# Project 2 - Home Appraiser

Chris Campbell

# Background

Since the onset of the COVID-19 pandemic in 2020, Americans, and millennials in particular, have had to reexamine and re-prioritize their lives. For many, this resulted in an address change. Twenty-six percent of Americans aged 25-40 have relocated during the pandemic, compared to just sixteen percent of the general public.

As hopeful future homeowners and remote workers depart America's largest urban cities in search of more comfortable space and affordable housing, it's important they have knowledge of not just the local housing market, but the key factors in determining a home's market value.

Additionally, appraisers and real estate companies need this insight as they accommodate the increased demand.

# Task at Hand

You've been contracted by **Zillow** to build a ML model that accurately forecasts home prices in **Ames, Iowa** given features of the home, the lot, the neighborhood, etc.

Zillow's plan is to expand this feature nationwide and release it for public use. In the meantime, it will remain in beta while we improve the model and platform.

Your job is to lay the foundation (no pun intended).

# The Data

Our dataset (from Kaggle) includes 80 columns of home attributes. Besides the home's actual sale price, features include:

- Zoning / property type
- Location, Land / Lot attributes
- Variety of home quality / condition scores
- Original build/remodel dates
- Utilities
- ...

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2051 entries, 109 to 10
Data columns (total 80 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   PID              2051 non-null   int64
 1   MS SubClass      2051 non-null   int64
 2   MS Zoning        2051 non-null   object
 3   Lot Frontage     1721 non-null   float64
 4   Lot Area         2051 non-null   int64
                 ...
 76  Mo Sold          2051 non-null   int64
 77  Yr Sold          2051 non-null   int64
 78  Sale Type        2051 non-null   object
 79  SalePrice        2051 non-null   int64
dtypes: float64(11), int64(27), object(42)
memory usage: 1.3+ MB
```

# Cleaning / EDA

- Identify numeric v. non-numeric columns
- Drop columns with over 100 null values
- Impute the mean value for columns with only a few nulls
- Feature engineer
- Assign the cleaned numeric features to a new variable: 'features_df'

# Initial Model Performance

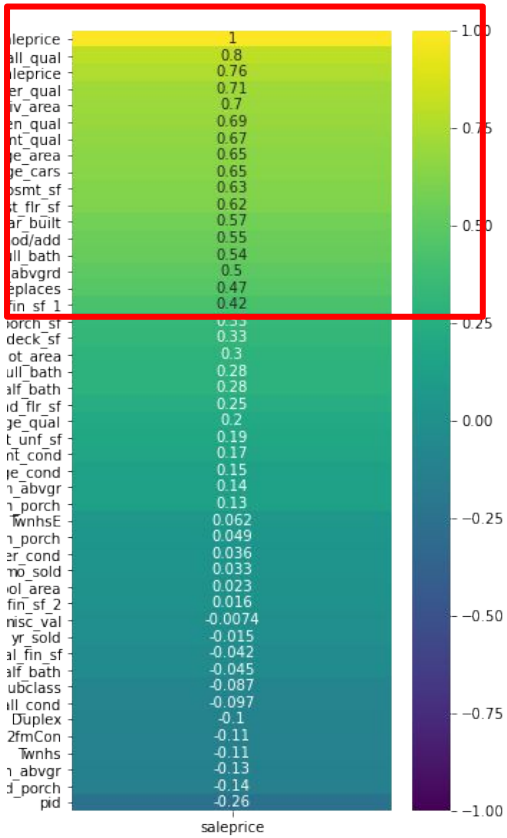
| Model | Cross-val Scores | Conf. Interval | R^2 Scores |
|---|---|---|---|
| Linear Regression | 0.876526155 , 0.847271898, -4.79911021e+11, 0.813626415, 0.510747322 | +- 383928816881.75 | Train: 0.89<br>Test: 0.83 |
| RidgeCV | 0.87476446, 0.85264051, 0.85614752, 0.8200298, 0.54903676 | +- 0.24 | Train: 0.89<br>Test: 0.83 |
| LassoCV | 0.86637163, 0.83751991, 0.87975804, 0.85351365, 0.61710012 | +- 0.20 | Train: 0.89<br>Test: 0.89 |

# Evaluation

- The r-squared train and test scores are consistent across all three models. They're also not terrible. However, the cross-val scores are much more varying.
- In each model's list of cross-val scores there is one that is noticeably different than the others. This indicates outliers in the data that our model cannot account for. The linear regression model highlights this well, as it does not apply regularization like the other two models.
- To account for this we'll reduce the number of features used in the model.

# Feature Reduction

- Function: corr_to_score() evaluated each model's performance based in the correlation of its features to the sale price
  - **0.40 - 0.50** proved the ideal minimum correlation rate each feature should have to sale price.
- After further examining the features that meet that criterion, as well as checking the multicollinearity or our features, we re-ran the models with our newly selected features.



# Model Re-Evaluation

| Model | Cross-val Scores | Conf. Interval | R^2 Scores |
|---|---|---|---|
| Linear Regression | 0.79, 0.86, 0.85, 0.865, 0.75 | +- 0.09 | Train: 0.84<br>Test: 0.84 |
| RidgeCV | 0.875, 0.855, 0.86, 0.83, 0.55 | +- 0.09 | Train: 0.84<br>Test: 0.84 |
| LassoCV | 0.80 0.85 0.84, 0.86, 0.76 | +- 0.08 | Train: 0.83<br>Test: 0.84 |

# Conclusion / Next Steps

- While our final output answered for the overfitting issue we originally noted, r-squared scores lowered slightly and were below the performance benchmark Zillow would need to begin beta testing.
- Based on our second set of scores, I would recommend utilizing the Lasso model as it had low variance and the tightest confidence interval.
- Next Steps for improving model performance will include:
    - Consider whether other non-numeric columns could be encoded (as we did with 'quality' and 'condition' columns)
    - Additional dummying of categorical columns
    - Removing multicollinearity from features to widen the range of features used in the model without overwhelming it.