PROJECT 2

Ames Housing Data and Kaggle Challenge

Bob, Boss, Gear

Background

The project aims to explore the relationship between the Sale Price of houses in Ames, Iowa and the various features of the houses. We will be looking at some of the factors affecting house prices and using this information we will be creating regression models to predict house prices based on these features.

The findings from this project can hopefully be used by real estate firms in Ames, Iowa to help them realise the importance of the some of the housing features as well as giving them a rough guideline on how new properties with these features could be priced at.

Datasets

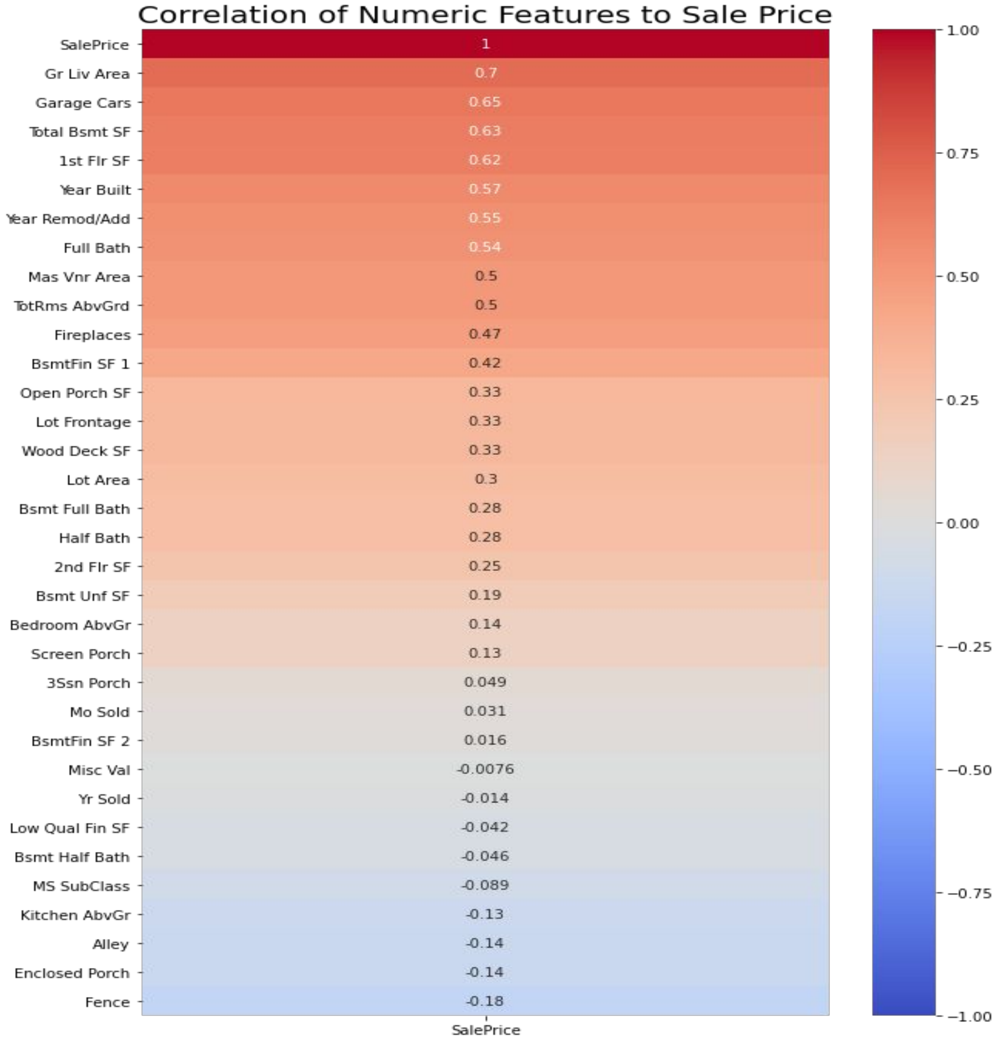
- train.csv - 2051 rows, 81 columns
- test.csv - 879 rows, 80 columns

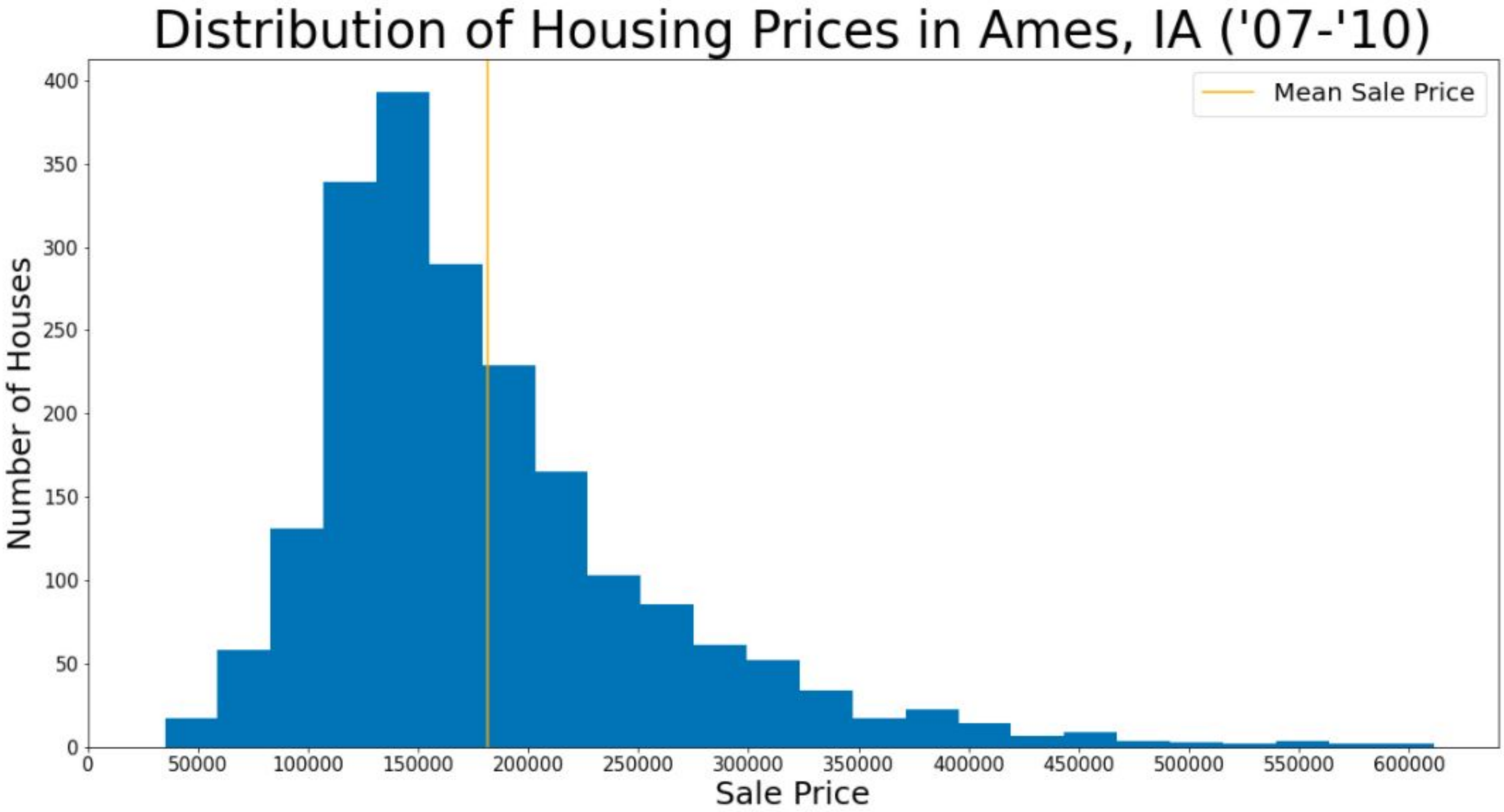
Column Types
- 23 nominal
- 23 ordinal
- 14 discrete
- 20 continuous variables
- 2 additional observation identifiers

	Id	PID	MS SubClass	MS Zoning	Lot Frontage	Lot Area	Street	Alley	Lot Shape	Land Contour	Utilities	Lot Config	Land Slope	Neighborhood	Condition 1	Condition 2
0	109	533352170	60	RL	NaN	13517	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	Sawyer	RRAe	Norm
1	544	531379050	60	RL	43.0	11492	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	SawyerW	Norm	Norm
2	153	535304180	20	RL	68.0	7922	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm
3	318	916386060	60	RL	73.0	9802	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Timber	Norm	Norm
4	255	906425045	50	RL	82.0	14235	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	SawyerW	Norm	Norm

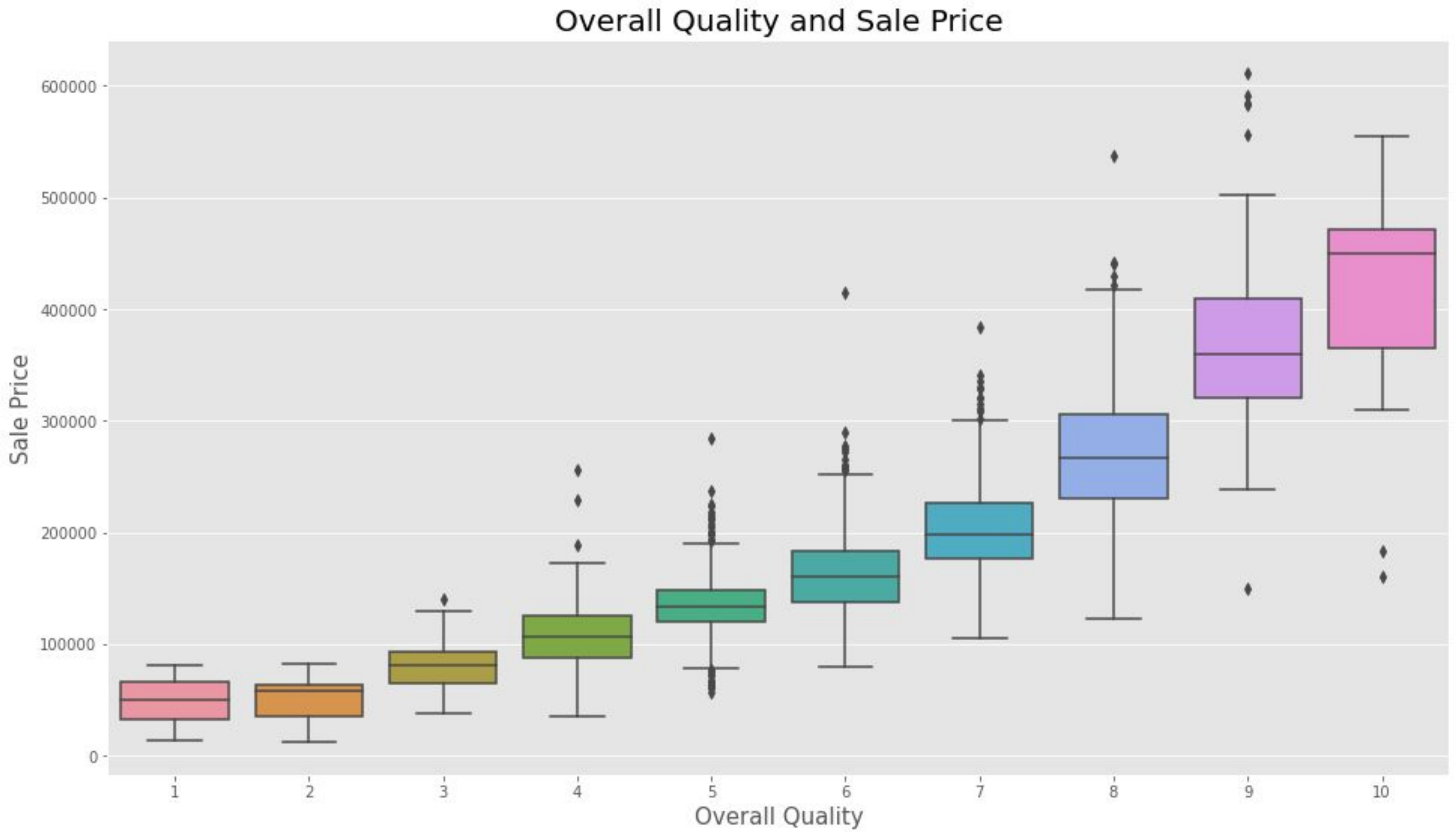
...

Data Exploration


Correlation of Numeric Features to Sale Price
SalePrice 1
Gr Liv Area 0.7
Garage Cars 0.65
Total Bsmt SF 0.63
1st Flr SF 0.62
Year Built 0.57
Year Remod/Add 0.55
Full Bath 0.54
Mas Vnr Area 0.5
TotRms AbvGrd 0.5
Fireplaces 0.47
BsmtFin SF 1 0.42
Open Porch SF 0.33
Lot Frontage 0.33
Wood Deck SF 0.33
Lot Area 0.3
Bsmt Full Bath 0.28
Half Bath 0.28
2nd Flr SF 0.25
Bsmt Unf SF 0.19
Bedroom AbvGr 0.14
Screen Porch 0.13
3Ssn Porch 0.049
Mo Sold 0.031
BsmtFin SF 2 0.016
Misc Val -0.0076
Yr Sold -0.014
Low Qual Fin SF -0.042
Bsmt Half Bath -0.046
MS SubClass -0.089
Kitchen AbvGr -0.13
Alley -0.14
Enclosed Porch -0.14
Fence -0.18
SalePrice
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00



Distribution of Housing Prices in Ames, IA ('07-'10)
Mean Sale Price
Number of Houses
0
50
100
150
200
250
300
350
400
0
50000
100000
150000
200000
250000
300000
350000
400000
450000
500000
550000
600000
Sale Price


Overall Quality and Sale Price
Sale Price
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
10
Overall Quality

Data Cleaning

Missing values imputation

- **Take a look to see how many null values need to be address**

```
df.isnull().sum().sort_values(ascending=False).head(27)
```

```
Pool QC          2042
Misc Feature     1986
Alley            1911
Fence            1651
Fireplace Qu     1000
Lot Frontage      330
Garage Finish     114
Garage Qual       114
Garage Yr Blt     114
Garage Cond       114
Garage Type       113
Bsmt Exposure      58
BsmtFin Type 2     56
Bsmt Cond          55
Bsmt Qual          55
BsmtFin Type 1     55
Mas Vnr Area       22
Mas Vnr Type       22
Bsmt Full Bath      2
Bsmt Half Bath      2
Garage Area         1
Garage Cars         1
Total Bsmt SF       1
Bsmt Unf SF         1
BsmtFin SF 2        1
BsmtFin SF 1        1
```

It seems like there are some features that

- contain **more than 1000 null values** (5 features)
- Others **below 1000 null values** (23 features)

First, we will gradually verify **what the missing information is interesting.**

- **Dealing with the worst offenders**

Pool QC is mostly null, then I will take a look at the values it does have

```
df['Pool QC'].value_counts()
```

```
Gd     4
TA     2
Fa     2
Ex     1
Name: Pool QC, dtype: int64
```

Pool Qc doesn’t contain any values for No pool, so I will assign those nulls as **‘NA’** to imply that these are houses that do not have pools.

Misc Feature (Miscellaneous feature) are uncommon features in a home

```
df['Misc Feature'].value_counts()
```

```
Shed    56
Gar2     4
Othr     3
TenC     1
Elev     1
Name: Misc Feature, dtype: int64
```

I think it’s a reasonable to again impute a value of **‘NA’** for the nulls here

Fence and Alley are two remaining features with mostly null values

```
fence_quality = ['MnPrv', 'GdPrv', 'GdWo', 'MnWw']
for quality in fence_quality:
    df.Fence = df.Fence.str.replace(quality, '1')
df.Fence.fillna(0, inplace=True)
df.Fence = df.Fence.apply(lambda x: int(x))
df.Fence.head(8)
```

```
alley_quality = ['Grvl', 'Pave']
for quality in alley_quality:
    df.Alley = df.Alley.str.replace(quality, '1')
df.Alley.fillna(0, inplace=True)
df.Alley = df.Alley.apply(lambda x: int(x))
df.Alley.head(8)
```

So, I will convert these both columns to a binary one where

- "1" indicates that **a property has a fence**
- "0" indicates that a **property has no fence**

Fireplace is another features with plenty of null values

```
df.drop(columns='Fireplace Qu', inplace=True)
```

Since we already have a numeric features, "**Fireplaces**" that indicate how many fireplaces are in each property, I feel comfortable dropping "**Fireplace Qu**" from the dataset

Garage-related Features

```
Garage Type
Garage Yr Blt
Garage Finish
Garage Cars
Garage Area
Garage Qual
Garage Cond
```

- **113 properties are missing garage-related values** → Imply that these properties do not have garage so **I will fill those values as 'NA'**
- Compare **the years that garages were built** with t**he years the properties were built** to see how many garages were built after the original construction
  - **362 properties** have garages with different build years than the property itself
  - Low enough then I feel comfortable **dropping the 'Garage Yr Blt'**
- **Fill "NA"** of the four remaining garage features **(Type, Finish, Qual, Cond)**

Basement-related Features

```
Bsmt Qual
Bsmt Cond
Bsmt Exposure
BsmtFin Type 1
BsmtFin SF 1
BsmtFin Type 2
BsmtFin SF 2
Bsmt Unf SF
Total Bsmt SF
Bsmt Full Bath
Bsmt Half Bath
```

- Since there don't seem to be any stray values for these features, Fill "NA" to the null categories

```
'Bsmt Qual'] = 'NA'
'Bsmt Cond'] = 'NA'
'Bsmt Exposure'] = 'NA'
'BsmtFin Type 1'] = 'NA'
'BsmtFin Type 2'] = 'NA'
```

- **Total Bsmt SF = BsmtFin SF 1 + BsmtFin SF2 + Bsmt Unf_SF → Drop the components**
- **Bsmt Full Bath & Bsnt Half Bath → Manually set those values to 0**

Last Miscellaneous Nulls

```
df.isnull().sum().sort_values(ascending=False).head()
```

```
Lot Frontage    330
Mas Vnr Type     22
Mas Vnr Area     22
SalePrice         0
Foundation        0
dtype: int64
```

```
df['Mas Vnr Type'].value_counts()
```

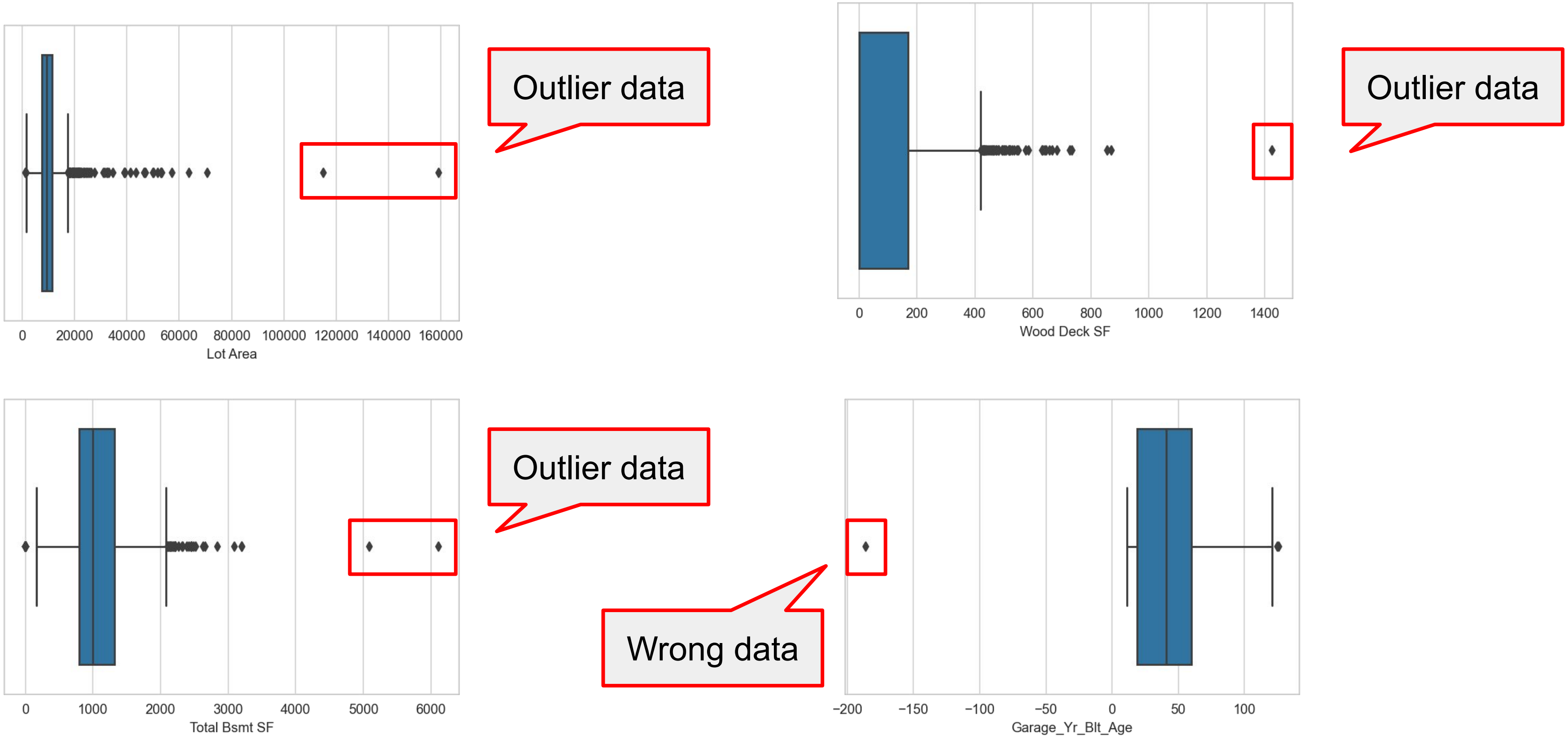
```
None       1216
BrkFace     628
Stone       168
BrkCmn       13
Name: Mas Vnr Type, dtype: int64
```

- Since most properties have no masonry work, I'll impute the mode of **'None'** and **0** for **Mas Vnr Type and Mas Vnr Area.**

```
df['Lot Frontage'].fillna(value=df['Lot Frontage'].mean(), inplace=True)
```

- We have too many nulls for **Lot Frontage** to **drop those properties from our dataset, but not enough nulls to drop the feature entirely.**
- Since it is unlikely that a property truly has zero linear feet of **Lot Frontage I will impute the mean value.**

Deleting some outlier data



Fixing nonsensical values...

- Replacing Garage Yr Blt value that is in the future (2207) with the year the house was built.

```
count    1937.000000
mean     1978.707796
std        25.441094
min      1895.000000
25%      1961.000000
50%      1980.000000
75%      2002.000000
max      2207.000000
Name: Garage Yr Blt, dtype: float64
```

Feature Engineering

Converting Year columns to age...

Converting...

- 'Year Built'
- 'Year Remod/Add'
- 'Garage Yr Blt'
- 'Yr Sold'

from actual Years to age

```
def convert_yrs_cols(df):
    yr_cols = ['Year Built','Year Remod/Add','Garage Yr Blt','Yr Sold']
    df_copy = df.copy(deep=True)
    for col in yr_cols:
        df_copy[col] = 2011 - df_copy[col]
    return df_copy
```

Adding Polynomial Features

1. Getting a list of features that has a strong correlation (0.6<) against SalePrice.
2. Using sklearn's PolynomialFeatures to create interaction terms between these features with degree of freedom = 3.

Overall Qual^3	Overall Qual^2 Gr Liv Area	Overall Qual^2 Garage Area	Overall Qual^2 Total Bsmt SF	Overall Qual^2 1st Flr SF	Overall Qual^2 Exter Qual_TA	Overall Qual Gr Liv Area^2	Overall Qual Gr Liv Area Garage Area	Overall Qual Gr Liv Area Total Bsmt SF	Overall Qual Gr Liv Area 1st Flr SF	Overall Qual Gr Liv Area Exter Qual_TA	Overall Qual Garage Area^2
216.0	53244.0	17100.0	26100.0	26100.0	0.0	13124646.0	4215150.0	6433650.0	6433650.0	0.0	1353750.0
343.0	103978.0	27391.0	44737.0	44737.0	0.0	31520188.0	8303386.0	13561702.0	13561702.0	0.0	2187367.0
125.0	26425.0	6150.0	26425.0	26425.0	25.0	5586245.0	1300110.0	5586245.0	5586245.0	5285.0	302580.0
125.0	36100.0	10000.0	9600.0	18600.0	25.0	10425680.0	2888000.0	2772480.0	5371680.0	7220.0	800000.0
216.0	52020.0	17424.0	24336.0	29916.0	36.0	12528150.0	4196280.0	5860920.0	7204770.0	8670.0	1405536.0

OneHotEncoding (Categorical data)

	MS Zoning	Street	Land Contour	Utilities	Condition 2	Roof Matl	Exter Qual	Bsmt Qual	Heating	Kitchen Qual	Paved Drive
0	RL	Pave	Lvl	AllPub	Norm	CompShg	Gd	TA	GasA	Gd	Y
1	RL	Pave	Lvl	AllPub	Norm	CompShg	Gd	Gd	GasA	Gd	Y
2	RL	Pave	Lvl	AllPub	Norm	CompShg	TA	TA	GasA	Gd	Y
3	RL	Pave	Lvl	AllPub	Norm	CompShg	TA	Gd	GasA	TA	Y
4	RL	Pave	Lvl	AllPub	Norm	CompShg	TA	Fa	GasA	TA	N

	MS Zoning_A (agr)	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	...
0	0	0	0	0	0	1	0	0	1	0	...
1	0	0	0	0	0	1	0	0	1	0	...
2	0	0	0	0	0	1	0	0	1	0	...
3	0	0	0	0	0	1	0	0	1	0	...
4	0	0	0	0	0	1	0	0	1	0	...

5 rows × 51 columns

Target Encoding (Categorical data)

Target Encoding replaces a categorical value with the mean of the **target** variable

	Lot Shape	Utilities	Land Slope	Exter Qual	Exter Cond	Bsmt Qual	Bsmt Cond	Heating QC	Kitchen Qual	Fireplace Qu	Garage Qual	Garage Cond	Pool QC	Bsmt Exposure	BsmtFin Type 1	BsmtFin Type 2	Electrical
0	IR1	AllPub	Gtl	Gd	TA	TA	TA	Ex	Gd	Gd	TA	TA	Gd	No	GLQ	Unf	SBrkr
1	IR1	AllPub	Gtl	Gd	TA	Gd	TA	Ex	Gd	TA	TA	TA	Gd	No	GLQ	Unf	SBrkr
2	Reg	AllPub	Gtl	TA	Gd	TA	TA	TA	Gd	Gd	TA	TA	Gd	No	GLQ	Unf	SBrkr
3	Reg	AllPub	Gtl	TA	TA	Gd	TA	Gd	TA	Gd	TA	TA	Gd	No	Unf	Unf	SBrkr
4	IR1	AllPub	Gtl	TA	TA	Fa	Gd	TA	TA	Gd	TA	TA	Gd	No	Unf	Unf	SBrkr

	Lot Shape	Utilities	Land Slope	Exter Qual	Exter Cond	Bsmt Qual	Bsmt Cond	Heating QC	Kitchen Qual
0	211848.670520	181551.602245	180358.476703	230802.484935	185258.202475	138023.926752	181760.117522	216027.607512	211629.451613
1	211848.670520	181551.602245	180358.476703	230802.484935	185258.202475	202537.582176	181760.117522	216027.607512	211629.451613
2	162925.812355	181551.602245	180358.476703	143270.978348	167623.023256	138023.926752	181760.117522	138986.705193	211629.451613
3	162925.812355	181551.602245	180358.476703	143270.978348	185258.202475	202537.582176	181760.117522	160174.009404	139501.607450
4	211848.670520	181551.602245	180358.476703	143270.978348	185258.202475	107752.166667	223969.550562	138986.705193	139501.607450

Standard scaling the features

Using StandardScaler to standardise these numerical features to avoid the model being sensitive to features with bigger magnitudes.

	MS SubClass	Lot Frontage	Lot Area	Overall Qual	Overall Cond	Year Built	Year Remod/Add	Mas Vnr Area	BsmtFin SF 1	BsmtFin SF 2	Bsmt Unf SF	Total Bsmt SF	1st Flr SF	2nd Flr SF	Low Qual Fin SF	Gr Liv Area	Bsmt Full Bath	Bsmt Half Bath
0	60	69.0552	13517	6	8	35	6	289.0	533.0	0.0	192.0	725.0	725	754	0	1479	0.0	0.0
1	60	43.0000	11492	7	5	15	14	132.0	637.0	0.0	276.0	913.0	913	1209	0	2122	1.0	0.0
2	20	68.0000	7922	5	7	58	4	0.0	731.0	0.0	326.0	1057.0	1057	0	0	1057	1.0	0.0
3	60	73.0000	9802	5	5	5	4	0.0	0.0	0.0	384.0	384.0	744	700	0	1444	0.0	0.0
4	50	82.0000	14235	6	8	111	18	0.0	0.0	0.0	676.0	676.0	831	614	0	1445	0.0	0.0

	MS SubClass	Lot Frontage	Lot Area	Overall Qual	Overall Cond	Year Built	Year Remod/Add	Mas Vnr Area	BsmtFin SF 1	BsmtFin SF 2	Bsmt Unf SF	Total Bsmt SF	1st Flr SF
0	0.069866	0.000000	0.512071	-0.078644	2.207728	-0.142227	-0.989479	1.089794	0.197117	-0.290862	-0.844026	-0.739359	-1.108838
1	0.069866	-1.223182	0.211664	0.622656	-0.509102	-0.805126	-0.609090	0.187536	0.422688	-0.290862	-0.655208	-0.321322	-0.634510
2	-0.864413	-0.049537	-0.317944	-0.779944	1.302118	0.620106	-1.084576	-0.571050	0.626569	-0.290862	-0.542817	-0.001124	-0.271195
3	0.069866	0.185192	-0.039047	-0.779944	-0.509102	-1.136575	-1.084576	-0.571050	-0.958932	-0.290862	-0.412443	-1.497605	-1.060900
4	-0.163704	0.607704	0.618586	-0.078644	2.207728	2.376787	-0.418896	-0.571050	-0.958932	-0.290862	0.243923	-0.848315	-0.841397

Feature Selection

Feature selection using Lasso/Elastic Net

Lasso/Elastic Net has the property of being able to eliminate features that are not important by setting their corresponding coefficients to zero.

```
coef = list(zip(elasticnet.coef_,X.columns))
important_feats = sorted(coef, key=lambda x: np.abs(x[0]),reverse=True)
```

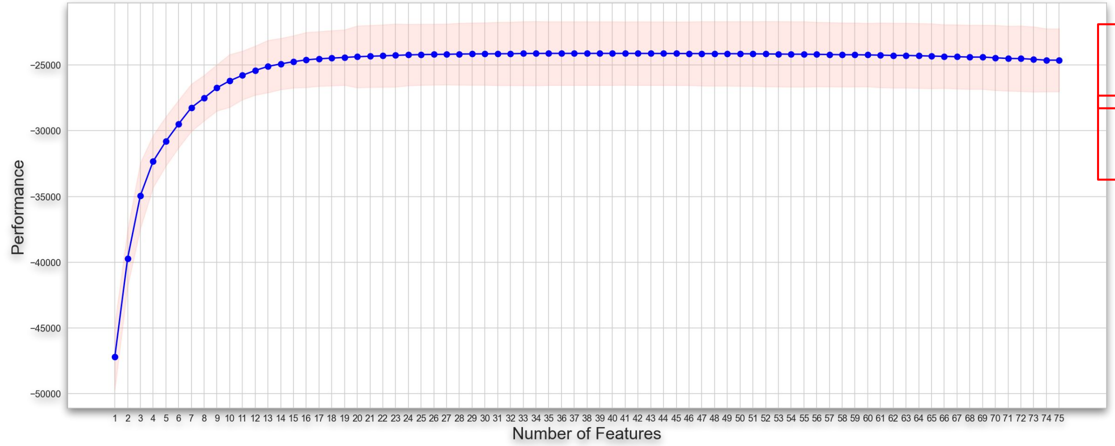
```
(19505.015532660454, 'Gr Liv Area'),
(18976.496977384428, 'Overall Qual^2 Gr Liv Area'),
(-9979.30973814250З, 'Gr Liv Area 1st Flr SF^2'),
(8499.53943390022l, 'Kitchen Qual^2 Gr Liv Area')]
```

```
(0.0, 'Roof Matl_CompShg'),
(0.0, 'Roof Matl_Membran'),
(-0.0, 'Roof Matl_Tar&Grv'),
(0.0, 'Roof Matl_WdShake'),
(0.0, 'Mas Vnr Type_BrkFace'),
(-0.0, 'Mas Vnr Type_None'),
(0.0, 'Mas Vnr Type_Stone'),
(0.0, 'Misc Feature_Gar2'),
(-0.0, 'Misc Feature_TenC'),
(0.0, 'Exter Qual^2'),
(0.0, 'Exter Qual Bsmt Qual'),
(0.0, 'Bsmt Qual^2'),
(0.0, 'Bsmt Qual Garage Area'),
```

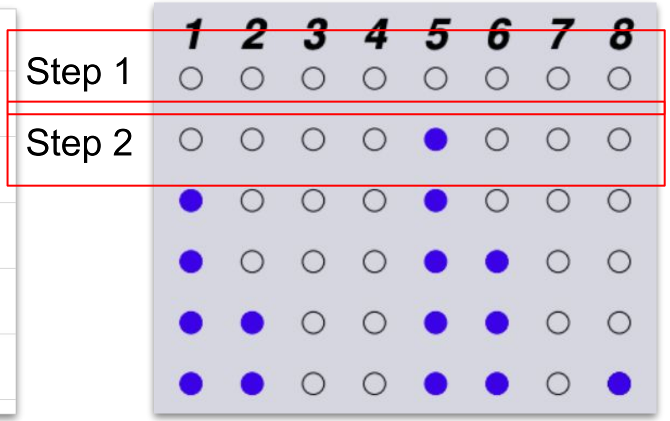
Sequential Forward Selection (SFS)

1. First, the best single feature is selected (i.e., using some criterion function).
2. Then, pairs of features are formed using one of the remaining features and this best feature, and the best pair is selected.
3. Next, triplets of features are formed using one of the remaining features and these two best features, and the best triplet is selected.
4. This procedure continues until a predefined number of features are selected.

Note : SFS performs best when the optimal subset is small.



Number of Features



Evaluation Metrics

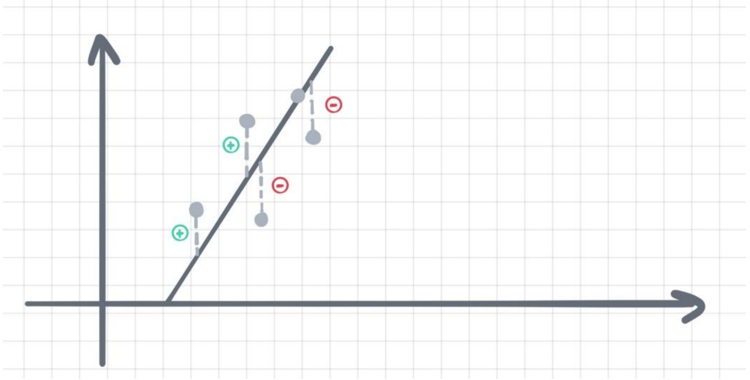
Root Mean Square Error & R-Squared

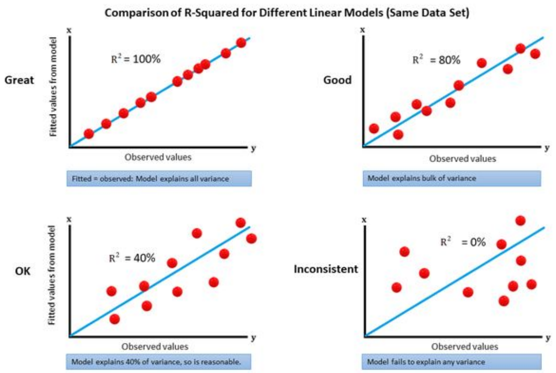
Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are.

R-squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

$$RMSE = \sqrt{\frac{1}{n} * \sum (prediction - actual)^2}$$

$$R^2 = 1 - \left(\frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2} \right)$$





The Models

Regression Models

The models that were used include…

- Linear Regression
- Ridge
- Lasso
- Elastic Net

Best Performing Model

Elastic Net

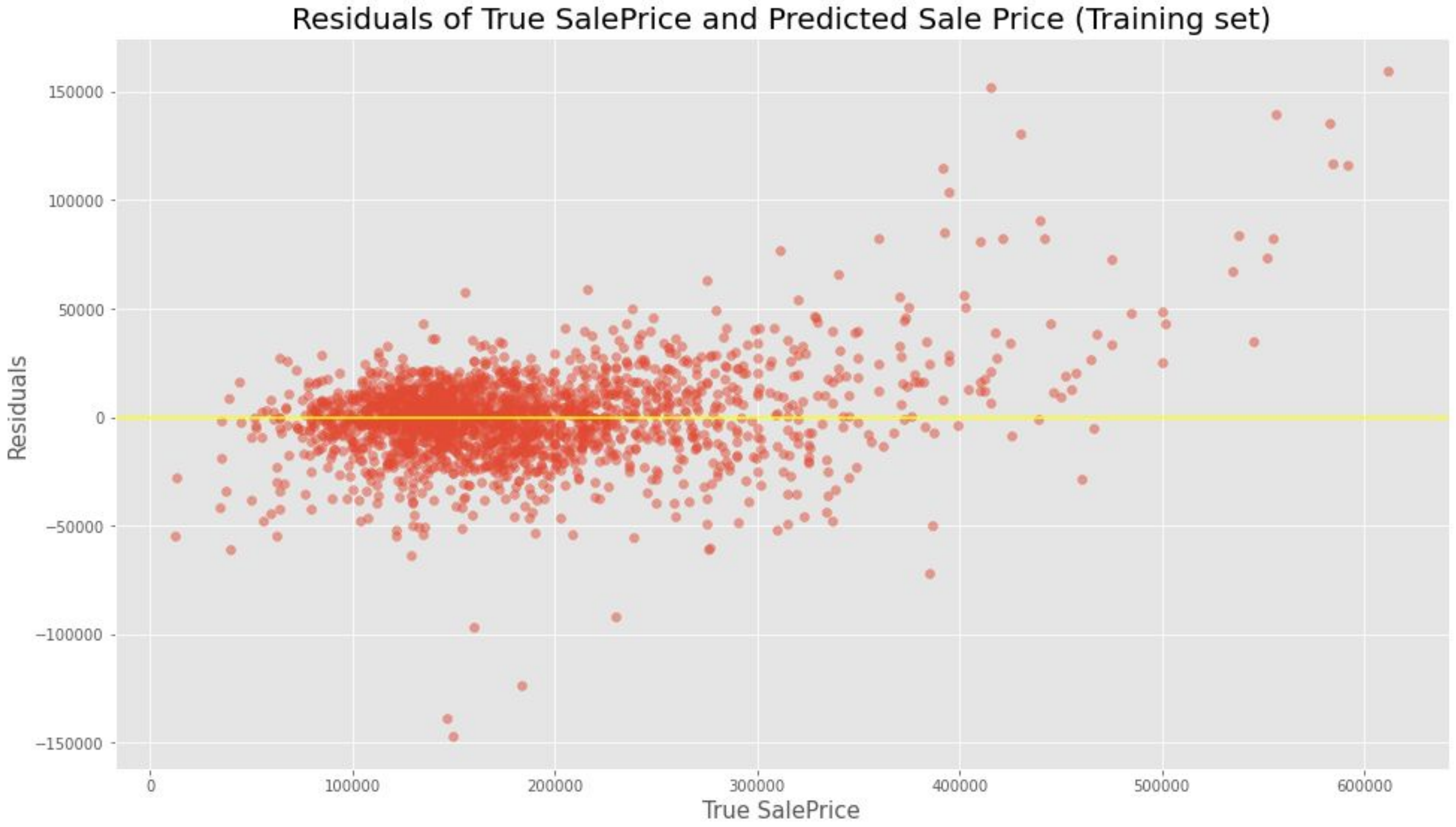
Model Description:

- Missing values have been imputed with *mean (numerical)* and *mode (categorical)*
- Categories columns that are ordinal have been encoded using *Target Encoder* (Mean Encoder)
- Categories columns that aren't ordinal have been encoded using *OneHotEncoder*
- *Polynomial Features* (degree = 3) performed on columns that have correlation of > 0.6 against Sale Price.
- Features *standardised*
- *Backward Elimination* to get important features (99 from 360 features)
- Elastic Net (best alpha = 674, best l1 ratio = 1)

Evaluation Metrics

Training R2	Cross Val R2	Training RMSE	Cross Val RMSE	Kaggle Private RMSE
0.930	0.905	20921	24251	25412


True Sale Price against Predicted Sale Price (Training set)
Predicted SalePrice
True SalePrice
500000
400000
300000
200000
100000
0
100000
200000
300000
400000
500000
600000


Residuals of True SalePrice and Predicted Sale Price (Training set)
Residuals
True SalePrice
150000
100000
50000
0
−50000
−100000
−150000
0
100000
200000
300000
400000
500000
600000

Conclusion

Conclusion

- Features that are important include
  - Above Ground Living Area
  - Overall Quality
  - Total square feet of basement area
- Features that negatively impact the house prices include
  - Year Built
  - Year Remod/Add
- Our model was able to predict the house prices with +/- 25000 error on average on unseen data.
- The model could be used by real estate businesses to give them an estimate of the prices of new properties entering the market.