Ames Housing Price Prediction Model

By: Alyse, Sim Yi

Background

Who are we?

HOUSE FLIPPING CONSULTANCY SERVICE

With Core Business in Other Midwestern States

We are expanding in the Neighbouring state of Iowa

with Ames as our first target

Background

Why Iowa? Why Ames?

- **Seller's Market:**
  - 3.9% Increase in Y-o-Y House Prices (2018)
  - Overall Increase in Number of Houses Sold
- **Booming Home Sales in Ames:**
  - 9.4% Increase in Y-o-Y Number of Houses Sold
- **Highly Profitable:**
  - Cheap renovation and remodelling costs
  - Average flip earns an average of US$ 54K
  - Top 50 cities in United States for profiting from house flips

Problem Statement

What are we trying to address?

Identify key driving factors for maximising returns from house flipping activities

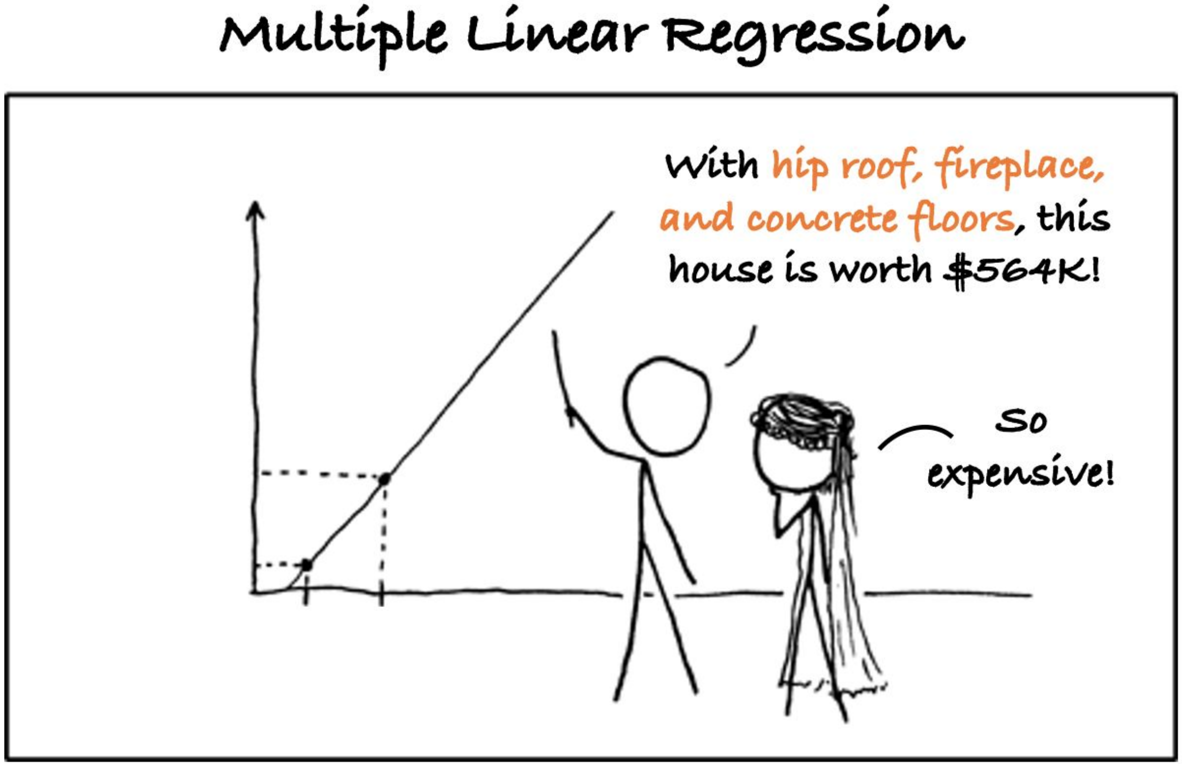
WHAT TO
BUY

HOW TO
RENOVATE

WHEN TO
SELL

Problem Statement

How are we going to do it?



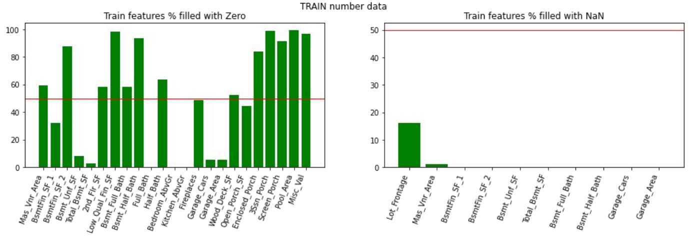
Project Overview

End-to-end management of project

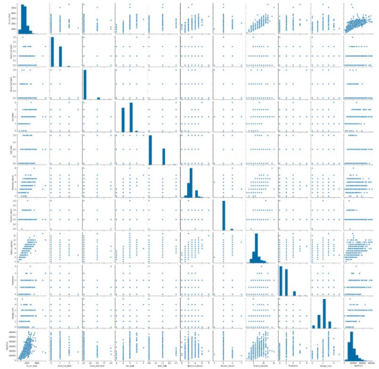
1. **Data Preparation**
   a. Data Cleaning
   b. Exploratory Data Analysis
2. **Exploratory Visualizations**
   a. Correlations with Dependent Variable
   b. Multicollinearity
   c. Skewness
3. **Modeling and Testing**
4. **Inferential Visualizations**
5. **Conclusion & Recommendations**

Data Processing

How we classified our data

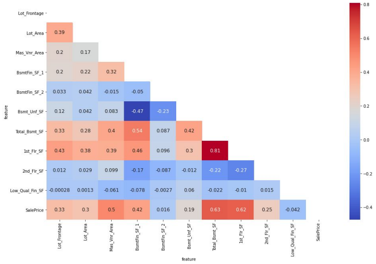


Spread / Scatter

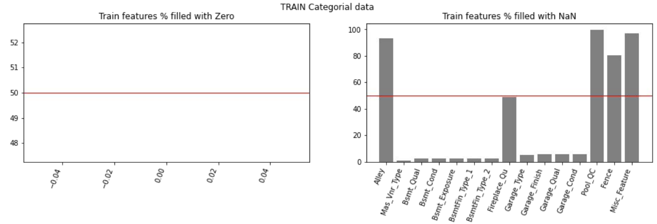


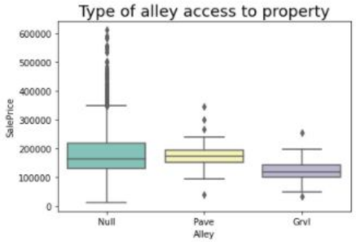
Numerical - Nominal and Ordinal

Correlation



Categorical - Nominal and Ordinal





Data Cleaning

How did we prepare our data for modelling?

1. Filling null values with :
   a. 0 for numerical values
   b. NA for categorical values
2. Changing Ordinal Categorical data to numerical values

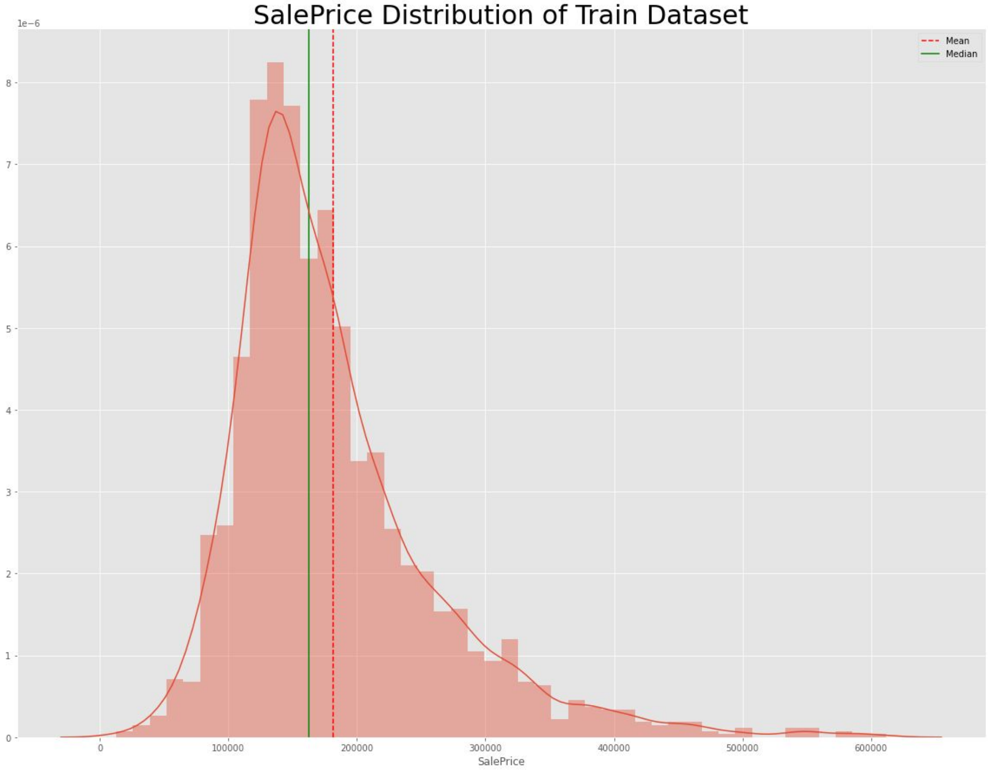
```
#filling dtypes object nan values
df['Bsmt Qual'].fillna('NA', inplace = True)
df['Bsmt Cond'].fillna('NA', inplace = True)
df['Bsmt Exposure'].fillna('NA', inplace = True)
df['BsmtFin Type 1'].fillna('NA', inplace = True)
df['BsmtFin Type 2'].fillna('NA', inplace = True)

#filling dtypes float64 nan values with number 0
df['BsmtFin SF 1'].fillna(0 , inplace = True)
df['BsmtFin SF 2'].fillna(0 , inplace = True)
df['Bsmt Unf SF'].fillna(0 , inplace = True)
df['Total Bsmt SF'].fillna(0 , inplace = True)
df['Bsmt Full Bath'].fillna(0 , inplace = True)
df['Bsmt Half Bath'].fillna(0 , inplace = True)
```

```
#Copied from Data cleaning to apply on Test data
labels = {'Exter Qual': {'Ex': 5, 'Gd': 4, 'TA': 3, 'Fa': 2, 'Po': 1} }
df_test.replace(labels, inplace=True)


labels = {'Exter Cond': {'Ex': 5, 'Gd': 4, 'TA': 3, 'Fa': 2, 'Po': 1} }
df_test.replace(labels, inplace=True)

labels = {'Bsmt Qual': {'Ex': 5, 'Gd': 4, 'TA': 3, 'Fa': 2, 'Po': 1, 'NA': 0} }
df_test.replace(labels, inplace=True)


labels = {'Bsmt Cond': {'Ex': 5, 'Gd': 4, 'TA': 3, 'Fa': 2, 'Po': 1, 'NA': 0} }
df_test.replace(labels, inplace=True)

labels = {'Bsmt Exposure': {'Gd': 4, 'Av': 3, 'Mn': 2, 'No': 1, 'NA': 0} }
df_test.replace(labels, inplace=True)
```

Exploration of Data

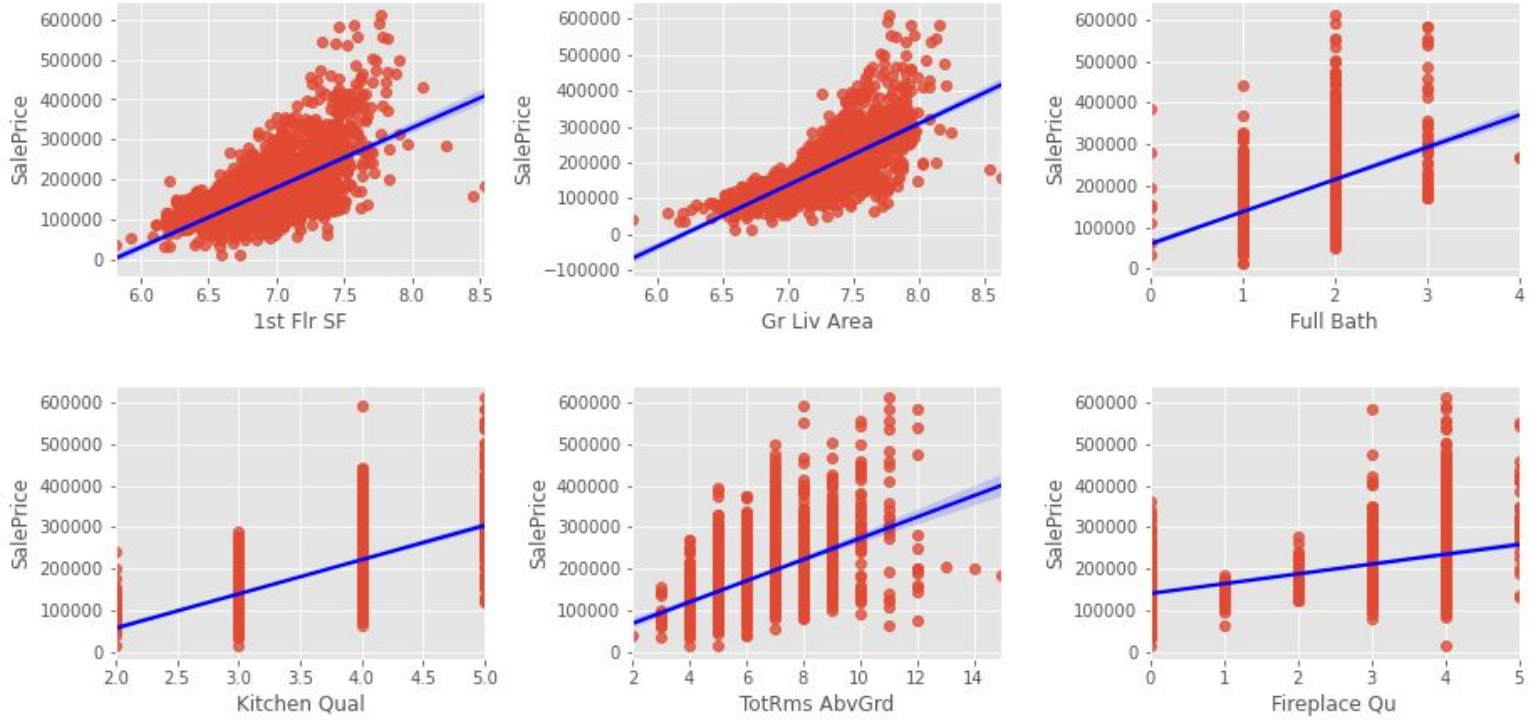
Continuous Data Plotting

1. Saleprice is positively skewed
2. the mean and median are more towards the lower prices

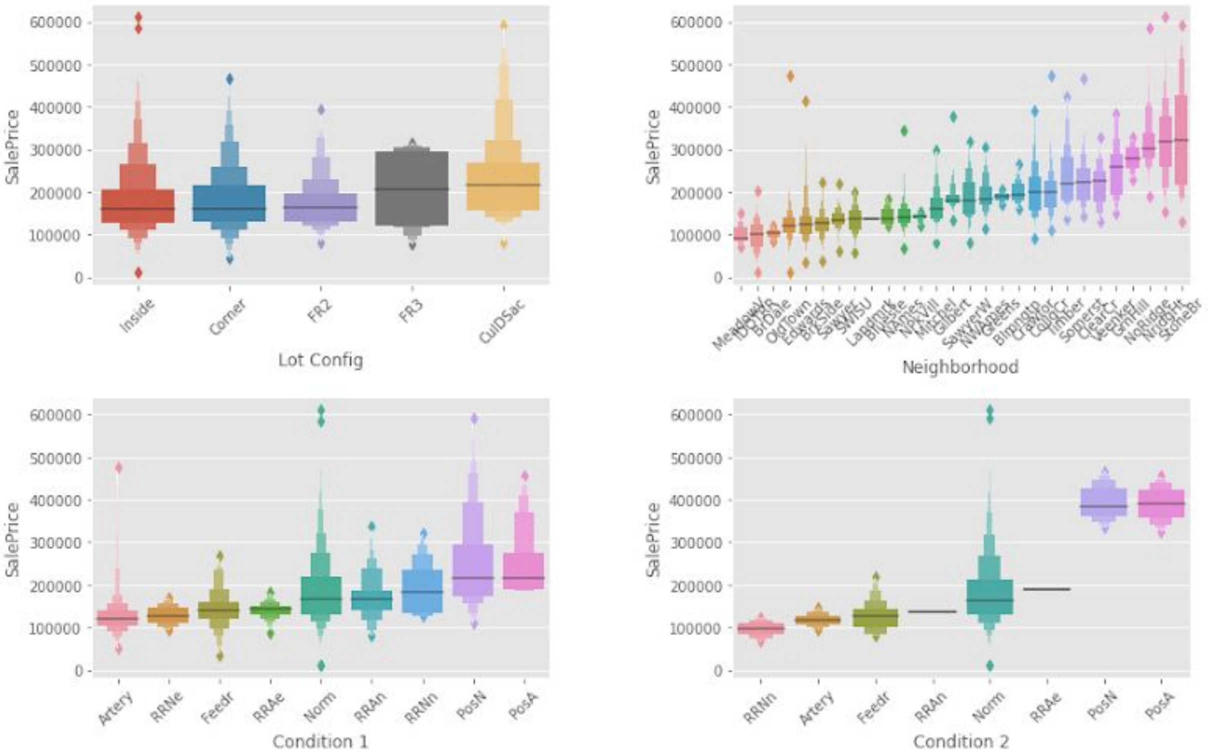


Exploration of Data

Continuous Data Plotting

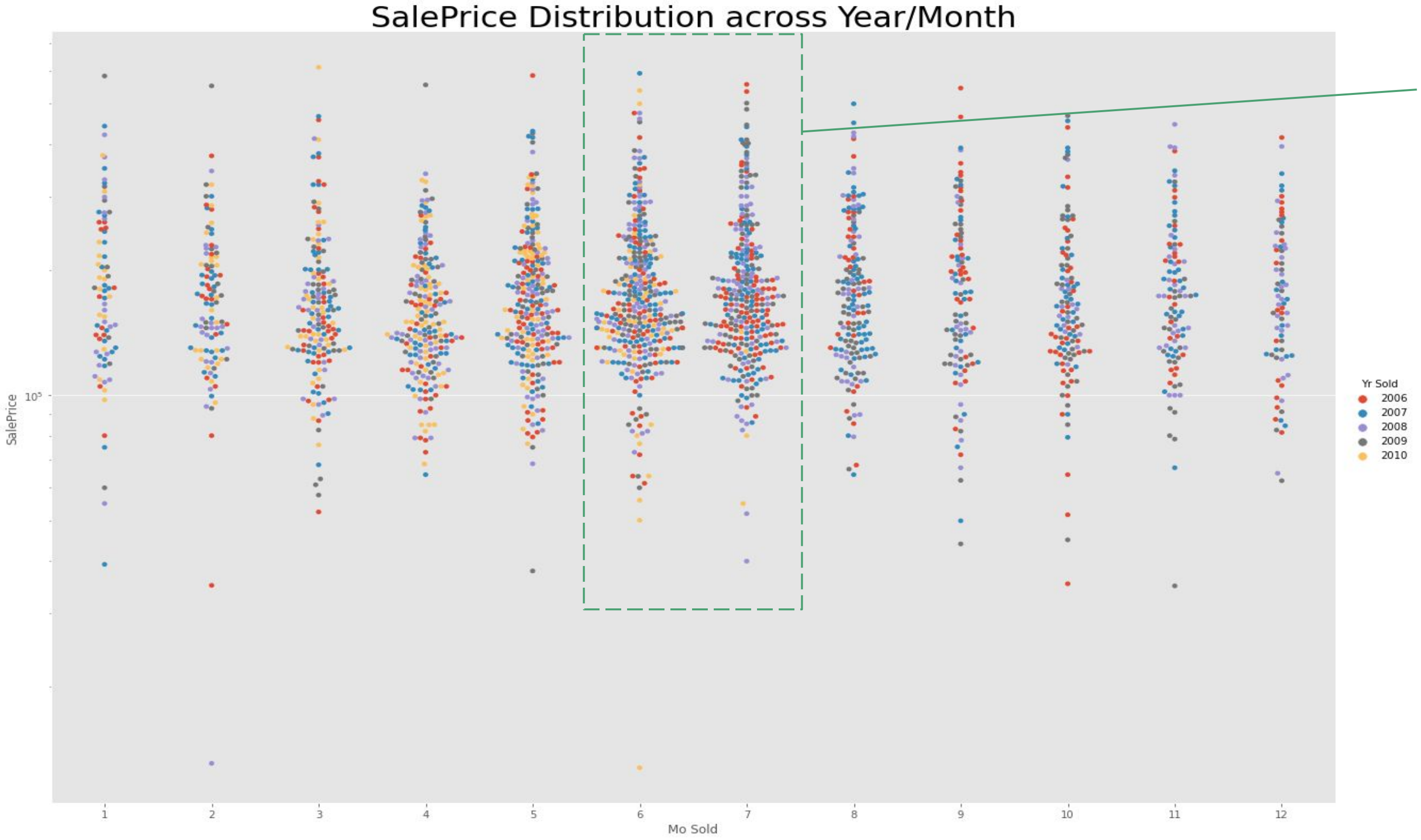


Exploration of Data
Discrete Data Plotting



Exploration of Data

Time Series Swarmplot

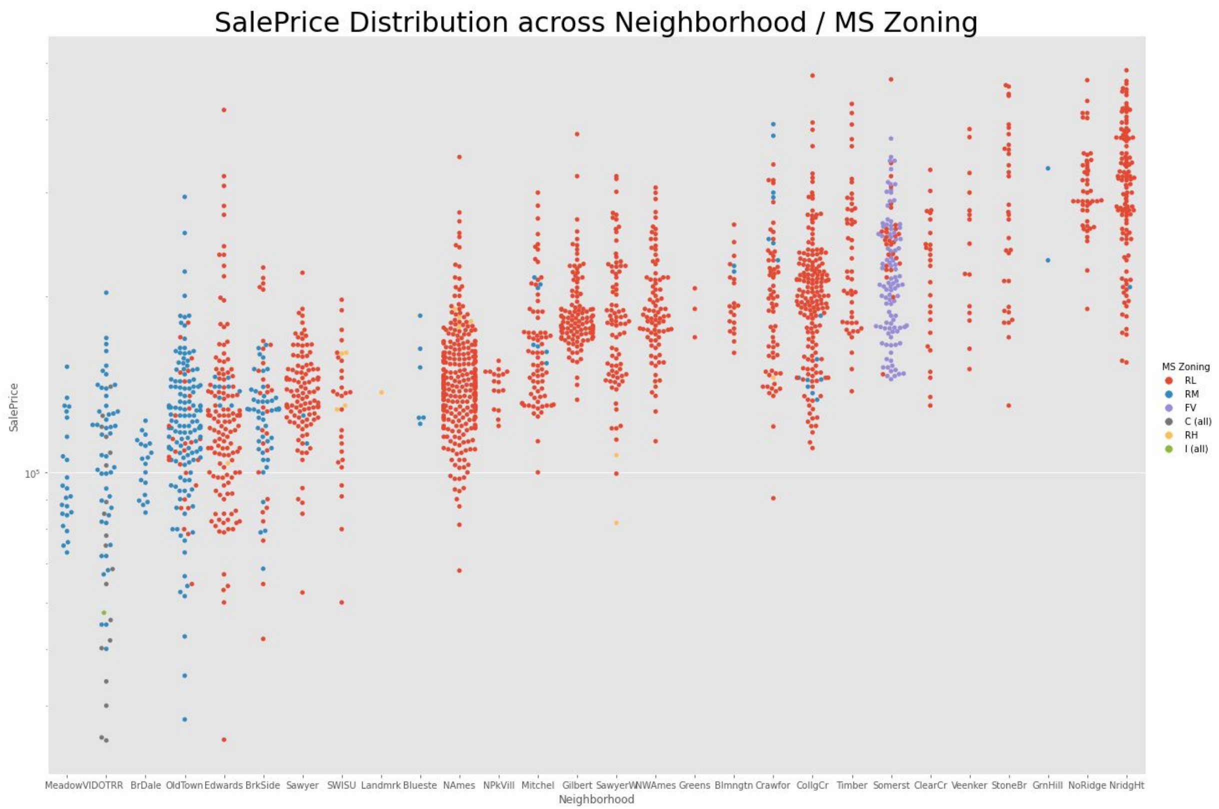


June and July

Highest sales volume

Exploration of Data

Neighbourhood Distribution Swarmplot



Neighbourhood factor is a key determinant of house prices

NOT IMPORTANT in short-term house flipping investment

Legend: MS Zoning

A Agriculture

C Commercial

FV Floating Village Residential

I Industrial

RH Residential High Density

RL Residential Low Density

RP Residential Low Density Park

RM Residential Medium Density

Feature Engineering

Standard Scaler, One_Hot Encoding, Polynomial Features

```python
num_data = df.select_dtypes(['int64', 'float64']).keys()
num_data = [x for x in num_data if ((x != 'SalePrice') & (x != 'Id'))]

nums = df[num_data]
ss = StandardScaler()
ss.fit(nums)
nums_scaled = ss.transform(nums)

nums_scaled.shape
```

```
(2018, 21)
```

```python
nums_scaled_pd = pd.DataFrame(nums_scaled, columns = num_data) #create pd for combining later
nums_scaled_pd.head()
```

	Year Built	Year Remod/Add	Bsmt Qual	Central Air	1st Flr SF	Full Bath	Kitchen Qual	TotRms AbvGrd	Garage Finish	Garage Cars	...	Overall Total Value	Exter Total Value	Total Flr SF	Fireplace Total Value
0	0.142892	0.995556	-0.552436	0.268819	-1.311506	0.799945	0.750726	-0.267463	0.312864	0.308899	...	1.416512	0.757663	0.154064	-0.893370
1	0.809074	0.614543	0.580507	0.268819	-0.583048	0.799945	0.750726	1.071845	0.312864	0.308899	...	0.199861	0.757663	1.311132	0.452018
2	-0.623217	1.090809	-0.552436	0.268819	-0.120233	-1.057695	0.750726	-0.937117	-0.806568	-1.023062	...	0.199861	0.757663	-0.922422	-0.893370
3	1.142165	1.090809	0.580507	0.268819	-1.229783	0.799945	-0.782640	0.402191	1.432296	0.308899	...	-1.016789	-0.715327	0.077311	-0.893370
4	-2.388599	0.424037	-1.685379	0.268819	-0.880397	0.799945	-0.782640	-0.267463	-0.806568	0.308899	...	1.416512	-0.715327	0.079529	-0.893370

5 rows × 21 columns

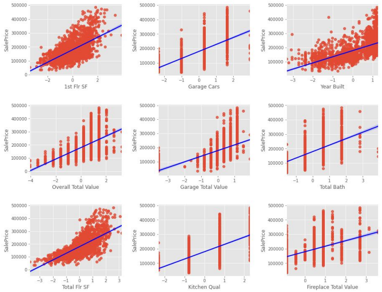
```python
obj_processed = pd.get_dummies(df[obj_data], columns = obj_data)
obj_processed.head()
```

	MS Zoning_C (all)	MS Zoning_FV	MS Zoning_I (all)	MS Zoning_RH	MS Zoning_RL	MS Zoning_RM	Street_Grvl	Street_Pave	Land Contour_Bnk	Land Contour_HLS	...	Heating_Grav	Heating_OthW	H
0	0	0	0	0	1	0	0	1	0	0	...	0	0	
1	0	0	0	0	1	0	0	1	0	0	...	0	0	
2	0	0	0	0	1	0	0	1	0	0	...	0	0	
3	0	0	0	0	1	0	0	1	0	0	...	0	0	
4	0	0	0	0	1	0	0	1	0	0	...	0	0	

5 rows × 123 columns

```python
#remove columns with NA count values
na_col = obj_processed.filter(regex = 'NA')
na_col_keys = na_col.keys()
na_col_keys
```

```
Index(['Neighborhood_NAmes', 'Mas Vnr Type_NA', 'Garage Type_NA'], dtype='object')
```




```python
print("Garage & Total & Year Features RMSE(L2): " +
      str(np.sqrt(-cross_val_score(lasso, X = X,y = y, cv = 5,
                                   scoring = 'neg_mean_squared_error').mean())))
```

```
Garage & Total & Year Features RMSE(L2): 30706.89039192311
```

```python
poly = PolynomialFeatures(degree = 2, interaction_only = True, include_bias=False)
```

```python
X_poly = poly.fit_transform(X)
```

```python
print("Garage & Total & Year Polynomial Features RMSE(L2): " +
      str(np.sqrt(-cross_val_score(lasso, X = X_poly, y = y, cv = 5,
                                   scoring = 'neg_mean_squared_error').mean())))
```

```
Garage & Total & Year Polynomial Features RMSE(L2): 23820.97583158794
```

Standard Scaler

Scale the data to the same level across dataset

One Hot Encoding

Convert categorical data into numerical data

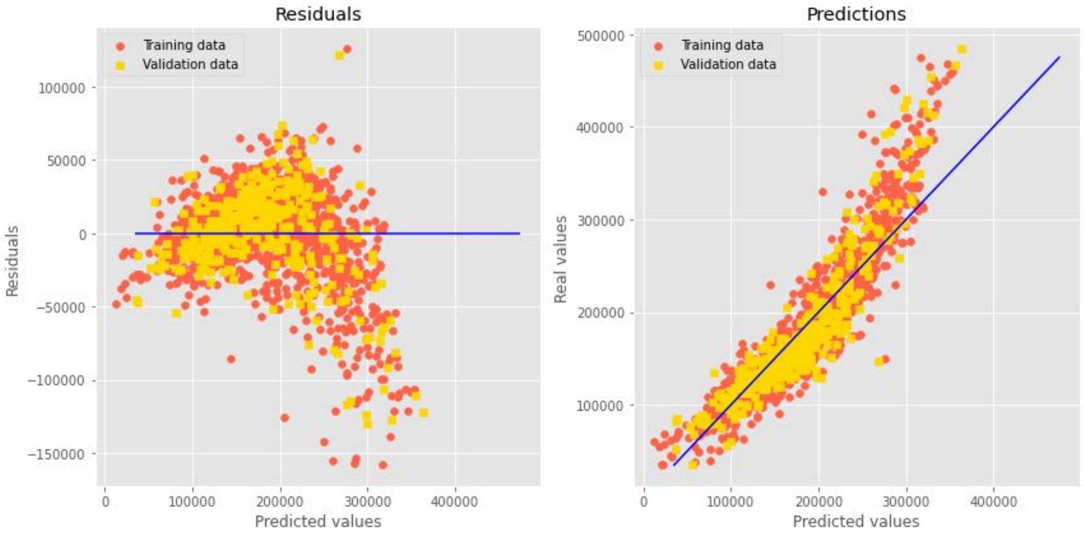
Polynomial Features/ Interaction Terms

Transform data to better fit the best fit line

Model Baseline

Linear Regression

- **Model Baseline**
- Model tuning
- Feature select
- Final model



R2 Score

- Train Score(Lr):
  - 0.898
- Validation Score(Lr):
  - 0.870
- Test Data Estimated score(Lr):
  - -8.607e+19

RMSE Scores

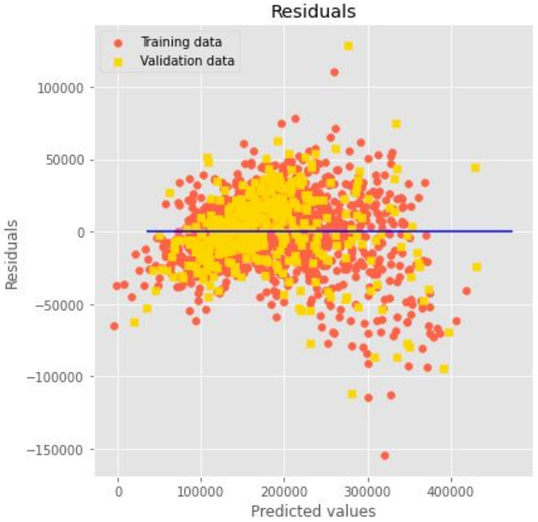
- Train RMSE(Lr):
  - 23224
- Validation RMSE(Lr):
  - 26995
- Test Data Estimated RMSE(Lr):
  - 53515

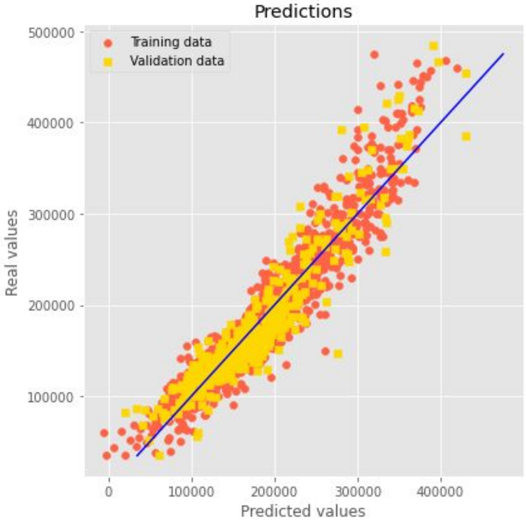
Model Tuning

Regularization

Using RidgeCV, LassoCV & ElasticNetCV (LassoCV chosen)

- Model Baseline
- **Model tuning**
- Feature select
- Final model





R2 Score

- Train Score(Lr):
  - 0.9029
- Validation Score(Lr):
  - 0.87680
- Test Data Estimated score(Lr):
  - 0.8771

RMSE Scores

- Train RMSE(Lr):
  - 22649
- Validation RMSE(Lr):
  - 26323
- Test Data Estimated RMSE(Lr):
  - 25480

Feature Selection
RFECV

- Model Baseline
- Model tuning
- **Feature select**
- Final model

```
rfecv = RFECV(lasso, step = 1, #using lasso model as the results are better among the 3 models
              min_features_to_select = 30,  #minimum of 30 features
              cv=5,
              scoring = 'neg_mean_squared_error') #calculating using mse

rfecv.fit(X_train, y_train)
```

```
RFECV(cv=5, estimator=Lasso(max_iter=2000), min_features_to_select=30,
      scoring='neg_mean_squared_error')
```

```
gd_coef = list(compress(X_train.columns, rfecv.support_))
print(str(len(gd_coef)) + ' features left after RFECV feature selection')
```

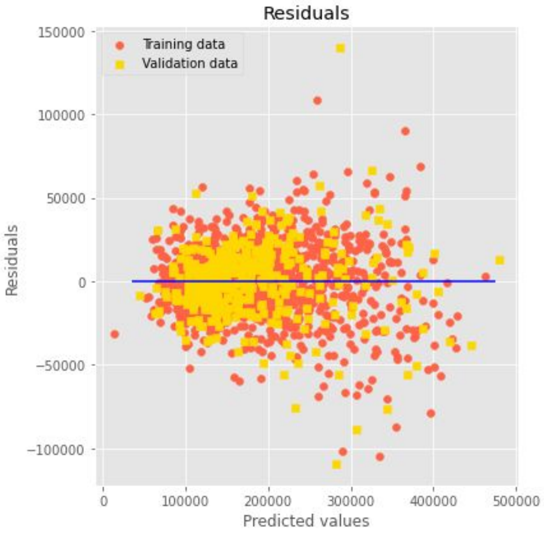
```
94 features left after RFECV feature selection
```

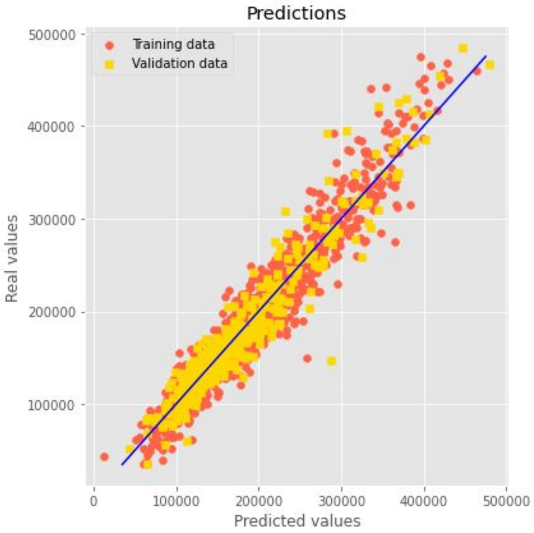
Using **RFECV** with the Lasso model to feature select the best coefficient features for further model tuning

Final Model

Final Model (Lasso Regression)

- Model Baseline
- Model tuning
- Feature select
- **Final model**





R2 Score

- Train Score(Lr):
  - 0.9318 (bef: 0.898)
- Validation Score(Lr):
  - 0.9163 (bef: 0.0.870)
- Test Data Estimated score(Lr):
  - 0.9123 (bef: -8.607e+19)

RMSE Scores

- Train RMSE(Lr):
  - 18993 (bef: 23224)
- Validation RMSE(Lr):
  - 21692 (bef: 26995)
- Test Data Estimated RMSE(Lr):
  - 21504 (bef: 53515)

Kaggle Submission

Submission Model (Ridge and Lasso Regression)

Submission and Description	Private Score	Public Score	Use for Final Score
submission_ridge.csv just now by Sim Yi add submission details	28300.91072	28325.32706	☐
submission_lasso.csv a few seconds ago by Sim Yi add submission details	27930.73538	27626.17192	☐

Key challenges

What are some limitations we faced in our modelling exercise?

Data Collection

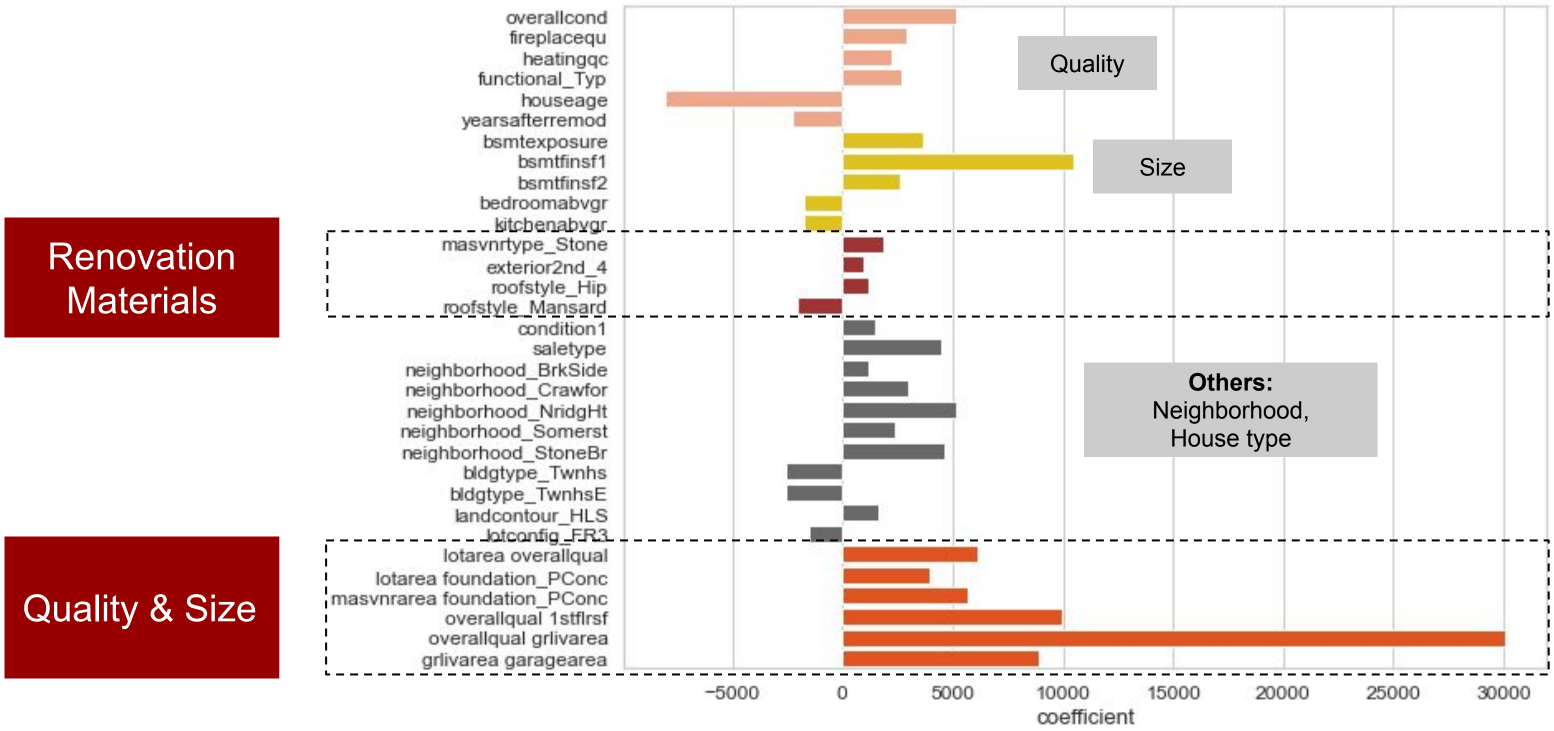
- Lack of **post-2010 data** that are extremely vital to understanding trend of house prices due to 2008 Global Financial Crisis
- **Demographic & Economic data:** How has the demographic and economic factors over the four years (and after) impacted housing market in Ames?
- **Lacking of certain data** that could be helpful for our recommendations (e.g resale rate of property, key amenities around house, transportation services etc.)

Modelling

- Many Features were **highly correlated** to each other, making it difficult to choose which features to use (multicollinearity issue)
- Use of other **more advanced modelling techniques** (ie. RandomForest) could be more efficient in our prediction model.

Results from our Model

What can we learn from our regression model?



Recommendations

How to maximise returns from house flipping?

WHAT TO BUY?

- Poor quality houses + Big area ➔ ***Cheapest & Highest Potential***
- Houses with garage and basements ➔ ***Lift house prices***

HOW TO RENOVATE?

- Poured Concrete Foundation
- Hip-style roof
- Metal Exterior Covering
- Stone Masonry Veneer

WHEN TO SELL?

- Summer (June-July)

Ensure that renovations lead to much higher quality

Big houses + Excellent quality leads to disproportionately higher sale price.