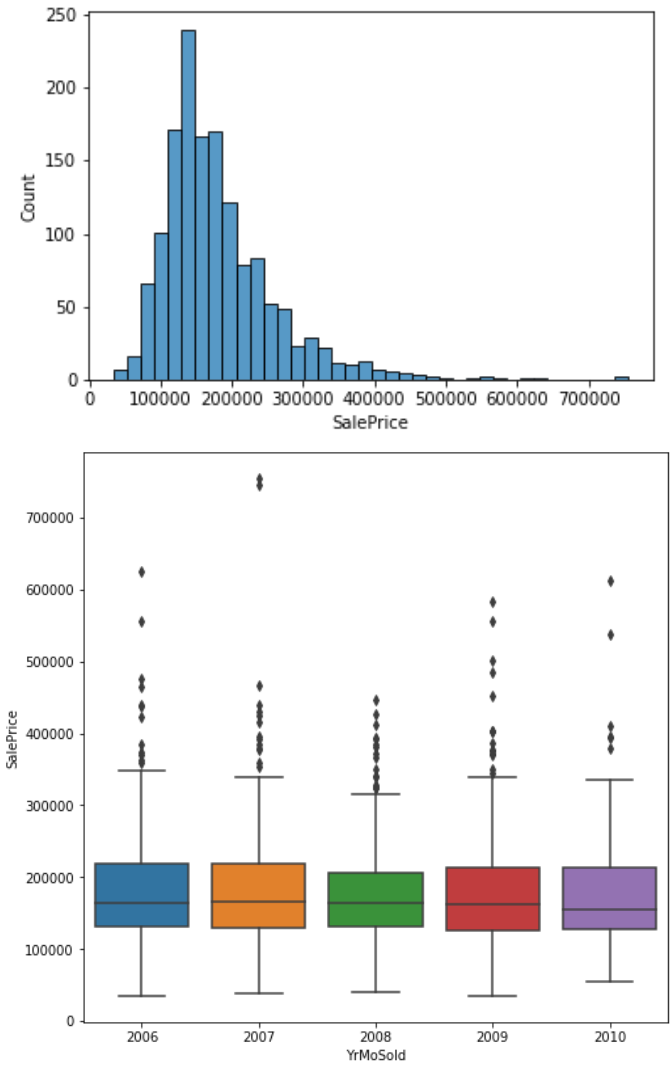
I proceeded to identify all the missing data points and decided to impute the missing values for all object features (except "Electronic") with 'Na' as the data dictionary indicated that these values are not missing, but the properties simply do not have those particular features. I imputed the one missing value for the feature "Electronic" with the most common value ("Standard Breaker") and MasVnrArea with 0. For LotFrontage, I decided against multiple imputation (in favor of time), and went with a simple imputation of the median.

Following an initial visual inspection of the year and month features, I decided to combine the year and month. Later, I decided to drop the month altogether and only keep the selling year – a decision I will discuss further in my commentary section. I also created dummy variables for all categorical features in the data set.

Before moving on to modeling, I calculated the correlation coefficient for all features, and retrieved Feature/Target pairs and Feature/Feature pairs with a correlation coefficient of |0.4| or higher. Here, I found that the features with the highest correlations to SalePrice are Ground Living Area, Total Basement Square Feet, First Floor Square Feet, External Quality (Average), number of Full Baths, Total Rooms Above Ground, and Kitchen Quality (Average).

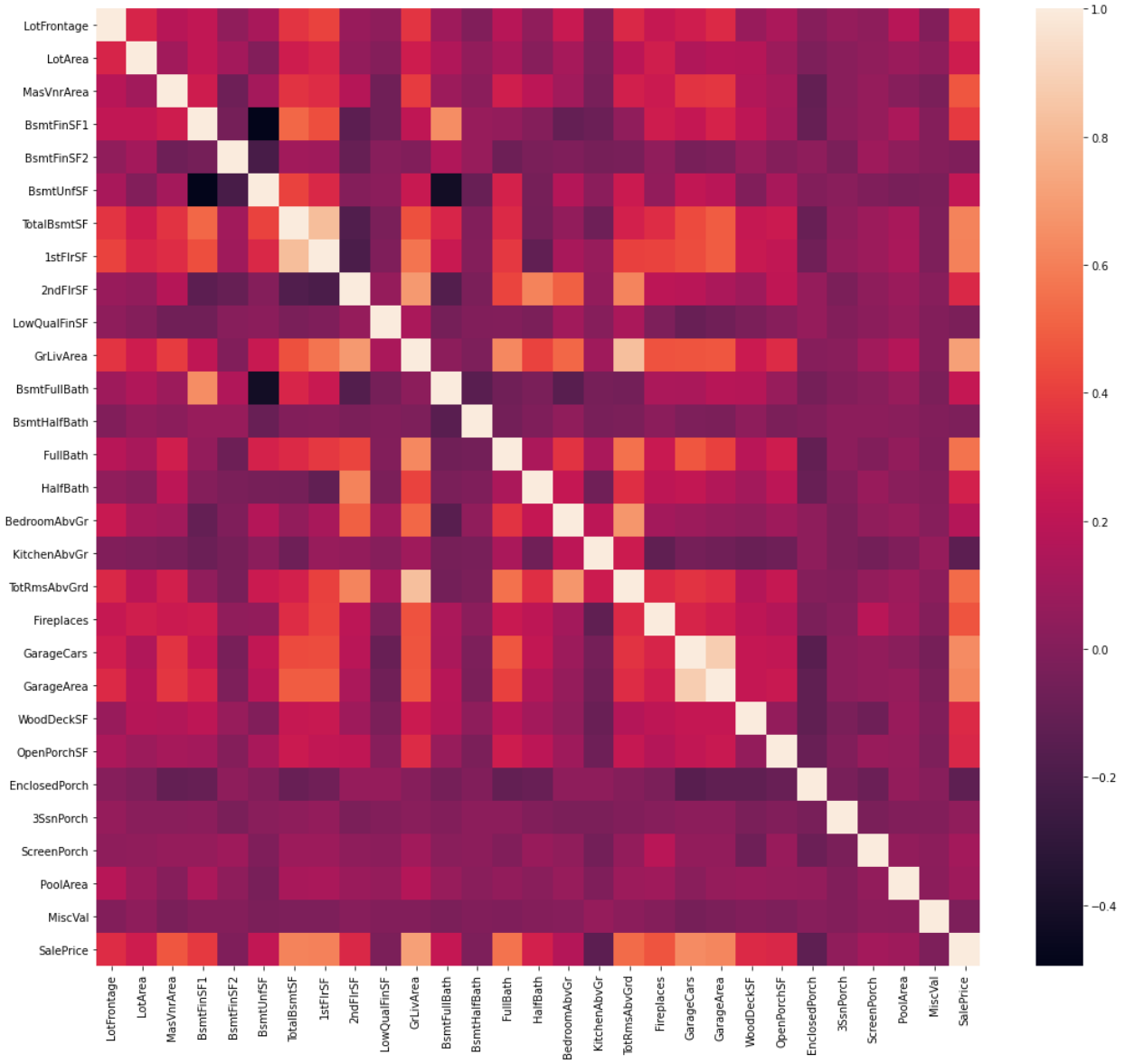
STORYTELLING & INFERENTIAL STATISTICS

To begin with, I looked at the distribution of all the housing prices in my data set as well as the differences for each year.




The housing prices ranged between $34900 and $755000, with a median of $163000 and a standard deviation of $79442.5. The difference in closing prices between the selling years were relatively small, with a slightly lower range in the closing prices for year 2008 (likely due to the economic situation that year).

A first glance at the correlation coefficients between closing price and the continuous variables provided some further insight to possible high-impact features. Visualized in the heatmap, below, I found that the following features are highly correlated (Pearson correlation coefficient of ±0.4 or higher) with the sale price: Ground Living Area, Total Basement Square Feet, First Floor Square Feet, External Quality (Average), number of Full Baths, Total Rooms Above Ground, and Kitchen Quality (Average).




A complete list of all feature correlations can be found in the notebook Exploratory Data Analysis cell numbers 33 and 34.

## BASELINE MODELING

I decided to begin with building three of the most common regression base models for this project: Linear Regression, Ridge Regression, and Lasso Regression. In spirit of building a base model, I kept this step very simple (doing little to no hyperparameter tuning). I evaluated these models on their $R^2$ scores and MAEs (Mean Average Errors).

The Linear Regression base model performed surprisingly poorly for my dataset. It had an $R^2$ score of -5.942855744917795e+18 and a MAE of $26135556089128.79. For the Ridge Regression base model, I needed to finetune the alpha parameter. I built a loop to iterate over alpha values 0.1 to 1.1 in increments of 0.1 and found that the best alpha values for the Ridge Regression model is 0.1. Without any further fine tuning, I built a Ridge Regression model that scored an $R^2$ of 0.8353 and a MAE of $20668.8593. I then repeated the same process for my Lasso Regression model and determined that best alpha for the base model is 0.4. The Lasso Regression had an $R^2$ score of 0.7903 and a MAE of $22843.4313.

In addition to the scores above, I identified the top 3 positive and top 3 negative impact features for the Ridge Regression model: Fence, Exterior Covering, Roof Material (Metal), Roof Material (Standard), Proximity to railroad, Pool; as well as its 95% upper and lower worst-case boundaries: $38546 (upper)/ $182998 (lower). After having built the three base models, I decided to move forward with my Ridge Regression base model. Before continuing,

## EXTENDED MODELING

Once I established my baseline model, I wanted to see if feature selection could improve its performance. I built a loop that iterated over a range of k numbers, created a subset of the top k positive and top k negative features. I, then, used each subset to retrain the model with a Cross-Validation (CV) function and retrieved the MAE and MAPE scores for each of the subsets. This iteration determined that (of the values it iterated over) the best performing set of features was to use the top 143 positive and top 143 negative features. Rebuilding the model with the feature selection made it score an $R^2$ of 0.8428, a MAE of $21112.9928 and a Mean Average Percent Error (MAPE) of 0.1242.

Now that I had established my base model, I was curious to see if ensemble learning models would perform better. I decided to build a Random Forest Regression model and a Light Gradient Boosting Machine (LGBM) and compare their performance with my base model. First, I built a Random Forest Regressor base model. I then performed a Grid Search CV to fine tune the hyperparameters *min_samples_leaf* and *n_estimators*. In addition, I performed feature selection, using the scikit Learn function *SelectFromModel* and found that the optimal features were the top 202 features with highest importance. The final Random Forest Regression model scored an $R^2$ of 0.8269, a MAE of $19113.8267, and a MAPE of 0.1095.

Finally, I built a LGBM base model, performed a Grid Search CV to fine tune the hyperparameters *num_leaves* and *min_child_samples*, and performed feature selection using the *SelectFromModel* function. I found that the optimal values were 10 for *num_leaves* and 30 for *min_child_samples*. The best number of features was 72 (complete list of features can be found in notebook Advanced Modelling, below cell #25).

FINDINGS

| Model | $R^2$ Score | MAE | MAPE |
|---|---|---|---|
| Linear Regression Base | -5.942855744917795e+18 | $26135556089128.79 | - |
| Lasso Regression Base | 0.7903 | $22843.4313 | - |
| Ridge Regression Base | 0.8353 | $20668.8593 | - |
| Ridge Regression FS | 0.8428 | $21112.9928 | 12.42% |
| Random Forest Base | 0.8526 | $18405.5242 | - |
| Random Forest Hyper | 0.8268 | $19127.3626 | - |
| Random Forest FS | 0.8269 | $19113.8267 | 10.95% |
| LGBM Base | 0.8507 | $17327.2001 | - |
| LGBM Hyper | 0.8678 | $16973.4657 | - |
| LGBM FS | 0.8669 | $16921.6723 | 9.3% |

**Base = baseline model, Hyper = fine-tuned hyperparameters, FS = Feature Selection*

When comparing my three base models (Linear Regression, Lasso Regression, and Ridge Regression), the Ridge Regression base model had the overall best performance. The Linear Regression base model performed really poorly (which could be due to the large amount of dummy variables). During extended modeling, the Light Gradient Boosting Machine had a slightly higher $R^2$ score than the fine-tuned and feature selection Ridge Regression model but a significantly better MAE. The Random Forest base model also performed well in comparison to the final Ridge Regression model, but had a slightly worse performance than the LGBM.

CONCLUSION AND FUTURE WORK

Since the goal of the project is to help optimize budget planning/forecasting for property buyers/sellers and real estate agents alike, I will be suggesting moving forward with the final LGBM model (including hyperparameter tuning and feature selection). While it doesn't have the highest $R^2$ score, it does have the lowest MAPE (9.3%)/MAE ($16921.6723) of them all. The model's upper ($54397.84) and lower ($44272.87) worst case scenario boundaries are also reasonable and should be presented to users of this tool.

For future work, I would suggest exploring the following areas:

- Whether/How multiple imputation of missing data could improve the quality of the training data
- Possible seasonal differences in final sale prices
- How removing certain features from the dataset would impact model performance
- Forecasting values over time

RECOMMENDATIONS FOR THE CLIENTS

- Present the upper and lower worst-case boundaries to users and how to apply them
- Use only the selected 72 features as input to model when users select properties

CONSULTED RESOURCES

Packages used:

- Pandas, NumPy, Matplotlib, Seaborn, scikit Learn
- LightGBM
- Pickle

A. Dubey, "Feature Selection Using Random forest," *towards data science*, 14 Dec. 2018, https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f

J. Brownlee, "How to Choose a Feature Selection Method For Machine Learning," *Machine Learning Mastery*, 27 Nov. 2019, https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

J. Brownlee, "How to Develop a Light Gradient Boosted Machine (LightGBM) Ensemble," *Machine Learning Mastery*, 25 Nov. 2020, https://machinelearningmastery.com/light-gradient-boosted-machine-lightgbm-ensemble/

Springboard - Logistic Regression Advanced case study: https://github.com/lilorent/Springboard/blob/master/mini-projects-case-studies/Logistic%20Regression%20Advanced/Logistic%20Regression%20Advanced%20Case%20Study.ipynb

T. Srivastava, "Tuning the parameters of your Random Forest model," *Analytics Vidhya,* 9 Jun. 2021, https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/