Nick Calow

Ames, Iowa Housing Analysis

Overview

- Data Science Problem: How can one predict future sale prices based upon housing data in Ames, Iowa?
- Using model building techniques and some experimental feature selection, the problem has begun to have been solved

Initial Data Cleaning

- So many ordinal variables
- Numeric values: good analysis

```
Exter_dict = {'Po': 0, 'Fa': 1, 'TA': 2, 'Gd': 3, 'Ex': 4}
#First set of replacements
train['Exter Qual'].replace(Exter_dict, inplace = True)
train['Exter Cond'].replace(Exter_dict, inplace = True)
train['Heating QC'].replace(Exter_dict, inplace = True)
train['Kitchen Qual'].replace(Exter_dict, inplace = True)

Bsmt_dict = {'NA': 0, 'Po': 1, 'Fa': 2, 'TA': 3, 'Gd': 4, 'Ex': 5}
#Second set
train['Bsmt Qual'].replace(Bsmt_dict, inplace = True)
train['Bsmt Cond'].replace(Bsmt_dict, inplace = True)
train['Fireplace Qu'].replace(Bsmt_dict, inplace = True)
train['Garage Qual'].replace(Bsmt_dict, inplace = True)
train['Garage Cond'].replace(Bsmt_dict, inplace = True)
train['Pool QC'].replace(Bsmt_dict, inplace = True)

Bsmt_ex_dict = {'NA': 0, 'No': 1, 'Mn': 2, 'Av': 3, 'Gd': 4}
#First special case set
train['Bsmt Exposure'].replace(Bsmt_ex_dict, inplace = True)

Func_dict = {'Sal': 0, 'Sev': 1, 'Maj2': 2, 'Maj1': 3,
             'Mod': 4, 'Min2': 5, 'Min1': 6, 'Typ': 7}
#Second special case
train['Functional'].replace(Func_dict, inplace = True)

Fence_dict = {'NA': 0, 'MnWw': 1, 'GdWo': 2, 'MnPrv': 3, 'GdPrv': 4}
#Third
train['Fence'].replace(Fence_dict, inplace = True)

Lot_dict = {'IR3': 0, 'IR2': 1, 'IR1': 2, 'Reg': 3}
#Fourth
train['Lot Shape'].replace(Lot_dict, inplace = True)
```

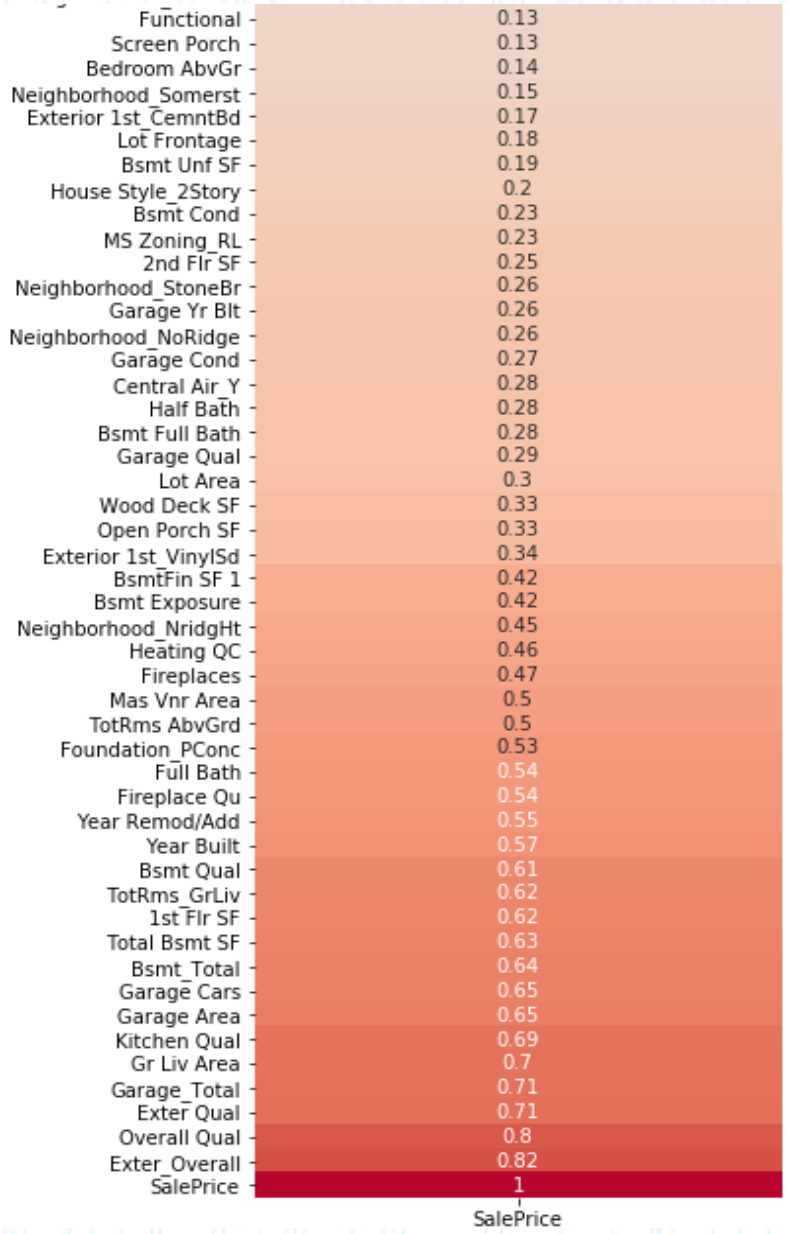
Feature Engineering

- Garage and Basement
- Interesting interactions
- Few dummies(Neighborhood, House style, etc.)

```
1
2  train['Bsmt_Total'] = train['Bsmt Qual'] * train['Bsmt Cond'] *
3                        train['Bsmt Exposure'] * train['Total Bsmt SF']
4
5
6  train['Garage_Total'] = train['Garage Yr Blt'] * train['Garage Cars'] *
7                          train['Garage Area'] * train['Garage Qual'] *
8                          train['Garage Cond']
9
10 train['TotRms_GrLiv'] = train['TotRms AbvGrd'] * train['Gr Liv Area']
11
12 train['Exter_Overall'] = train['Exter Qual'] * train['Overall Qual']
```

Initial Feature Selection

- Slice from a larger Heatmap
- Some high interactions
- Some trouble finding the right mix

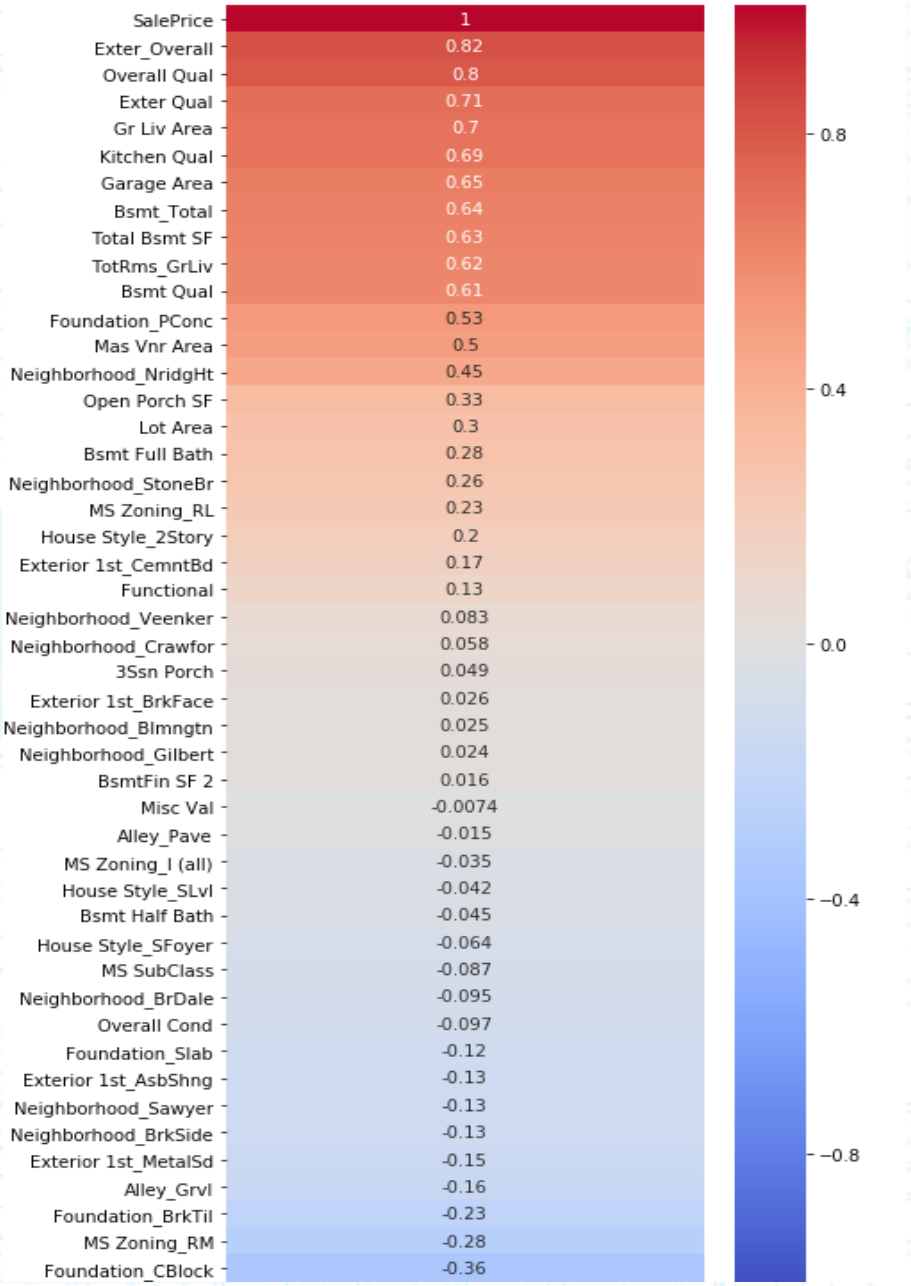


“The Experiment”

```
ideal_r2 = 0
r2_features = []
ideal_rmse = 99999999
rmse_features = []
y = train['SalePrice']
feature_length = len(features_init)#Every possible numeric feature
print('start')
for n in range(20, 50): # I didn't actually want to run it for 24 hours.
    counter = 1 #To be comprehensive, the range would be 1 to feature_length. Tuning.
    while counter <= (feature_length * n):
        unique_features = []
        features = features_init.sample(n) #Random sample of n features
        X = train[features]

        X_train, X_test, y_train, y_test = train_test_split(X,y,random_state = 42)
        model_ridge = RidgeCV(cv = 5)
        ss = StandardScaler() #Split and fit
        ss.fit(X_train)
        X_train_sc = ss.transform(X_train)
        X_test_sc = ss.transform(X_test)
        model_ridge.fit(X_train_sc, y_train)

        r2_temp = model_ridge.score(X_test_sc, y_test) #R2 value
        rmse_temp = RT_MSE(model_ridge, X_test_sc, y_test) #Root mean squared error

        if r2_temp > ideal_r2: #Tested two different values for the minimum
            r2_features = features
            ideal_r2 = r2_temp
        if rmse_temp < ideal_rmse:
            rmse_features = features
            ideal_rmse = rmse_temp

        if not features in unique_features: #Makes sure every combination is tried
            unique_features.append(features)
            counter += 1

    print(n)   #Track progress
print(ideal_r2)
print(ideal_rmse)
```

Results

- The wonders of black box programming.
- Other metrics at play here
- R^2 ended up being around 90%(with log transformation)



Conclusions

- Points for consideration:
  - Strong foundation
  - Large area
  - Right neighborhood
  - High quality
- The best solution may not be readily apparent.