Project 2: Ames Housing Data

Zachary Katsnelson
General Assembly, DSI-10, Toronto
Dec. 17 2020

Ames Housing Data: Defining the Problem

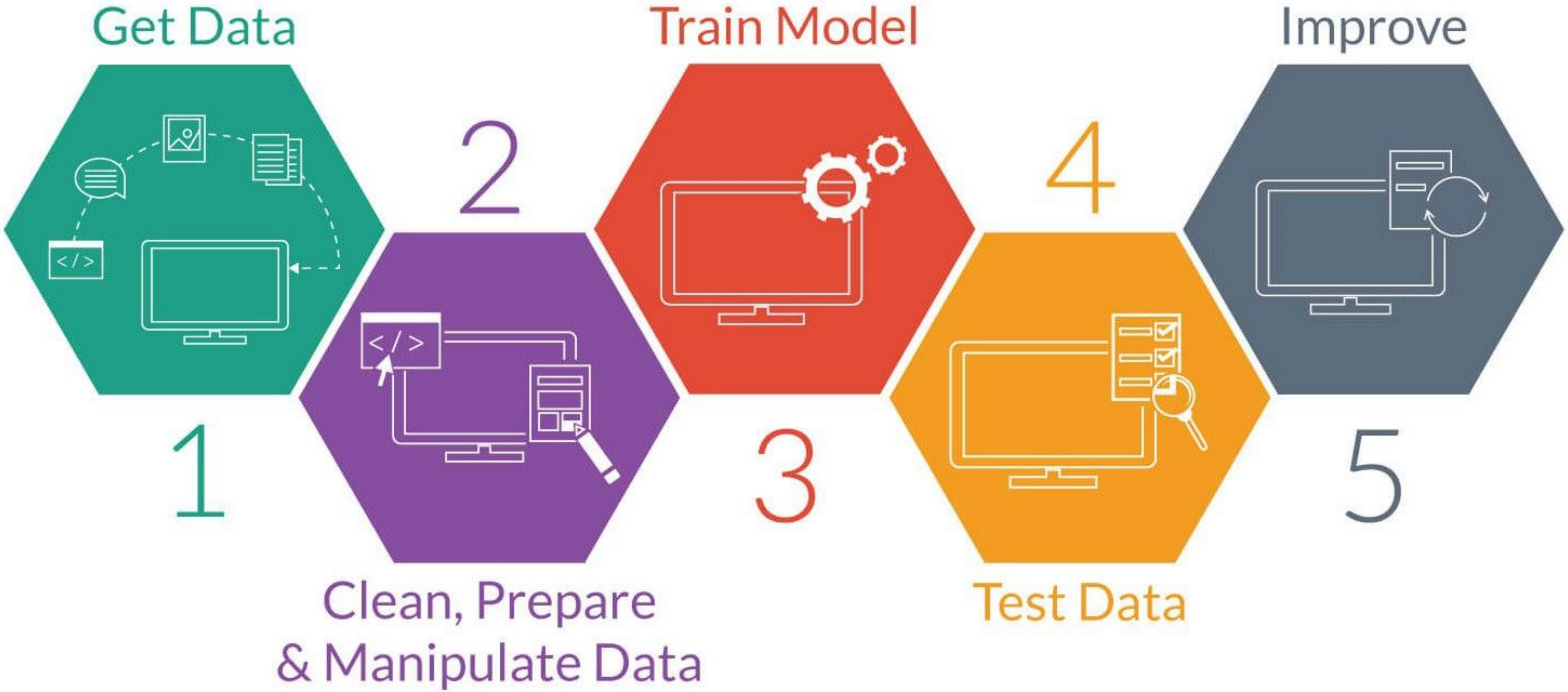
What is the problem?

1. Determine the best model for predicting sales prices of houses in Ames, Iowa.
2. Find features that best correlate to predicted sales price.

How are we going to try to solve it?

Through investigation and utilization of the Ames housing dataset with over 70 columns of different features relating to houses.

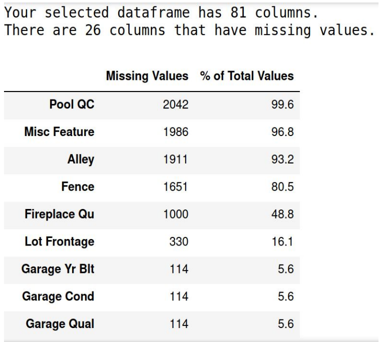
Machine Learning Process



Step 1: Data Cleaning and Encoding

Null Values

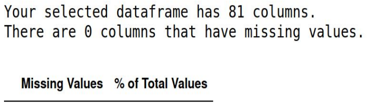
- Replace all ordinal values with ‘None’.
- Replace all continuous values with 0.
- Total null values should equal 0.



```
Your selected dataframe has 81 columns.
There are 26 columns that have missing values.
```

	Missing Values	% of Total Values
Pool QC	2042	99.6
Misc Feature	1986	96.8
Alley	1911	93.2
Fence	1651	80.5
Fireplace Qu	1000	48.8
Lot Frontage	330	16.1
Garage Yr Blt	114	5.6
Garage Cond	114	5.6
Garage Qual	114	5.6

→

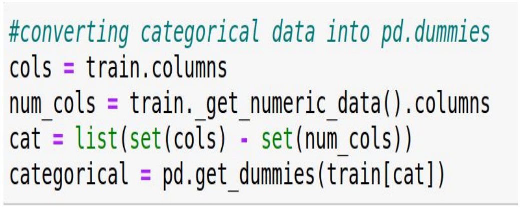


```
Your selected dataframe has 81 columns.
There are 0 columns that have missing values.
```

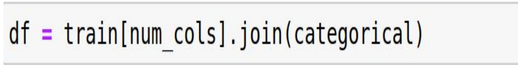
	Missing Values	% of Total Values

Transforming Categorical Variables using Pandas Dummies

- Create new dataset (df) that only contains numerical data.
- Convert all categorical data into dummy/indicator variables.



```
#converting categorical data into pd.dummies
cols = train.columns
num_cols = train._get_numeric_data().columns
cat = list(set(cols) - set(num_cols))
categorical = pd.get_dummies(train[cat])
```



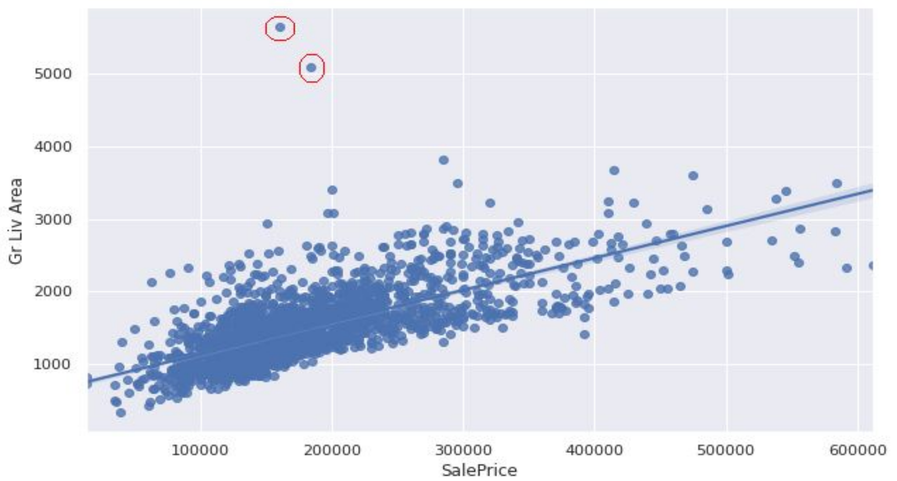
```
df = train[num_cols].join(categorical)
```

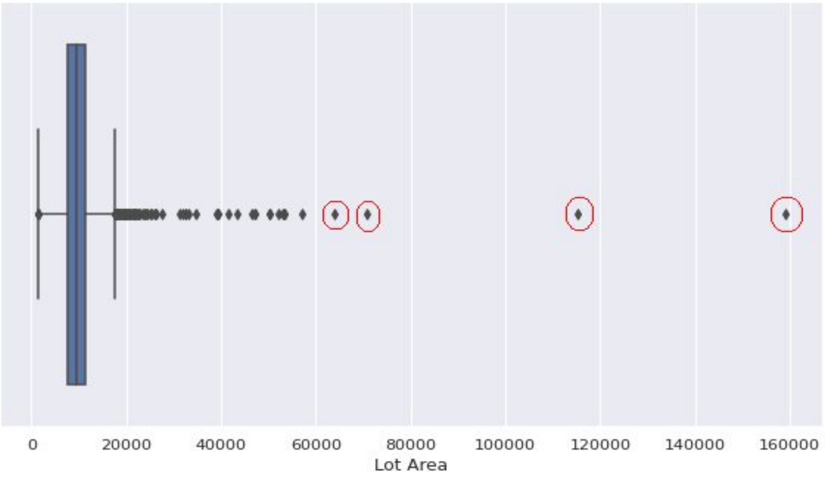
Step 1: Data Cleaning and Encoding (cont'd)

- Remove 2 Outliers in Gr. Liv Area.
- Remove any house with a Lot Area exceeding 60,000 square feet.

Results:

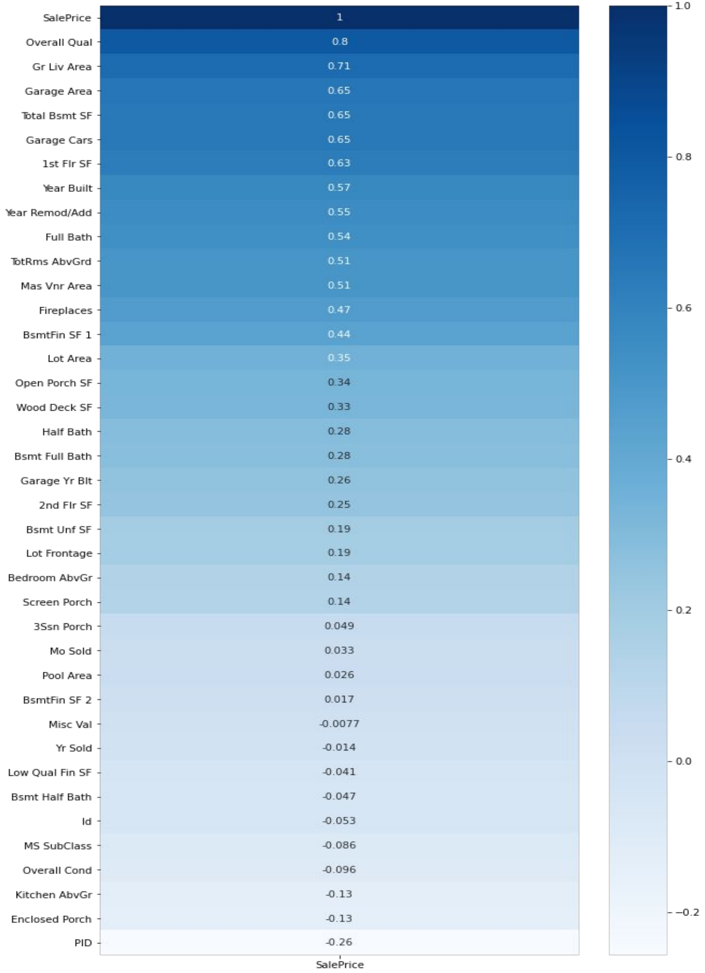
Cross-validation mean score (of cv = 10, using Linear Regression) improved from **0.77** to **0.86**.

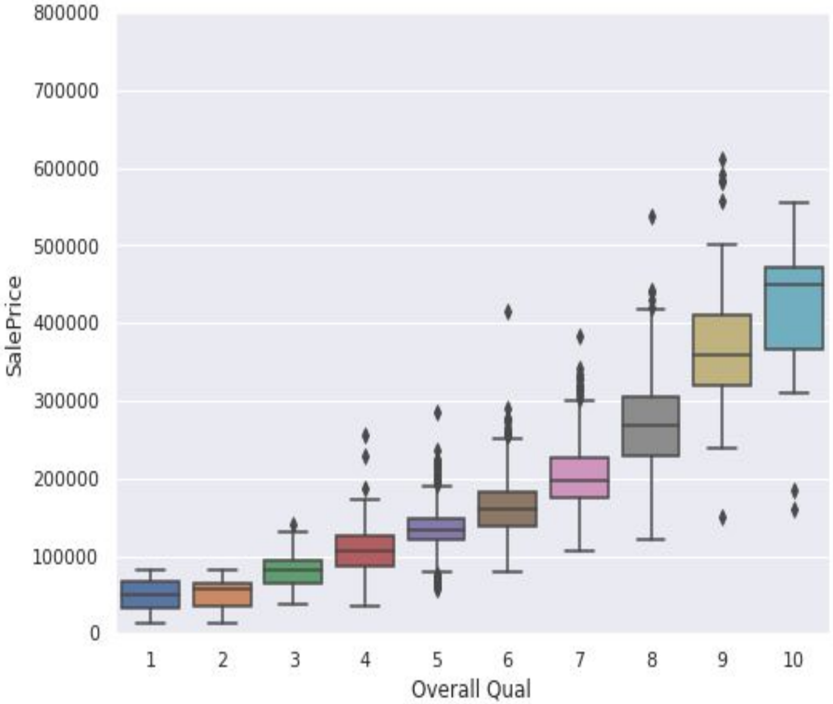


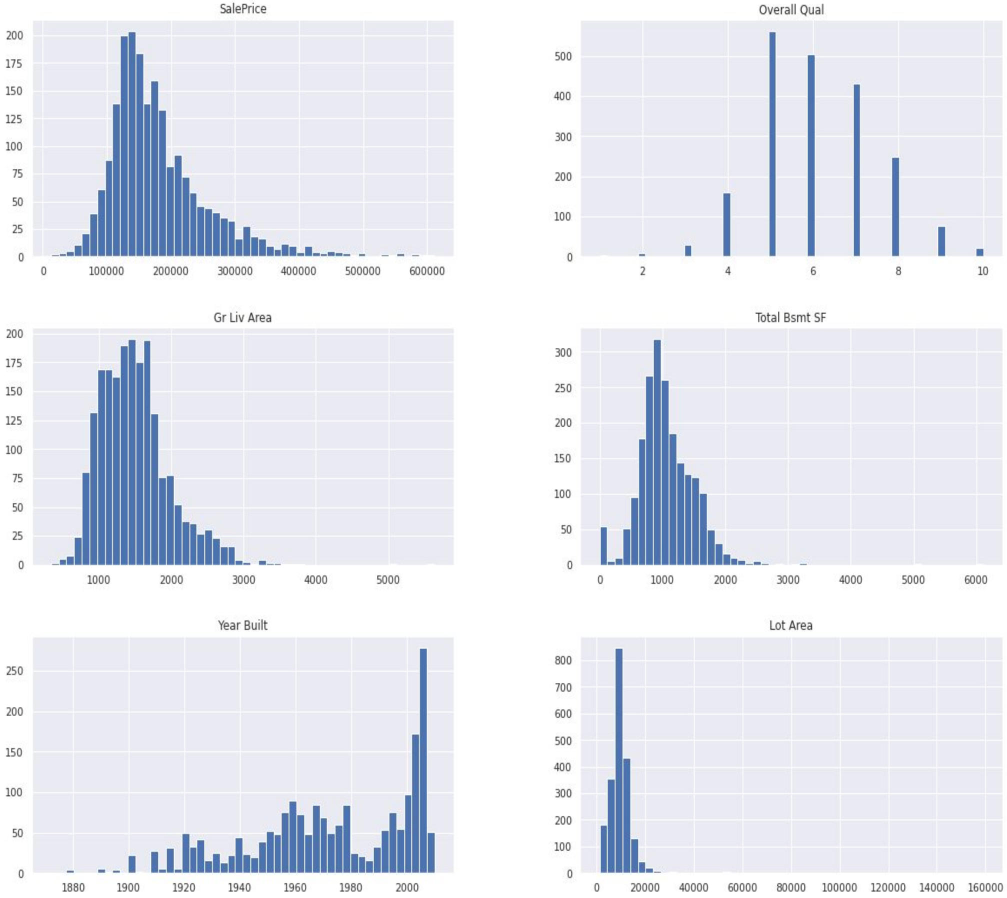


Step 2: Data Visualization and Correlation

- Overall Quality of home has greatest positive correlation with sales price.
- There are multiple SF columns, pertaining to the total, 1st floor and 2nd floor, and porch, respectively.
- Multiple columns pertaining to baths.



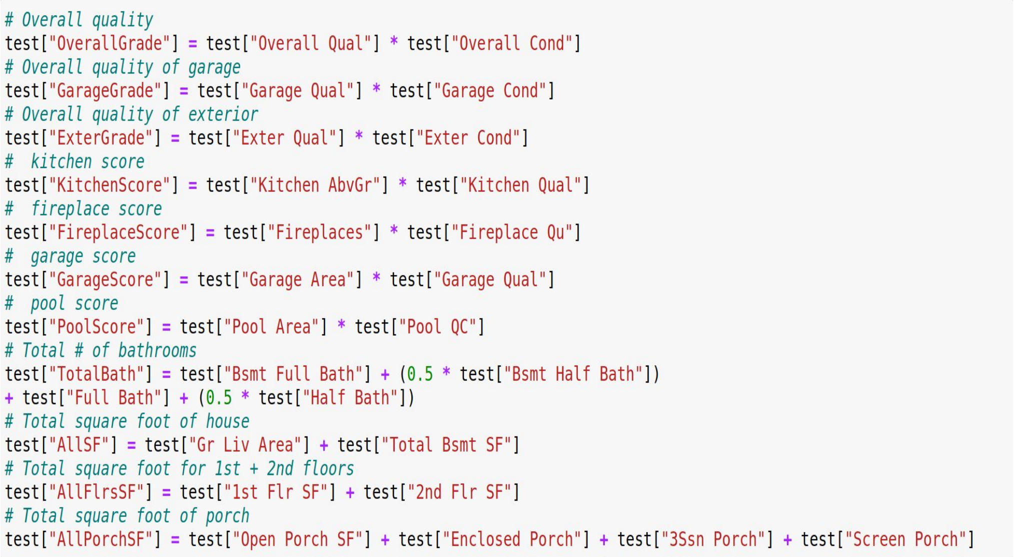

SalePrice
800000
700000
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
10
Overall Qual


SalePrice
0
100000
200000
300000
400000
500000
600000
0
25
50
75
100
125
150
175
200
Overall Qual
2
4
6
8
10
0
100
200
300
400
500
Gr Liv Area
1000
2000
3000
4000
5000
Total Bsmt SF
0
1000
2000
3000
4000
5000
6000
Year Built
1880
1900
1920
1940
1960
1980
2000
Lot Area
0
20000
40000
60000
80000
100000
120000
140000
160000

Step 3: Feature Engineering

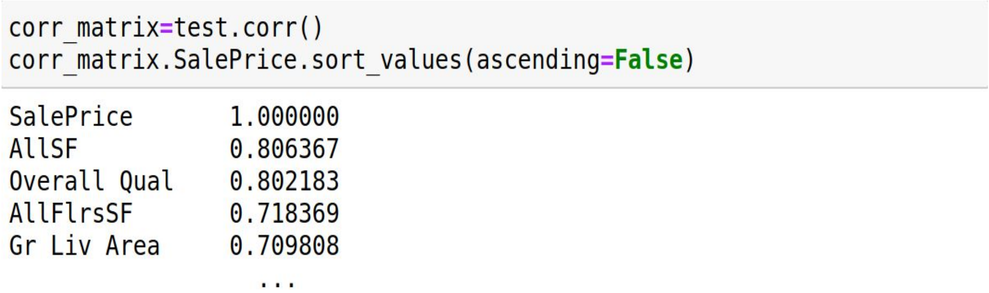
The Features that gave the best ***signal*** for the model were:

- Overall Grade = Overall Quality * Overall Condition
- Total Bath = Basement Full Bath + (0.5 * Basement Half Bath) + Full Bath + (0.5 * Half Bath).
- AllSF = Gr Living Area + Total Bsmt Square foot
- AllFloorsSF = Total SF for 1st + 2nd floors
- ALLPorchSF= Total SF for porch



```
# Overall quality
test["OverallGrade"] = test["Overall Qual"] * test["Overall Cond"]
# Overall quality of garage
test["GarageGrade"] = test["Garage Qual"] * test["Garage Cond"]
# Overall quality of exterior
test["ExterGrade"] = test["Exter Qual"] * test["Exter Cond"]
#  kitchen score
test["KitchenScore"] = test["Kitchen AbvGr"] * test["Kitchen Qual"]
#  fireplace score
test["FireplaceScore"] = test["Fireplaces"] * test["Fireplace Qu"]
#  garage score
test["GarageScore"] = test["Garage Area"] * test["Garage Qual"]
#  pool score
test["PoolScore"] = test["Pool Area"] * test["Pool QC"]
# Total # of bathrooms
test["TotalBath"] = test["Bsmt Full Bath"] + (0.5 * test["Bsmt Half Bath"])
+ test["Full Bath"] + (0.5 * test["Half Bath"])
# Total square foot of house
test["AllSF"] = test["Gr Liv Area"] + test["Total Bsmt SF"]
# Total square foot for 1st + 2nd floors
test["AllFlrsSF"] = test["1st Flr SF"] + test["2nd Flr SF"]
# Total square foot of porch
test["AllPorchSF"] = test["Open Porch SF"] + test["Enclosed Porch"] + test["3Ssn Porch"] + test["Screen Porch"]
```

A test done using orbital encoder on all values. Coefficients of each were then measured and tested using baseline model of LinearRegression().



```
corr_matrix=test.corr()
corr_matrix.SalePrice.sort_values(ascending=False)
```

```
SalePrice       1.000000
AllSF           0.806367
Overall Qual    0.802183
AllFlrsSF       0.718369
Gr Liv Area     0.709808
                 ...
```

Step 4: Modelling

We will use GridSearchCV to find the best model, using the cross-validated R2 score as our scoring metric. We will compare and test the following models, each with a scaled dataset using StandardScaler():

1. Linear Regression
2. K-Neighbours Regressor. Parameters = n_neighbours, weightings, distance
3. Ridge Regression. Parameters = alpha
4. Lasso Regression. Parameters = alpha
5. Elastic Net. Parameters = alpha, l1_ratio

Final Scores

Model	Best Score (R2, cv = 10)	Best Parameters
LinearRegression	0.865	Default
KNeighboursRegressor	0.813	(5, 'distance', 1)
Ridge	0.880	{'alpha': 1000}
Lasso	0.881	{'alpha': 1000}
ElasticNet	0.882	{'alpha': 1, 'l1_ratio': 0.5}