Ames Housing Project

By: Team Lollapalooza

Problem Statement

- Our client, a real estate consultant based in Ames, Iowa, provides their clients with building recommendations to increase the value of their clients' properties.
- They have approached us in hopes of applying a data-driven approach to improve their methodology.
- Breaking this request into two parts:
  - 1. How can we reliably predict property prices?
  - 2. What are the most impactful features can we improve or add, to increase the value of the property?

Workflow

1. Data cleaning
    a. Numericization
    b. Removal of outliers
    c. Inclusion / exclusion of variables
2. Exploratory Data Analysis
    a. Data visualization
    b. Feature selection
    c. Feature engineering
3. Model evaluation
    a. Baseline model
    b. Comparative evaluation
4. Identify features that provides greatest increase in property values.
    a. Coefficient comparison
5. Conclusion & Recommendation

Data Cleaning

```python
# all can change to 0
train[cont_vars_na] = train[cont_vars_na].fillna(0)
```

```python
# all can change to NA
train[ordinal_vars_na] = train[ordinal_vars_na].fillna('NA')
```

```python
# all change to NA except Mas Vnr Type to None, 4 vars with null values
for var in nominal_vars_na:
    if var == 'Mas Vnr Type':
        train[var] = train[var].fillna('None')
    train[var] = train[var].fillna('NA')
```

Fig.1. Code snippet that impute missing values

- Impute the missing values for the continuous, nominal and ordinal features

```python
# Overall Qual, Overall Cond already in int datatype
# so only need to convert for the rest of ordinal variables
# NA is assigned to 0
def convert_ordinal_features(df, features):
    for feature in features:
        if feature == 'Lot Shape':
            df[feature] = df[feature].map({'IR3':1, 'IR2':2, 'IR1':3,'Reg':4})
        elif feature == 'Exter Qual' or feature == 'Heating QC' or feature == 'Kitchen Qual':
            df[feature] = df[feature].map({'Po':1, 'Fa':2, 'TA':3,'Gd':4, 'Ex':5})
        elif feature == 'Bsmt Qual' or feature == 'Fireplace Qu':
            df[feature] = df[feature].map({'NA':0, 'Po':1, 'Fa':2, 'TA':3,'Gd':4, 'Ex':5})
        elif feature == 'Bsmt Exposure':
            df[feature] = df[feature].map({'NA':0, 'No':1, 'Mn':2, 'Av':3,'Gd':4})
        elif feature == 'BsmtFin Type 1':
            df[feature] = df[feature].map({'NA':0, 'Unf':1, 'LwQ':2, 'Rec':3,'BLQ':4, 'ALQ':5, 'GLQ':6})
        elif feature == 'Garage Finish':
            df[feature] = df[feature].map({'NA':0, 'Unf':1, 'RFn':2, 'Fin':3})
```

Fig.2. Code snippet that convert values of ordinal features to numerical

- Convert value of ordinal features to numerical values

Data Cleaning: Numericizing Values

Old Column Data	New Column Data	Description
Ex	5	Excellent
Gd	4	Good
TA	3	Average/Typical
Fa	2	Fair
Po	1	Poor
nan	0	No (Feature)

Old Column Data	New Column Data	Description (Home Functionality)
Typ	8	Typical Functionality
Min1	7	Minor Deductions 1
Min2	6	Minor Deductions 2
Mod	5	Moderate Deductions
Maj1	4	Major Deductions 1
Maj2	3	Major Deductions 2
Sev	2	Severely Damaged
Sal	1	Salvage only

Before:

1. Values have ordinal nature
2. Values makes sense to humans who know English
3. Some values missing

After:

1. Values retain ordinal nature
2. Values makes sense to computers
3. Missing value replaced with ‘0’ to depict an absence of the variable

Limitation: Fair isn’t necessarily twice as impactful as Poor

Data Cleaning: Single Column Dummification

No. of Observation	Old Column Data	New Column Data	Description (Misc. Feature)
56	Shed	1	Shed (100+ sqft)
4	Gar2	1	2nd Garage
3	Othr	1	Others
1	TenC	1	Tennis Court
0	Elev	1	Elevator
1,987	nan	0	No Misc. Feature

Proportion of observations with Misc. Features: 3.1%

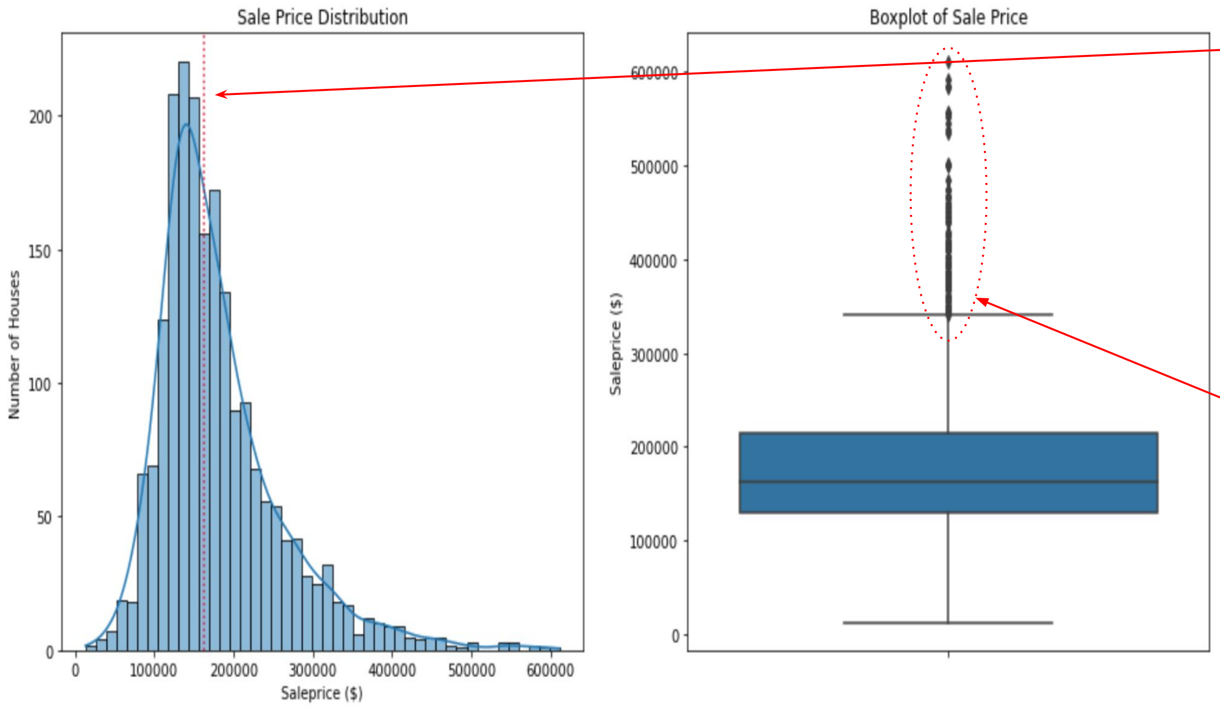
Pros:

1. Better categorization
2. Retention of variable

Cons:

1. 'Surprise' miscellaneous feature
2. May not be useful in either form

EDA: House Sale Prices in Ames

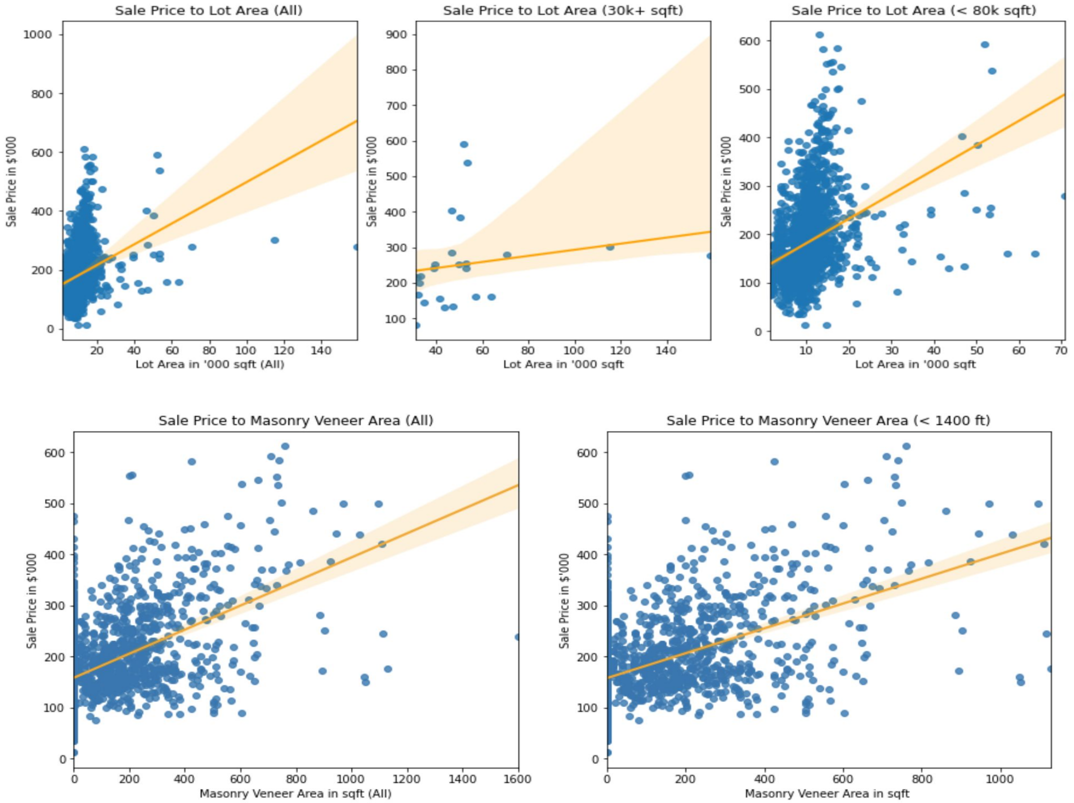


- Median Price = $162,500
- Right-Skewed
- 61 Outliers

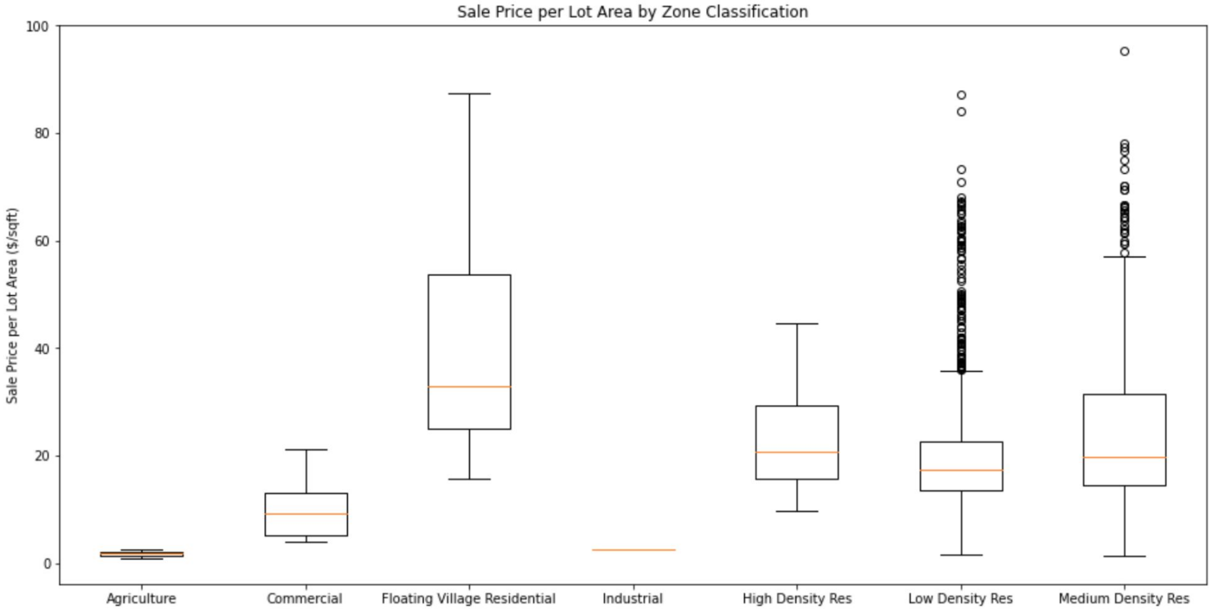
EDA: Crucial Removal of Outliers

With outliers

Without outliers

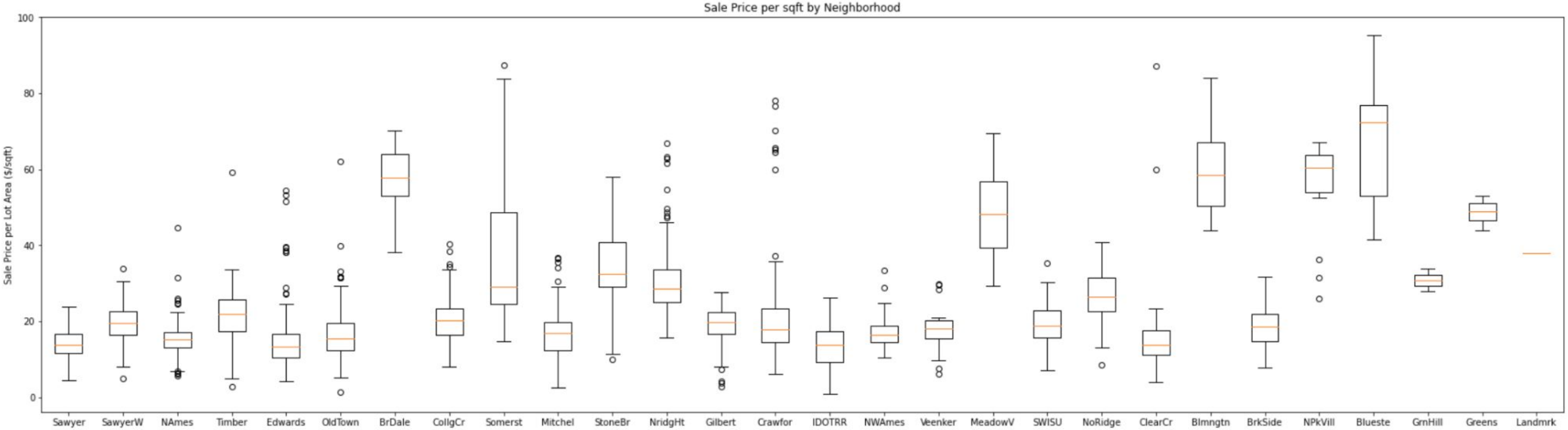


Observation 1: Zoning Matters!



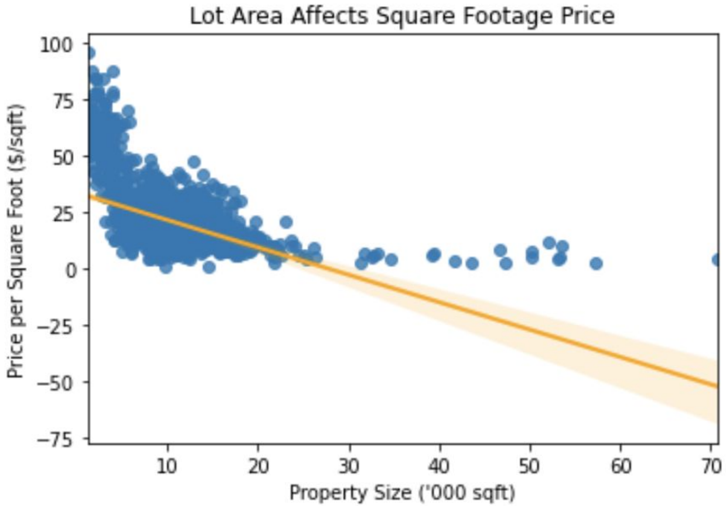
- Property Zones are based on city plans
- Land appropriated for housing purposes are valued higher

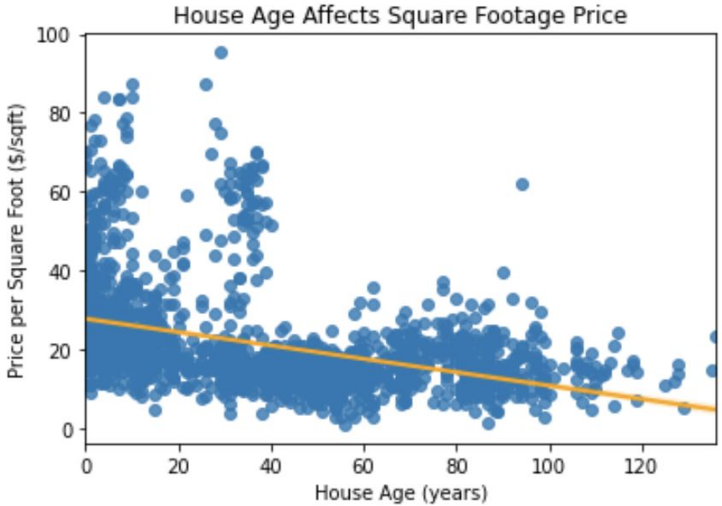
Observation 2: Neighborhood Matters Too!



- The **neighborhood** a property belongs to plays a significant role in the sale price

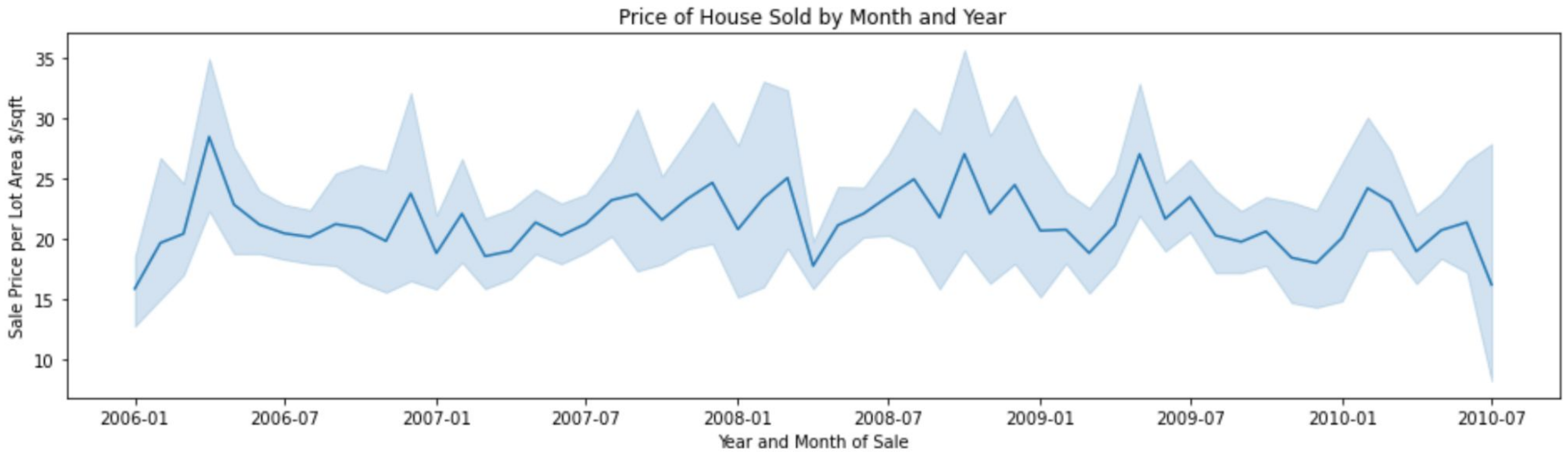
Some Variables are Negatively Correlated





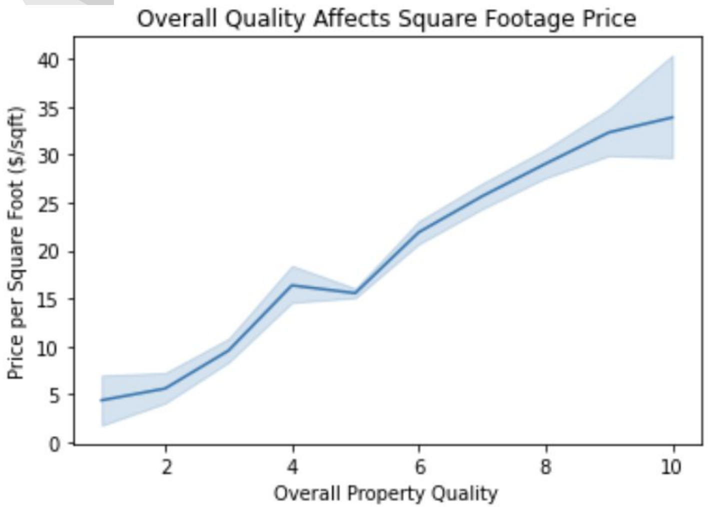
- Negatively correlated variables are also present
- Smaller units like shoebox units in expensive areas can command an inherent premium
- Older houses are generally cheaper

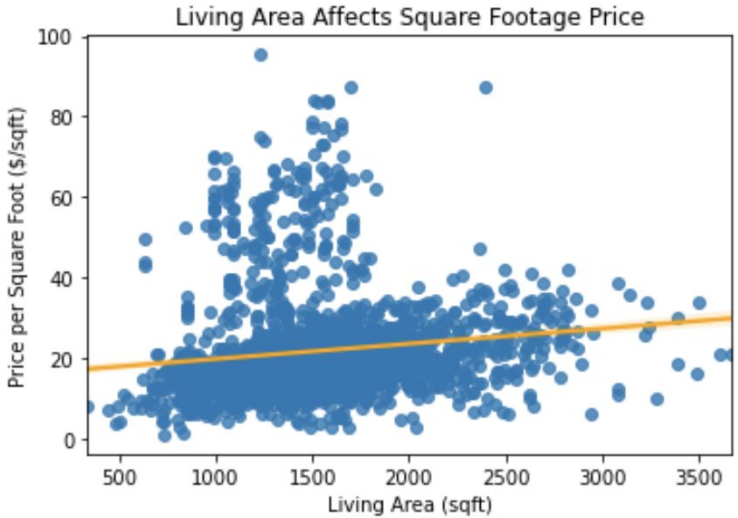
Observation: Volatility in Sale Price



- We observe volatility in sale price per lot area of properties but there is no distinguishable trend
- This debunks the myth that ‘house prices always rise’
- Seasonality of sale price per square footage may not be a reliable indicator

Most Variables are Positively Correlated





- Most variables are positively correlated, eg. higher quality or size of property feature -> higher property value / sqft
- However, some variables are more strongly correlated than others

Feature Selection – Continuous/Discrete

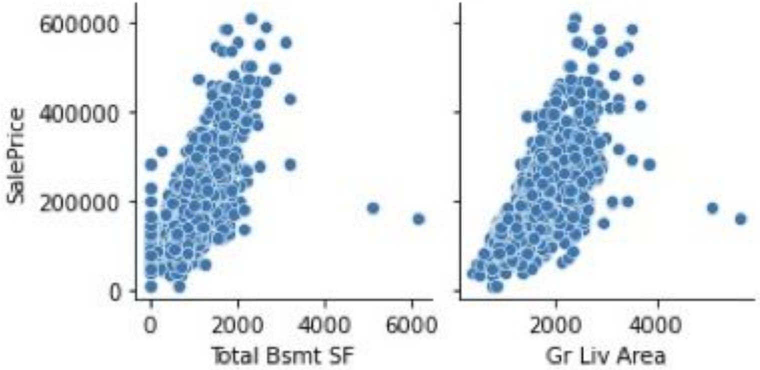


Fig.3. Continuous features that are correlated with Sale Price

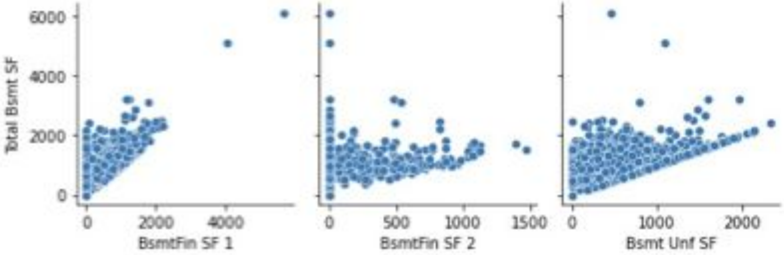


Fig.4. Collinearity between Continuous features

- Pair plots are used to find the correlation between continuous/discrete relationship
- Collinearity between continuous features can be removed
- Reduced down to 2 continuous features: Total Bsmt and Gr Liv Area

Feature Selection - Continuous/Discrete

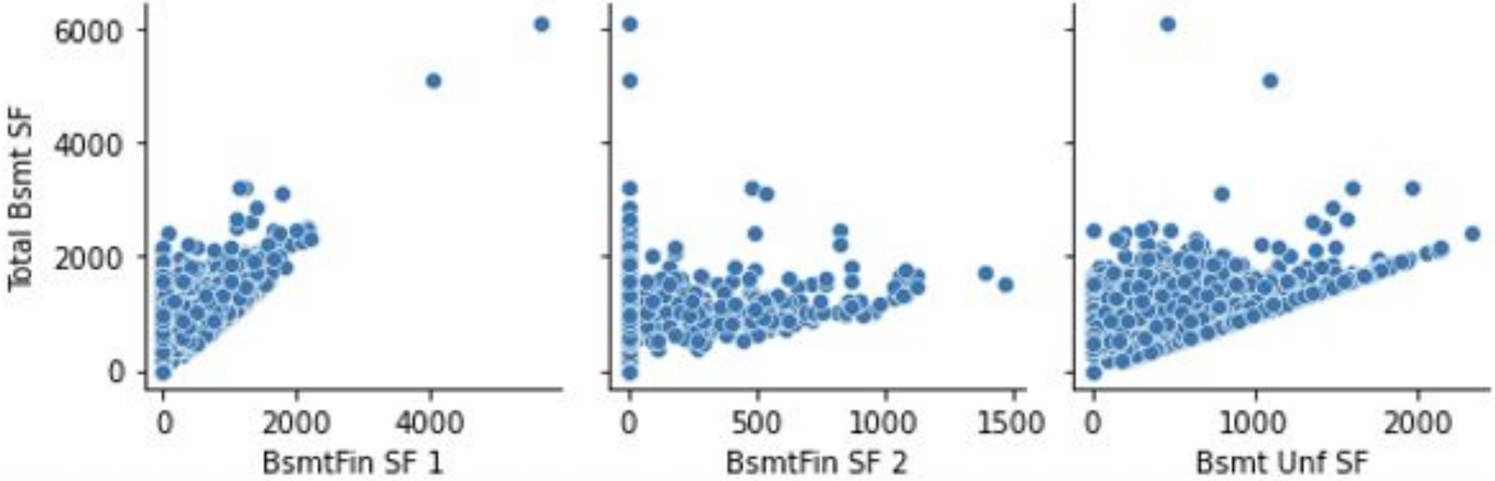


Fig.11. Collinearity between Continuous features

- Pair plots are used to find the correlation between continuous/discrete relationship
- Collinearity between continuous features

Feature Selection - Continuous/Discrete

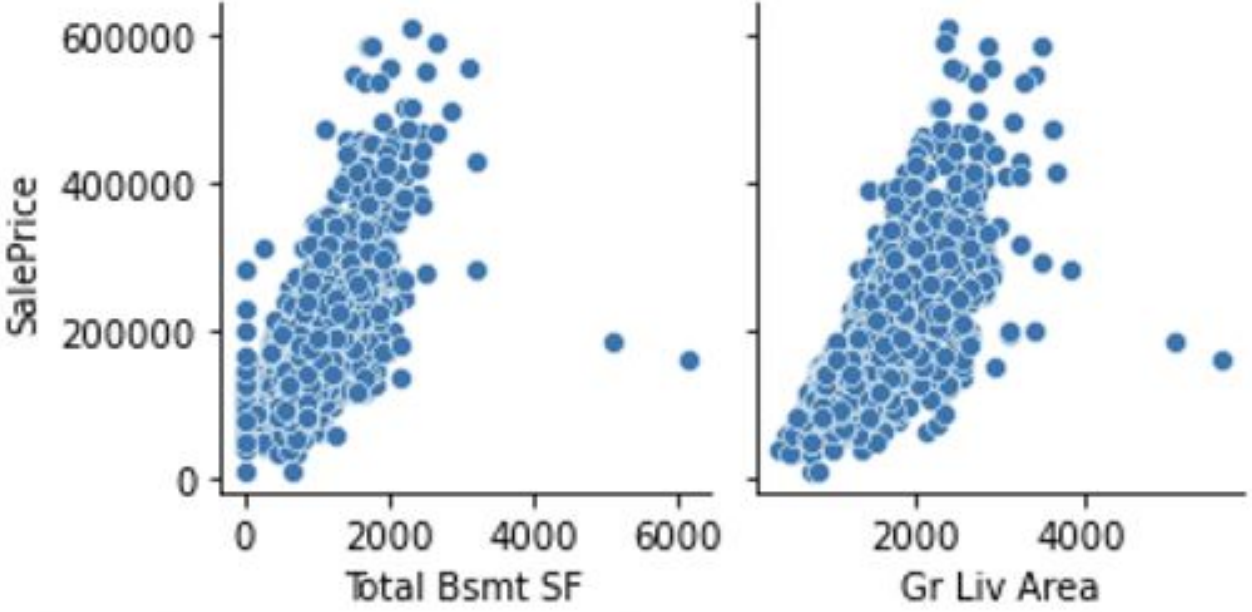


Fig.12. Continuous features that are correlated with Sale Price

- Pair plots are used to find the correlation between continuous/discrete and target feature, SalePrice
- Total Bsmt SF and Gr Liv Area correlated with SalePrice

Feature Selection - Nominal

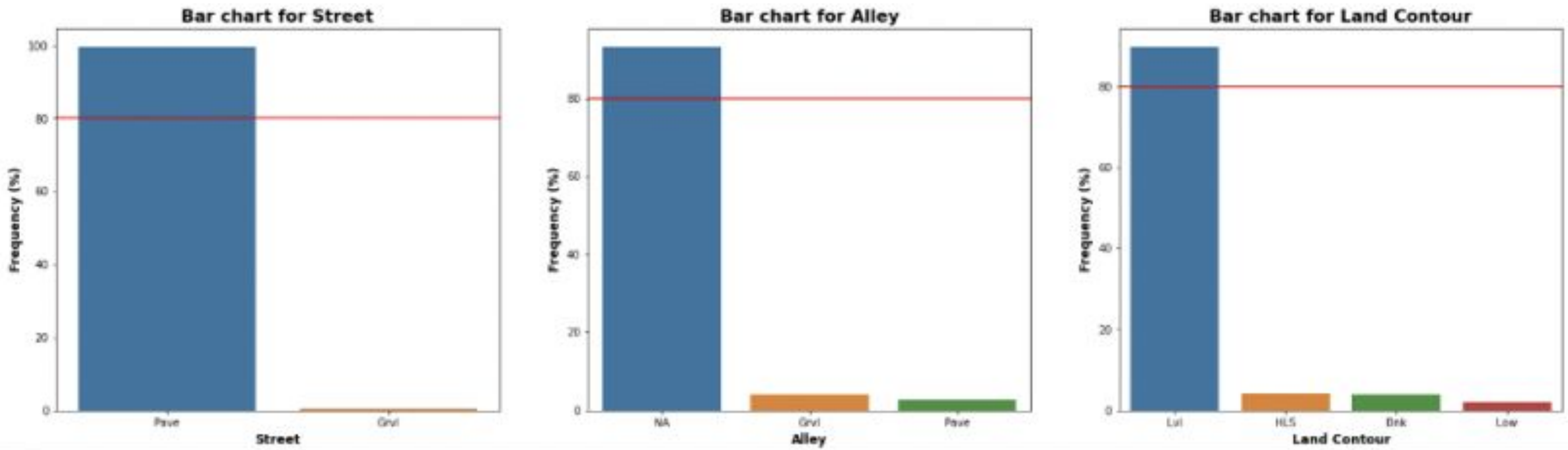


Fig.13. Nominal features with overwhelming occurrence of a single category

- Bar charts used to visualize nominal features with a single category > 80% occurrence

Feature Selection - Ordinal

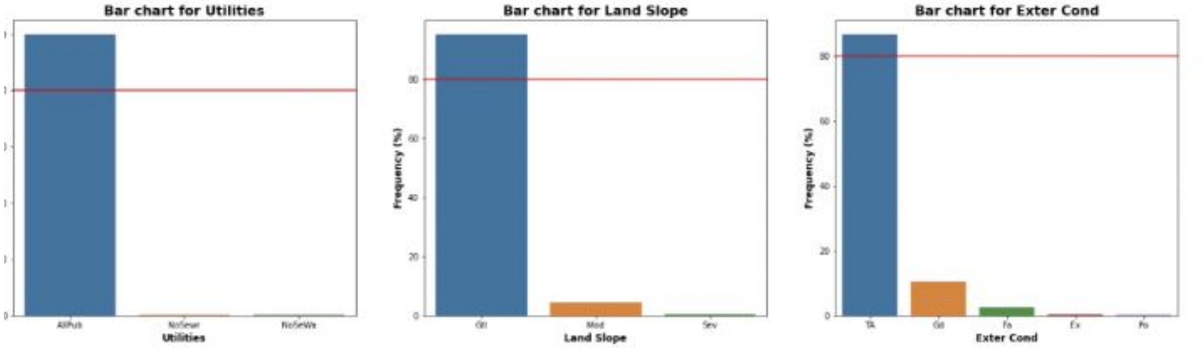


Fig.14. Ordinal features with overwhelming occurrence of a single category

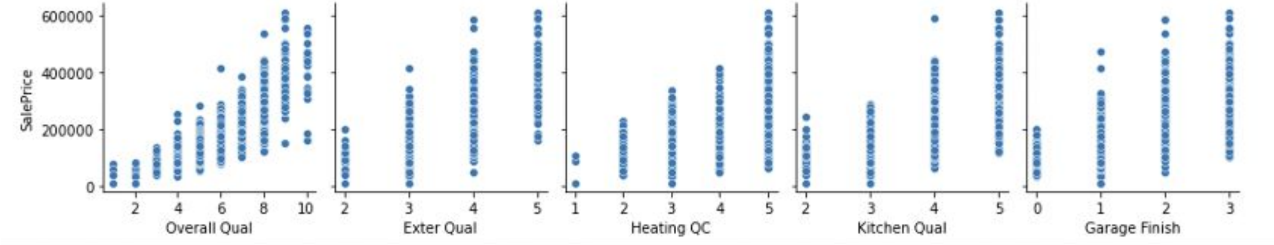


Fig.15. Ordinal features that are correlated with Sale Price

- Bar charts used to visualize ordinal features with a single category > 80% occurrence
- Pair plots are used to find the correlation for ordinal features after converted to numerical values

Feature Engineering - Nominal

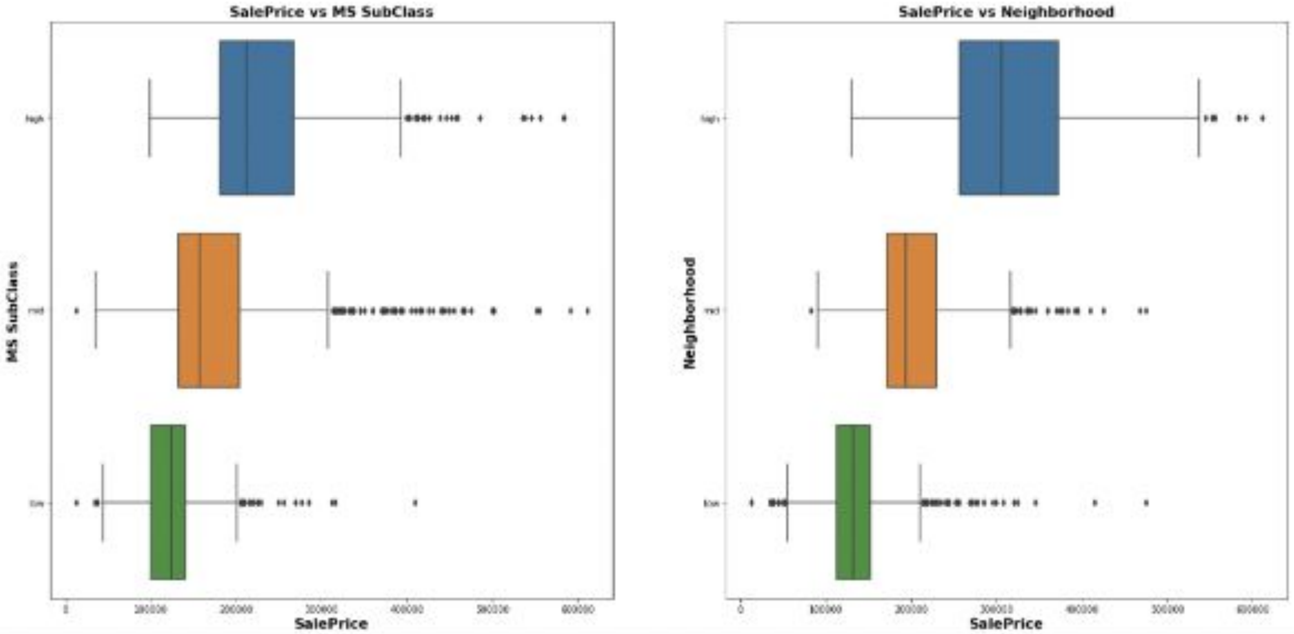


Fig.16. Nominal features with more than 15 categories group into sub-classes

- Nominal features with more than 15 categories are split into 3 sub-classes
- The split is based on the median Sale price into ‘high’, ‘mid’ and ‘low’

Feature Selection – Nominal Features

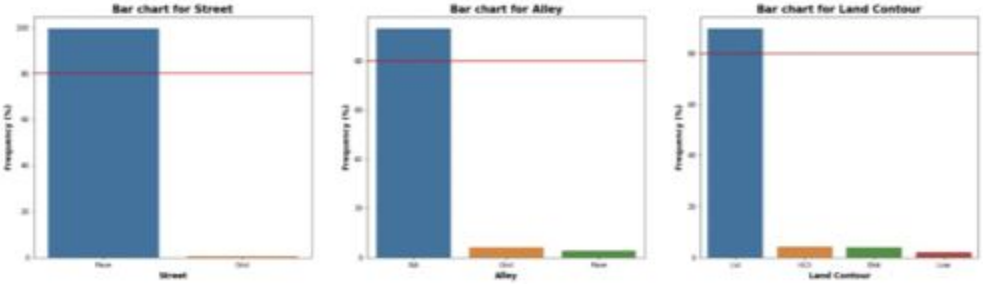


Fig.5. Nominal features with overwhelming occurrence of a single category

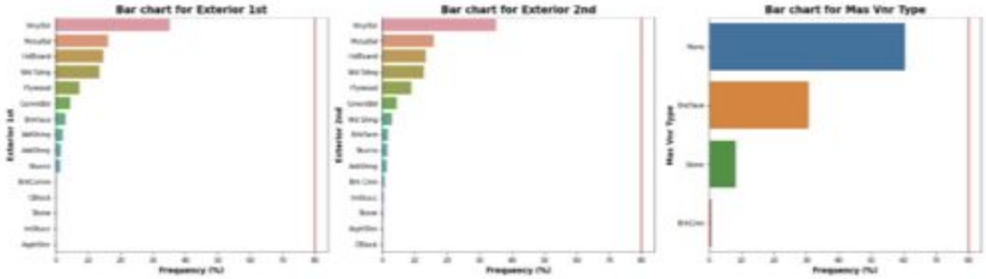
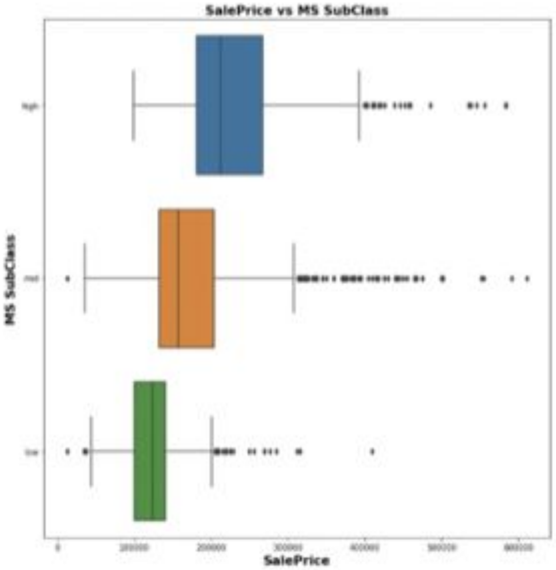


Fig.6. Nominal features without overwhelming occurrence of a single category

- Bar charts used to visualize nominal features with a single category > 80% occurrence
- Reduced to 11 nominal features

Feature Engineering – Nominal Features



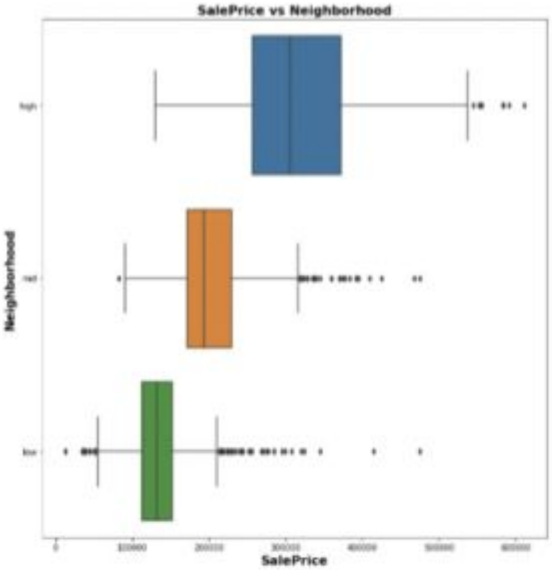


Fig.6. Nominal features with more than 15 categories group into sub-classes

- Nominal features with more than 15 categories are split into 3 sub-classes
- The split is based on the median Sale price into 'high', 'mid' and 'low

Feature Selection – Ordinal Features

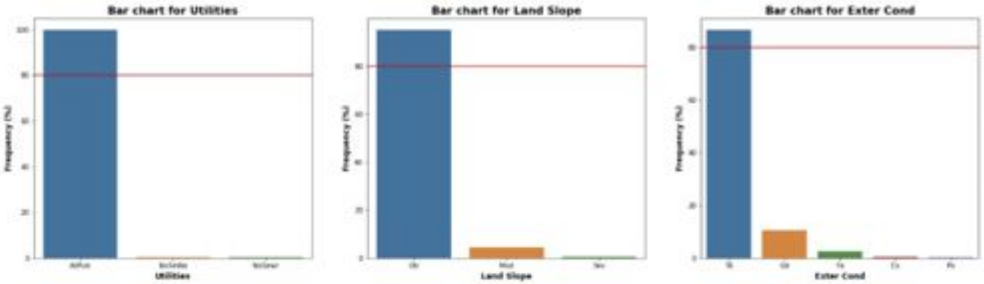


Fig.7. Ordinal features with overwhelming occurrence of a single category

- Bar charts used to visualize ordinal features with a single category > 80% occurrence
- Reduced to 11 ordinal features

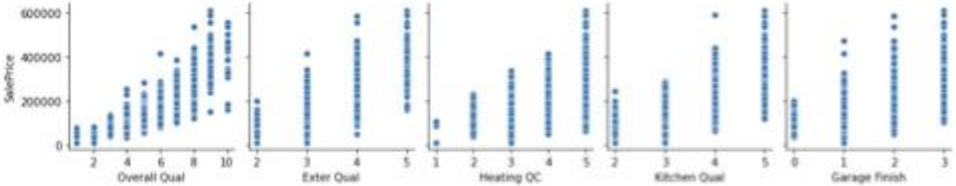


Fig.8. Ordinal features that are correlated with Sale Price

- Pair plots are used to find the correlation for ordinal features after converted to numerical values
- Further reduced to 5 ordinal features

Model Evaluation – Root Mean Square Error (RMSE)

Table 1. Table of comparison for different regression models

	Description	Hyperparameter	Number of Features	CV RMSE	Kaggle RMSE
Model 1	Linear Reg.	-	2	49996.5	46816.5
Model 2	Linear Reg.	-	85	34992.9	35610.1
Model 3	Ridge Reg.	alpha = 26	85	34541.8	34482.2
Model 4	Lasso Reg.	alpha = 97.7	38	34328.9	34264.7
Model 5	ElasticNet Reg.	alpha – 0.02, l1_ratio = 0.3	85	34546.9	34441.8

- Lasso Regression model has the best predictive performance in terms of RMSE
- Used lesser features than other models

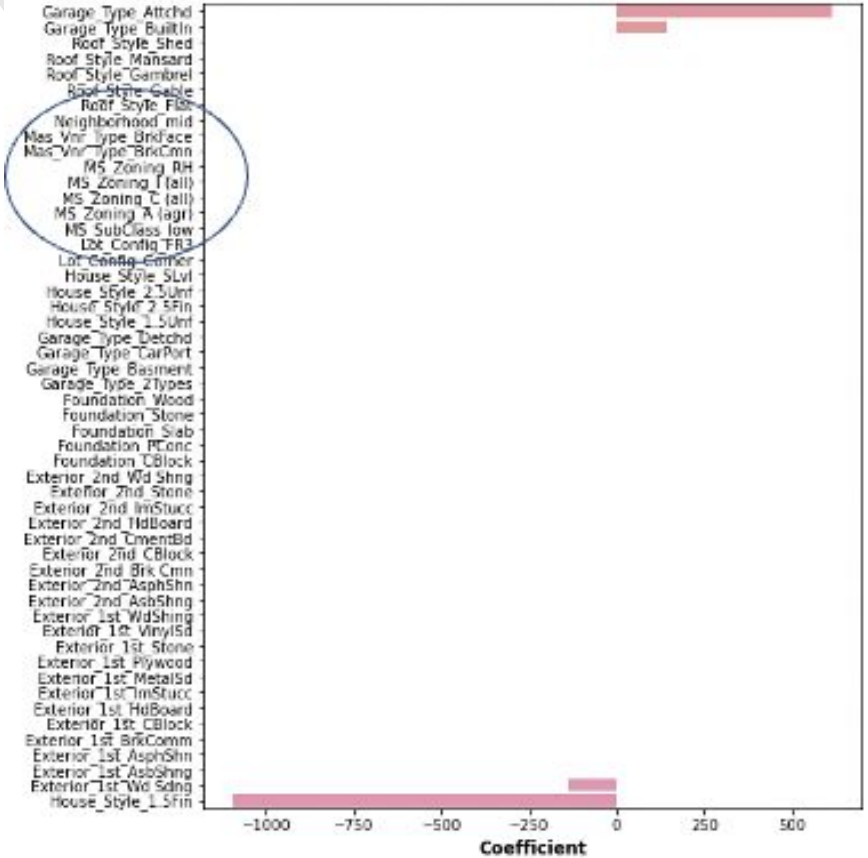
Model Evaluation - Root Mean Square Error (RMSE)

Model No.	Model Used	Alpha	L1 Ratio	No. of Features	CV RMSE	Kaggle RMSE
1	Linear Regression	n/a	n/a	2	49,997	46,817
2	Linear Regression	n/a	n/a	85	34,993	35,610
3	Ridge Regression	26	n/a	85	34,542	34,482
4	Lasso Regression	97.7	n/a	38	34,329	34,265
5	Elastic Net Regression	0.02	0.3	85	34,547	34,442

Initial Model Selected: Lasso Model

- Lowest RMSE
- Lowest No. of Features Used

Model Evaluation - Root Mean Square Error



- Lasso model has 0 coefficient for some of the MS Zone and Neigborhood features

Model Evaluation - Root Mean Square Error (RMSE)

Model No.	Model Used	Alpha	L1 Ratio	No. of Features	CV RMSE	Kaggle RMSE
1	Linear Regression	n/a	n/a	2	49,997	46,817
2	Linear Regression	n/a	n/a	85	34,993	35,610
3	Ridge Regression	26	n/a	85	34,542	34,482
4	Lasso Regression	97.7	n/a	38	34,329	34,265
5	Elastic Net Regression	0.02	0.3	85	34,547	34,442

Final Model Selected: Elastic Net

- Inclusion of Important Variables
- Good tradeoff against less features and greater accuracy

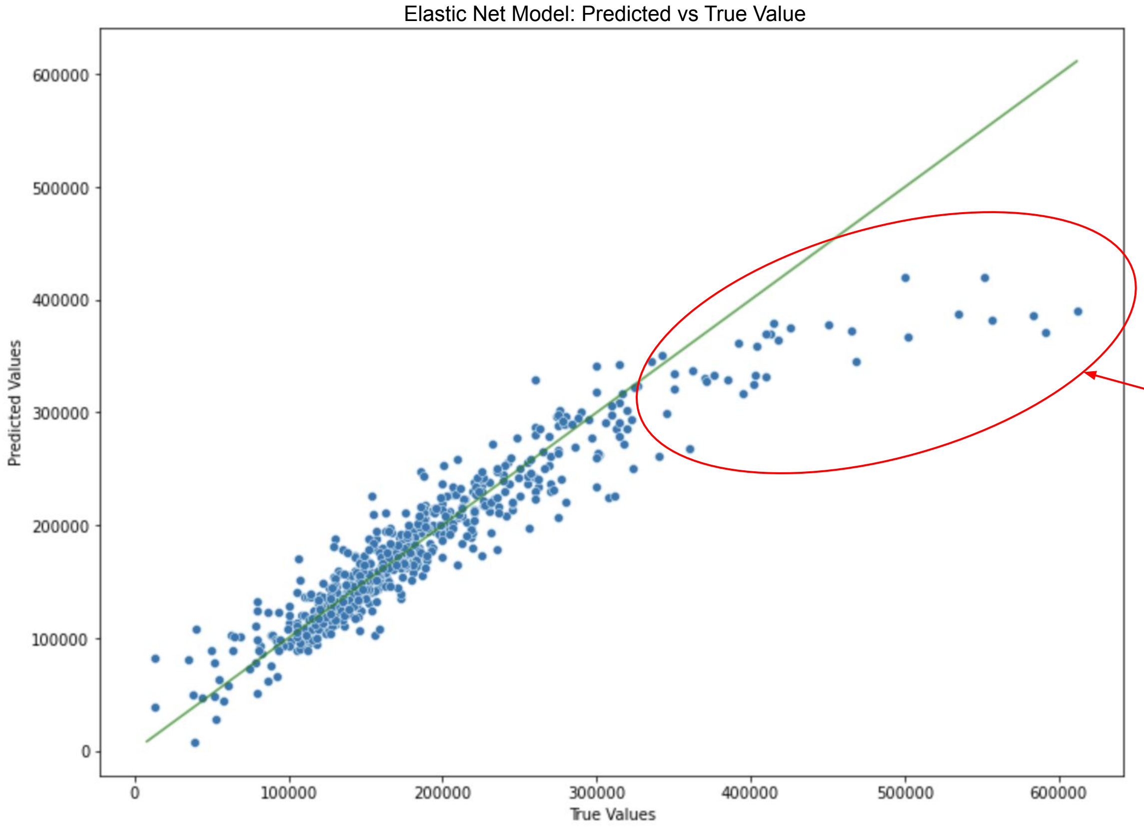
Model Evaluation - Root Mean Square Error (RMSE)

Table 1. Table of comparison for different regression models

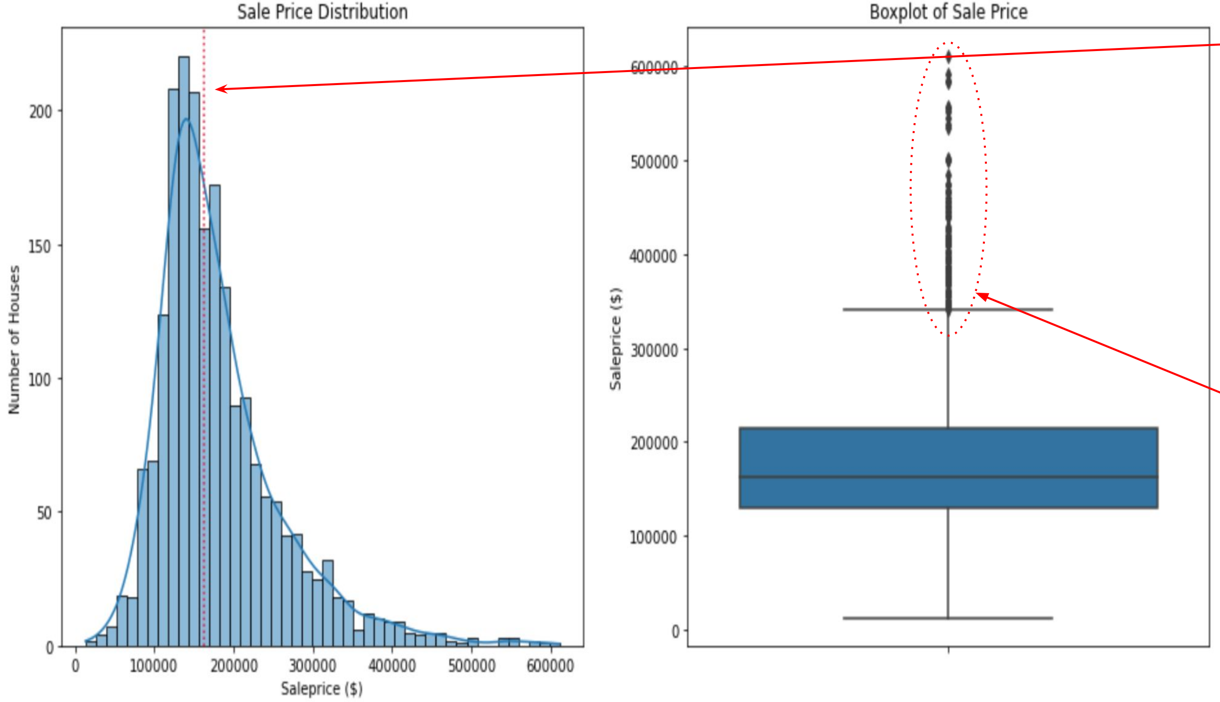
	Description	Hyperparameter	Number of Features	CV RMSE	Kaggle RMSE
Model 1	Linear Reg.	-	2	49996.5	46816.5
Model 2	Linear Reg.	-	85	34992.9	35610.1
Model 3	Ridge Reg.	alpha = 26	85	34541.8	34482.2
Model 4	Lasso Reg.	alpha = 97.7	38	34328.9	34264.7
Model 5	ElasticNet Reg.	alpha = 0.02, l1_ratio = 0.3	85	34546.9	34441.8

ElasticNet model chosen

How well is did our model predicting sale price?



- Accurate predictions below $300,000
- Under-predictions above $300,000



- Median Price = $162,500
- Right-Skewed
- 61 Outliers

Estimate of Model Performance

Model	Hyperparams	Num Features	Train RMSE	Holdout RMSE
OLS (1 Feature)	-	1	56210	58465
OLS (All Features)	-	199	24079	30629
Ridge	Alpha=327	199	26526	30604
Lasso	Alpha=893	66	27160	30842
ElasticNet	Alpha=0.64 ratio=0.5	77	27766	28643

Optimum Model

Model Evaluation – Top 10 features

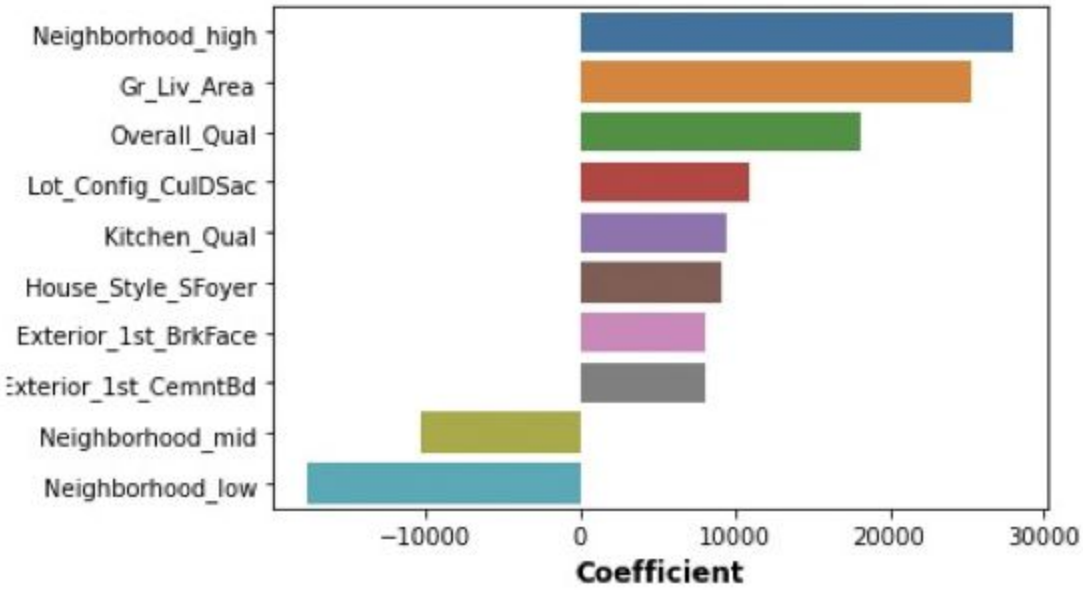


Fig.9. Top 10 features that affect the Sale Price

Baseline Model

Train/Test	Metric	Result
Train	R^2	0.126
Test	R^2	0.116
Train	Cross Val Score R^2	0.104
Train	RMSE	$74,194
Test	RMSE	$74,047

We create a baseline model based only on one variable: lot area.

This variable is used as it is the default variable to use in assessing a property.

Low R^2 and High Root Mean Squared Error suggests that this baseline model has a very limited predictive power.

Elastic Net Model (Numeric Categories)

Train/Test	Metric	Result
n/a	alpha	87.625
n/a	L1 Ratio	1.0
Train	R^2	0.867
Test	R^2	0.868
Train	RMSE	$28,975
Test	RMSE	$28,585

We ran a elastic net model based only on numeric categories such as Garage Area and Lot Area.

The main purpose is to see whether there is a clear preference towards one type of penalization as opposed to the other.

Result: Enet model with large alpha range (1 to 100) suggests that a Lasso Model is more appropriate.

Lasso Model 1: 11 Variables

Train/Test	Metric	Result
n/a	alpha	1.0
Train	R^2	0.883
Test	R^2	0.872
Train	RMSE	$27,175
Test	RMSE	$28,156

Lasso model of both quantitative and qualitative features.

Observations:

1. Inclusion of qualitative features provide greater predictive power.
2. Feature selection allows us to have a better model even with lower number of variables.

Target RMSE: $32,044

Variables Used: Neighborhood, Total Living Area, Overall Quality, Total Bsmt SF, House Age, Property SubClass, Garage Area, Lot Area, External Quality, 1st Floor Exterior, Heating Type

Lasso Model 2: 7 Variables

Train/Test	Metric	Result
n/a	alpha	1.0
Train	R^2	0.883
Test	R^2	0.872
Train	RMSE	$27,175
Test	RMSE	$28,156

Lasso Model 1:
11 Variables

Train/Test	Metric	Result
n/a	alpha	0.001
Train	R^2	0.868
Test	R^2	0.856
Train	RMSE	$28,815
Test	RMSE	$29,894

Lasso Model 2:
7 Variables

Lasso model based on a subset of variables used in Lasso Model 1

Trade-off between bias and variance

Although we get a lower predictive value, the decrease of more than ⅓ of variables may be worth it.

Target RMSE: $31, 962

Variables Used: Neighborhood, Total Living Area, Overall Quality, Total Bsmt SF, House Age, Property SubClass, Garage Area, ~~Lot Area, External Quality, 1st Floor Exterior, Heating Type~~

Limitations

- Cannot take into consideration time as a factor if we want to recommend as investment
- Does not factor in inflation

Conclusion

- Lasso regression has the best predictive performance among the models with 50% reduction in the number of features with RMSE of ~34K
- The Lasso regression model produce a set of coefficients for the respective features
- The top 3 features that will influence the price are location of the house, the total area of house and the overall quality of the house
- With a set of features, the model is able to predict the price of a house

Conclusion

1. Only using Lot Area as a measure of Sale Price is insufficient.
2. Mix of quantitative and qualitative variables perform better due to the premiums associated with the qualitative variables not captured in quantitative variables.
3. Using less variables may be worth it: targets high-value variables and reduces noise in the predictive model
4. While this model is significantly better than the baseline, we can improve with more data. Some recommended data to obtain:
    a. Neighborhood school zone
    b. HOA fees
    c. Neighborhood crime rate
    d. Access to services (banks, supermarkets, etc)
    e. Access to highways and other public services

Recommendation

Location: Choose a good neighborhood with house on a Cul-de-sac lot

Size: Maximize Ground Living Area

Modifications:
Build Brick Face & Cement Exterior Covering
Build Split Foyer
Assess Kitchen & Overall Quality