Fund managing a portfolio of properties in Ames Iowa
Role: Data professional working with a Real estate fund

CHEN TIANCHENG

Context of my project

Problem statement: To build a regression model which predicts property prices in Ames Iowa which is accurate and explainable to non data science folks.

Business will use the model to

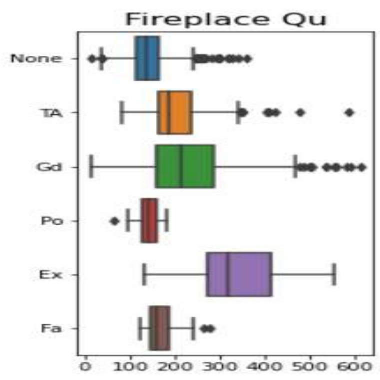
1. Identify the timing to purchase and sell properties
2. Redevelop and renovate properties to unlock the value of their properties
3. Use model as a litmus test versus other more time-consuming methods of valuation

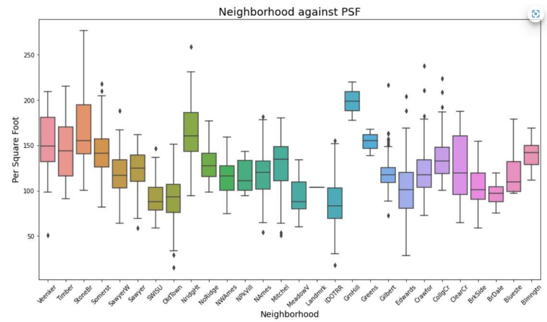
Interviewed a real estate agent to understand valuation methods, to build intuition.

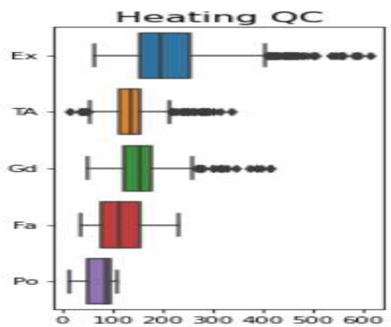
EDA insights

Important features that are found from EDA

1. Location, location, location! 28 Neighborhood
   - Self dividing 28 neighborhood into 5 tranches is less accurate as compared to get dummies
2. Quality and condition related, especially Excellent. Very hard to scale and give a weight.
3. No surprise that square foot related has relationship with Sale Price







Data cleaning and feature engineering

"Art is the reduction of the unnecessary" Pablo Picasso

NaN

Null values replaced with 0 and None mostly via deductive imputation.

Features

Maximum number of features at the start: 282

After feature engineering: 167

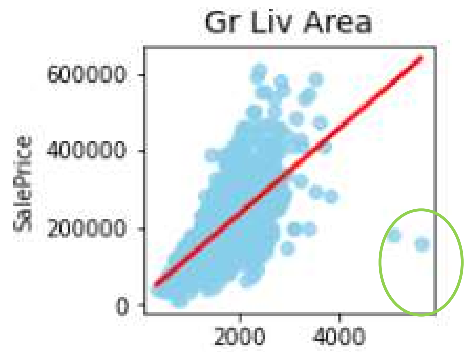
After Lasso (number of non-zero coefficients): 119

Data cleaning and feature engineering

Outliers managed

ID 1499 and 2181

Gross living area more than 4k, low price



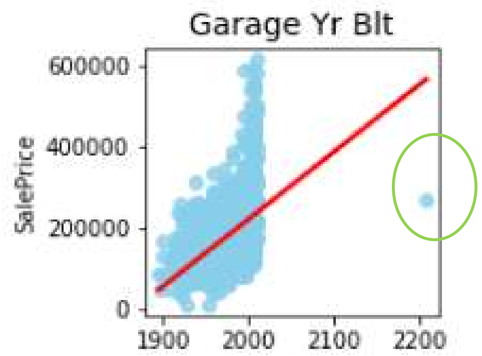
ID 1183 –

Cleared upon investigation for residual plot

Potential limitation in model

ID 1699 –

Garage Built in 2207, change to 2007



Feature engineering

1) Correlation analysis - Pairwise correlation
2) Correlation with target
3) Variance analysis - to drop low variation features
4) Backward elimination (RFE)
5) Multicollinearity
6) Lasso
7) Intuition/ EDA on irrelevant and redundant features

Feature engineering

1) Correlation analysis - Pairwise correlation

2) Correlation with target

Pairwise correlation of 1, drop 1 of the values

For high pairwise correlation,

drop lower correlation feature.

Dropped 29 features

	v1	v2	pair_corr	v1_y_corr	v2_y_corr
0	Central Air_N	Central Air_Y	1.000000	-0.277493	0.277493
1	Garage Qual_None	Garage Cond_None	1.000000	-0.230954	-0.230954
2	Garage Finish_None	Garage Qual_None	1.000000	-0.230954	-0.230954
3	Garage Finish_None	Garage Cond_None	1.000000	-0.230954	-0.230954
4	Street_Grvl	Street_Pave	1.000000	-0.069864	0.069864
5	Bldg Type_Duplex	MS SubClass_90	1.000000	-0.103759	-0.103759
6	Garage Yr Blt	Garage Qual_None	0.998579	0.258554	-0.230954
7	Garage Yr Blt	Garage Finish_None	0.998579	0.258554	-0.230954
8	Garage Yr Blt	Garage Cond_None	0.998579	0.258554	-0.230954
9	Exterior 1st_CemntBd	Exterior 2nd_CmentBd	0.988254	0.168285	0.157714
10	Bldg Type_2fmCon	MS SubClass_190	0.977761	-0.111478	-0.109317
11	Exterior 1st_VinylSd	Exterior 2nd_VinylSd	0.977551	0.342072	0.337486
12	Exterior 1st_MetalSd	Exterior 2nd_MetalSd	0.976454	-0.150017	-0.139501
13	House Style_SLvl	MS SubClass_80	0.954549	-0.042170	-0.031485
14	Roof Style_Gable	Roof Style_Hip	0.949635	-0.250635	0.265941
15	House Style_1.5Fin	MS SubClass_50	0.942502	-0.195938	-0.182463
16	Garage Cars	Garage Area	0.897174	0.648969	0.656008
17	Exter Qual_Gd	Exter Qual_TA	0.895227	0.447221	-0.601468
18	Exterior 1st_HdBoard	Exterior 2nd_HdBoard	0.885850	-0.114482	-0.102602
19	MS Zoning_FV	Neighborhood_Somerst	0.874843	0.106634	0.150013

Feature engineering

3) Variance analysis - to drop low variation features

(<0.009 variance features dropped)

Dropped 85 features

```
#Sort variance and mask data
low_variance = concat_df.var().sort_values(ascending=False)
low_variance = low_variance[low_variance.values < 0.009]
```

```
# Drop low variance features (var<0.009)
low_var_drop_list = [i for i in low_variance.index]
concat_df = concat_df.drop(low_var_drop_list, axis=1)
```

4) Backward elimination (RFE)

Removes the weakest feature (or features) until the specified number of features is reached. CV optimizes the number to get lowest scoring

Dropped 2 features

```
features = [col for col in housing._get_numeric_data().columns if col !='SalePrice']
features
X = housing[features]
y = housing['SalePrice']
```

```
from sklearn.feature_selection import RFECV
selector = RFECV(estimator=LinearRegression(), cv=20, scoring = 'neg_mean_squared_error')
selector.fit(housing.loc[:, housing.columns != 'SalePrice'], housing['SalePrice'])
print('Optimal number of features: %d'
 % selector.n_features_)
```

```
Optimal number of features: 166
```

RFECV has optimised the features to be 165. As such, 2 weakest features are identified which should increase the negative mean squared error score.

```
# Checking the column names which are selected
final_column =  list(housing.loc[:, housing.columns != 'SalePrice'].columns[selector.support_])
```

Model used

RMSE	Linear Regression	Ridge	Lasso
Train dataset	23,095	22,881	**22,874**
Holdout dataset	21,067	21,132	**21,041**
Estimate on unseen dataset	**20,195**	20,447	20,750
Kaggle score on chosen model	NA	NA	**19,459**

- All 3 models competitive

- Lasso chosen as least amount of features,

- Most consistent in train and holdout RMSE score.

Hyperparameter: Alpha of 238

Number of features left: 119, others use 167

Dropped 48 features

What I learn (wish I knew 2 weeks ago)

1. Kitchen sink method don’t work without data cleaning
   - Story of my first model which was a joke
2. A Practical Guide to Dimensionality Reduction Techniques
   https://www.youtube.com/watch?v=ioXKxulmwVQ
3. Recursive feature elimination
   https://www.linkedin.com/pulse/what-recursive-feature-elimination-amit-mittal
4. Feature selection is an iterative process, think, amend, rinse, repeat
5. Tradeoff
   - I could already hit 20k RMSE with 80 features. But to get 19k, I ended with 119. Is it worth it?

Areas of future work

1) Explore interaction, polynomial and log features if accuracy is not low enough for management.

2) Explore alternative feature engineering tools

3) Update of information such as mortgage foreclosure, arms length sales transaction or not

4) Future update of information