GENERAL ASSEMBLY

DSI - Project 2

Ames Iowa Housing Price Model

Presented by

Suthatta Dontriros

Outline

Problem Statement

Modeling Procedure

- Data Cleaning
- Feature Engineering
- Modeling

Findings

Recommendations

Background

I work for the estate consultant who gives the recommendation about the house price. My customer is the estate company who want to invest in the housing market. The house demand growth expands every year in Iowa, therefore my customers want to know how to increase the house price and which features have high value of house.

Problem Statement

Can we use data to predict housing prices ?

What are those features and what is their impact?

Missing Values

Data Cleaning and EDA

- Cleaning null data
- Removing outlier

Pre-processing and Feature Engineering

- OneHotEncoder()
- Train-test-split
- StandardScaler()
- Log transformation

Modeling

- Model feature selection
- Linear Regression, Ridge, Lasso
- Model evaluation

Missing Values

Missing value from train and test data

	training
pool qc	2042
misc feature	1986
alley	1911
fence	1651
fireplace qu	1000
lot frontage	330
garage finish	114
garage cond	114
garage qual	114
garage yr blt	114
garage type	113
bsmt exposure	58
bsmtfin type 2	56
bsmtfin type 1	55
bsmt cond	55
bsmt qual	55
mas vnr type	22
mas vnr area	22
bsmt half bath	2
bsmt full bath	2
garage cars	1
garage area	1
bsmt unf sf	1
bsmtfin sf 2	1
total bsmt sf	1
bsmtfin sf 1	1

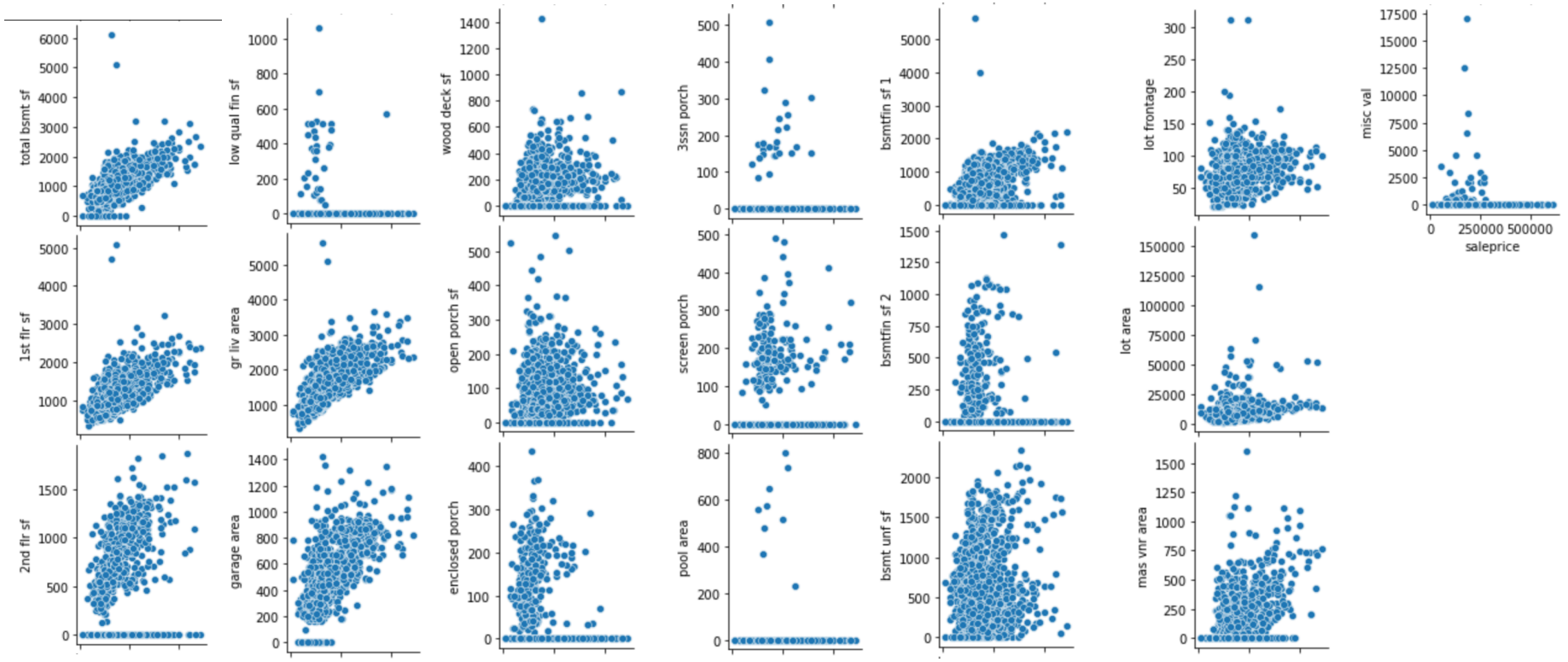
	testing
pool qc	875
misc feature	838
alley	821
fence	707
fireplace qu	422
lot frontage	160
garage cond	45
garage qual	45
garage yr blt	45
garage finish	45
garage type	44
bsmt exposure	25
bsmtfin type 1	25
bsmt qual	25
bsmtfin type 2	25
bsmt cond	25
mas vnr area	1
mas vnr type	1
electrical	1

```python
def fill_missing_value(data, list_col):

    """This function removes the missing values in 4 types
    1. fill 'None'
    2. drop column
    3. fill median
    4. drop row
    The inputs are data and list_col
    data = name of dataset
    list_col = list of column name which has missing values
    """

    fill_none_col = ['bsmt cond', 'bsmt qual', 'bsmtfin type 1', 'bsmtfin type 2', 'alley',
             'pool qc', 'misc feature', 'fence', 'garage cond', 'garage qual',
             'garage type', 'garage finish', 'fireplace qu', 'bsmt exposure']
    drop_col = ['mas vnr type', 'garage yr blt']
    fill_median_col = ['lot frontage', 'mas vnr area']
    drop_row = ['bsmt half bath', 'garage cars']
    fill_none_col_test = ['bsmt cond', 'bsmt qual', 'bsmtfin type 1', 'bsmtfin type 2', 'alley',
             'pool qc', 'misc feature', 'fence', 'garage cond', 'garage qual',
             'garage type', 'garage finish', 'fireplace qu', 'bsmt exposure', 'electrical']

    for col in list_col:
        if col in fill_none_col:
            data[col].fillna('None', inplace = True)
        elif col in drop_col:
            data.drop(columns = col, inplace = True)
        elif col in fill_median_col:
            data[col] = data[col].fillna(data[col].median())
        elif col in drop_row:
            data.dropna(subset =[col], inplace = True)
        elif col in fill_none_col_test:
            data[col].fillna('None', inplace = True)
```

Missing value removal

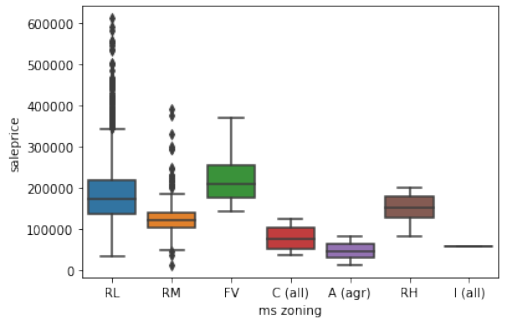
1. Filling 'None'
2. Column dropping
3. Filling median
4. Row dropping

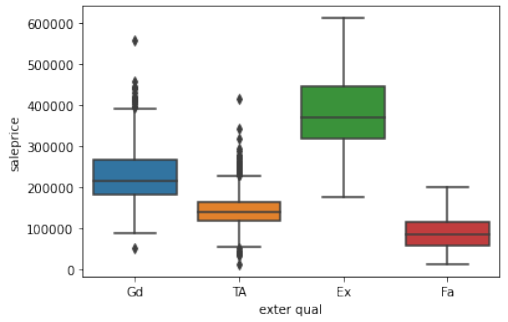
Identify outlier

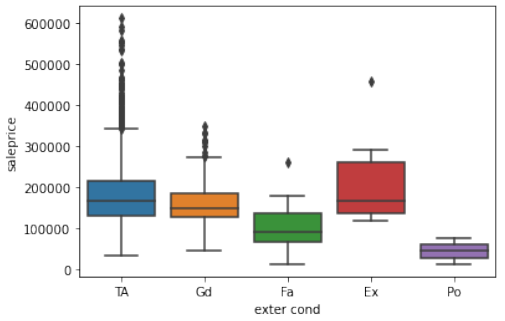


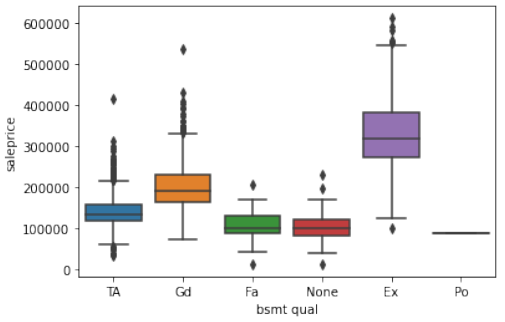

```
df_train = df_train[(df_train['lot frontage']< 150) & (df_train['lot area']< 75000)
                    & (df_train['mas vnr area']< 1250) & (df_train['bsmtfin sf 1']< 3000)
                    & (df_train['bsmtfin sf 2']< 1250) & (df_train['total bsmt sf']< 4000)
                    & (df_train['1st flr sf']< 4000) & (df_train['low qual fin sf'] < 600)
                    & (df_train['gr liv area']< 4000) & (df_train['garage area']< 1200)
                    & (df_train['wood deck sf']< 1000) & (df_train['open porch sf']< 400)
                    & (df_train['enclosed porch']< 400) & (df_train['3ssn porch']< 300)
                    & (df_train['screen porch']< 400) & (df_train['misc val']< 10000)]
```

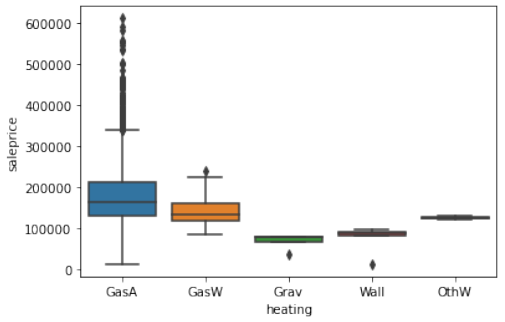
Categorical data

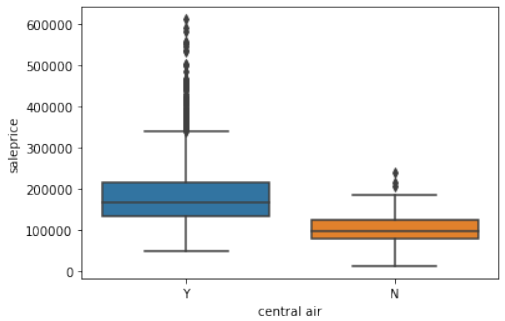


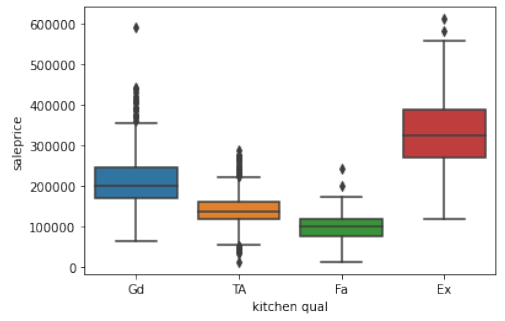


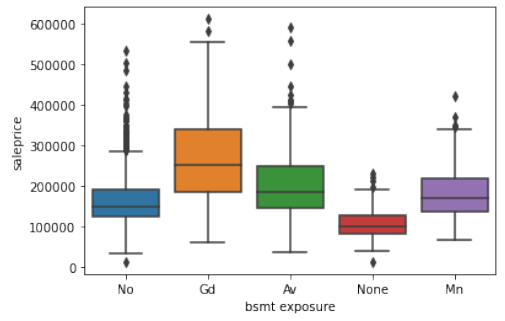


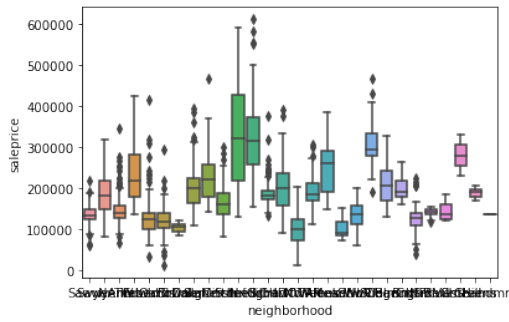


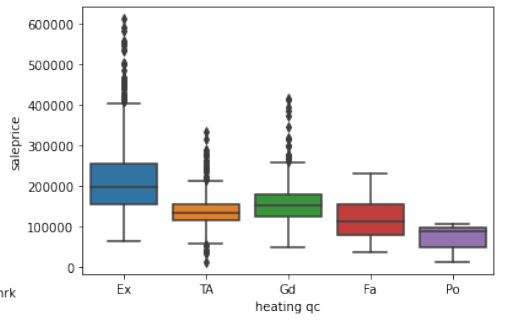


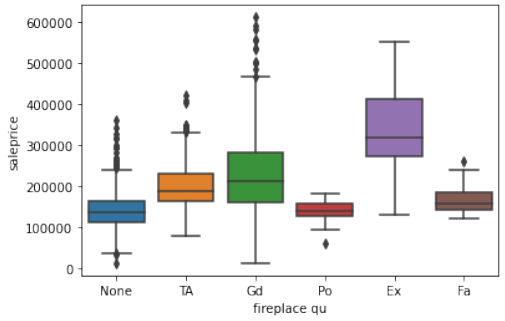


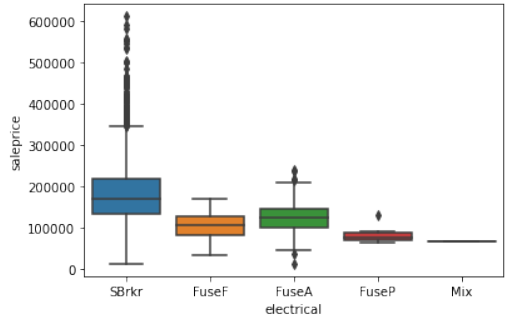













```
#The column that I choose
cat_col_selected = ['ms zoning', 'exter qual', 'exter cond', 'bsmt qual',
                    'heating', 'central air', 'kitchen qual', 'bsmt exposure',
                    'neighborhood', 'heating qc', 'fireplace qu', 'electrical']
```

Model 1 - correlation > 0.5

Correlation > 0.5

	column	correlation
0	saleprice	1.000000
1	overall qual	0.805039
2	gr liv area	0.715225
3	total bsmt sf	0.669156
4	garage area	0.656710
5	garage cars	0.651665
6	1st flr sf	0.649916
7	year built	0.587512
8	year remod/add	0.554741
9	full bath	0.537201
10	mas vnr area	0.510809
11	totrms abvgrd	0.504544

R-square and RMSE

```
model(X_train_sc, y_train, X_test_sc, y_test)
```

```
Linear Regression
R2 on training data:  0.8449026364810206
Linear R2 on testing data:  0.7609010493472357
Linear rmse:  31434.14190338744

Ridge Regression
R2 on training data:  0.844881852607136
Ridge R2 on testing data:  0.7615897956628761
Ridge rmse:  31429.924968025713

Lasso Regression
R2 on training data:  0.84476530260039
Lasso R2 on testing data:  0.7626750148595394
Lasso rmse:  31416.646775496054
```

Ridge

alpha = 9.8

R2 on training data = 0.84

R2 on testing data = 0.76

Lasso

alpha = 194

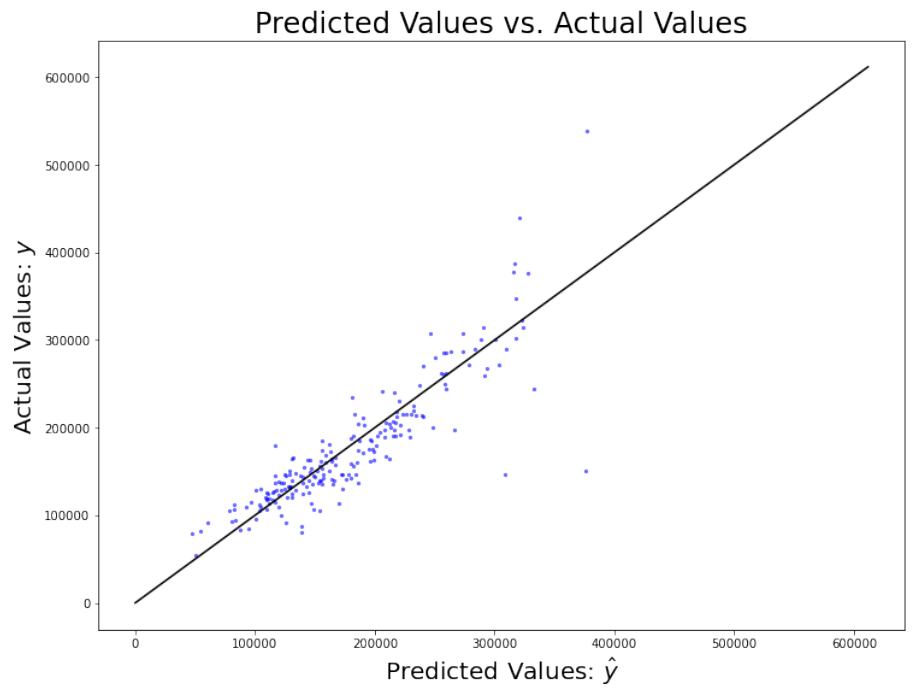
R2 on training data = 0.84

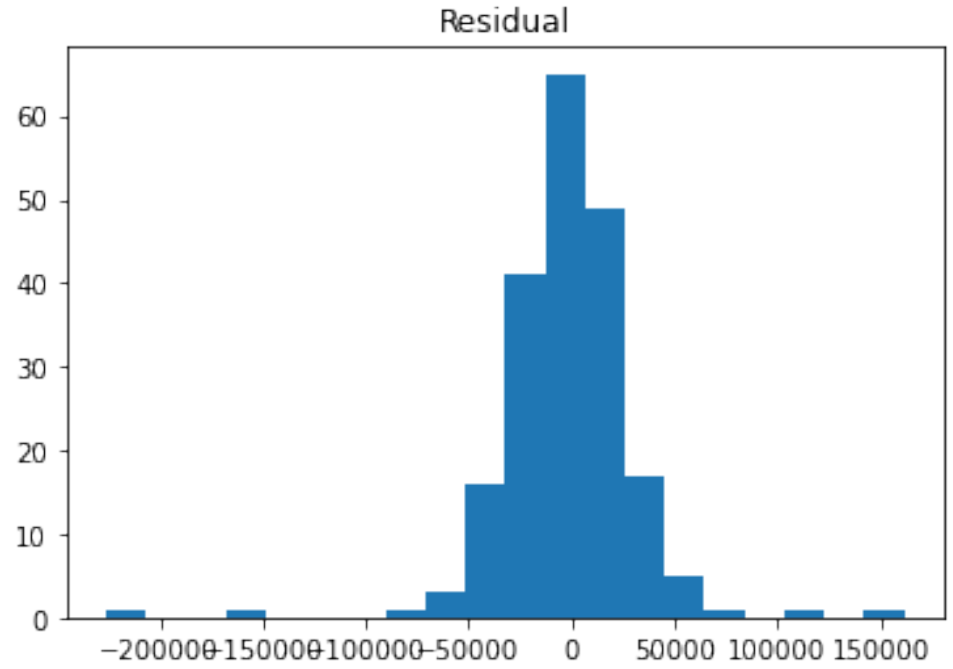
R2 on testing data = 0.76

- All 3 models have the same R2 and RMSE.
- The accuracy is about 84% on training data and 76% on testing data.
- All 3 models are overfit model.

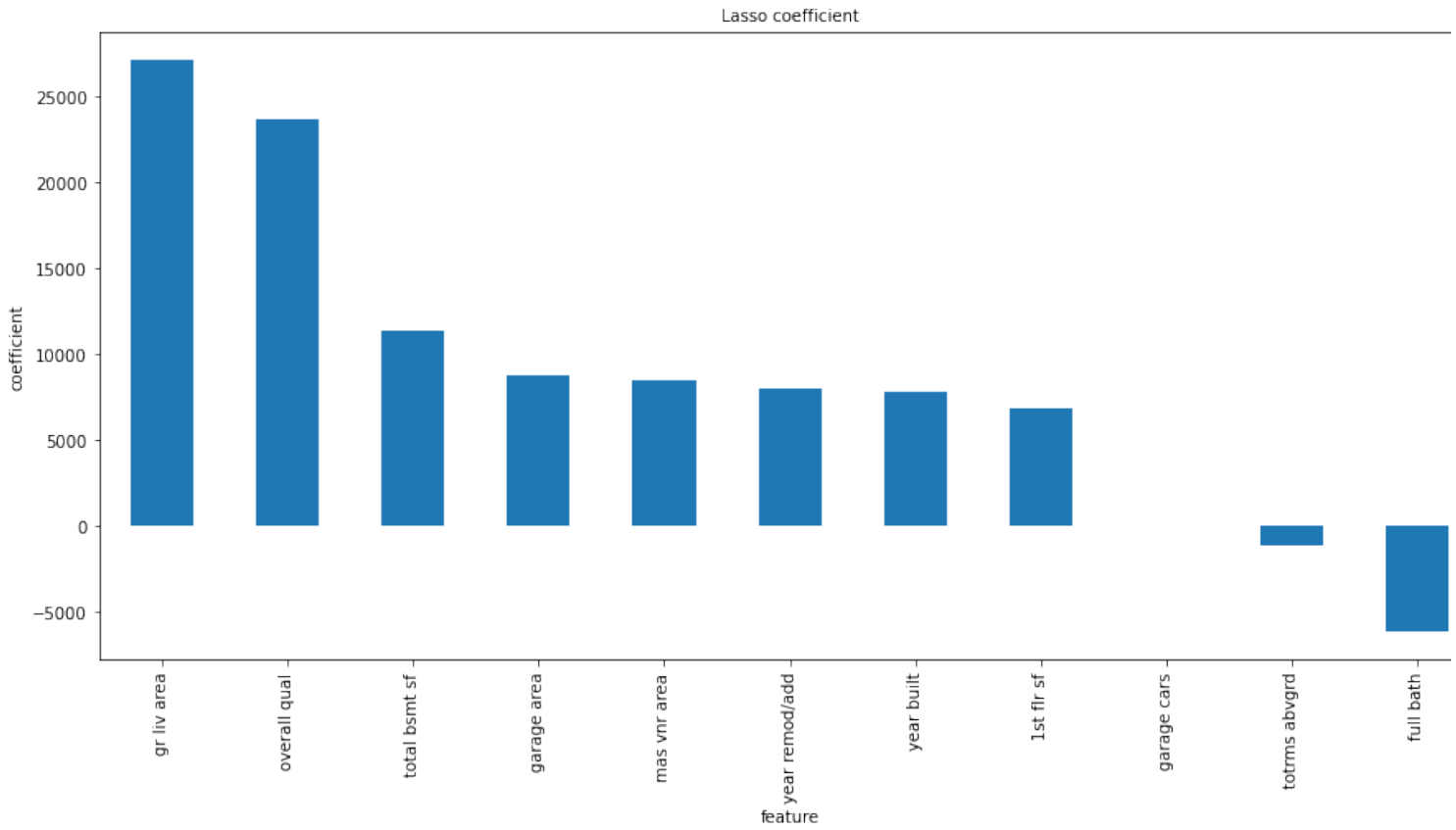
Model 1 Evaluation

From Lasso model, the scatter plot shows the prediction has some error, and the residual graph is the normal distribution.





Model 1 – Lasso coefficient



- Lasso model can halp me filter features. The bar chart shows that 'garage cars', 'totrms abvgrd' and 'full bath' should be filtered out.
- The result of prediction is exported to .csv file named submission_1.
- The score on Kaggle is 33996.03805.

Model 2 – Lasso coefficient > 0 and one-hot encoder

```
features_2 = ['overall qual', 'gr liv area','garage area','total bsmt sf','1st flr sf',
              'year built','year remod/add'] + list(df_training_encoder.columns)
```

features

```
encoder_training = OneHotEncoder(sparse = False, handle_unknown = 'ignore')
encoder_training.fit(df_train[cat_col_selected])
df_training_encoder = pd.DataFrame(encoder_training.transform(df_train[cat_col_selected]),
                                   columns = encoder_training.get_feature_names(df_train[cat_col_selected].columns))
df_testing_encoder = pd.DataFrame(encoder_training.transform(df_test[cat_col_selected]),
                                   columns = encoder_training.get_feature_names(df_test[cat_col_selected].columns))
```

OneHotEncoder()

```
model(X_train_sc_2,y_train_2, X_test_sc_2, y_test_2)
```

```
Linear Regression
R2 on training data:  0.9068526852543879
Linear R2 on testing data:  0.8478004894877978
Linear rmse:  7.841575632778682e+16

Ridge Regression
R2 on training data:  0.906852206555349
Ridge R2 on testing data:  0.8483773909414956
Ridge rmse:  25270.510903381568

Lasso Regression
R2 on training data:  0.9065298151630463
Lasso R2 on testing data:  0.8510627565849391
Lasso rmse:  25160.54018886906
```

Ridge

alpha = 10.0

R2 on training data = 0.90

R2 on testing data = 0.84

Lasso

alpha = 157.69

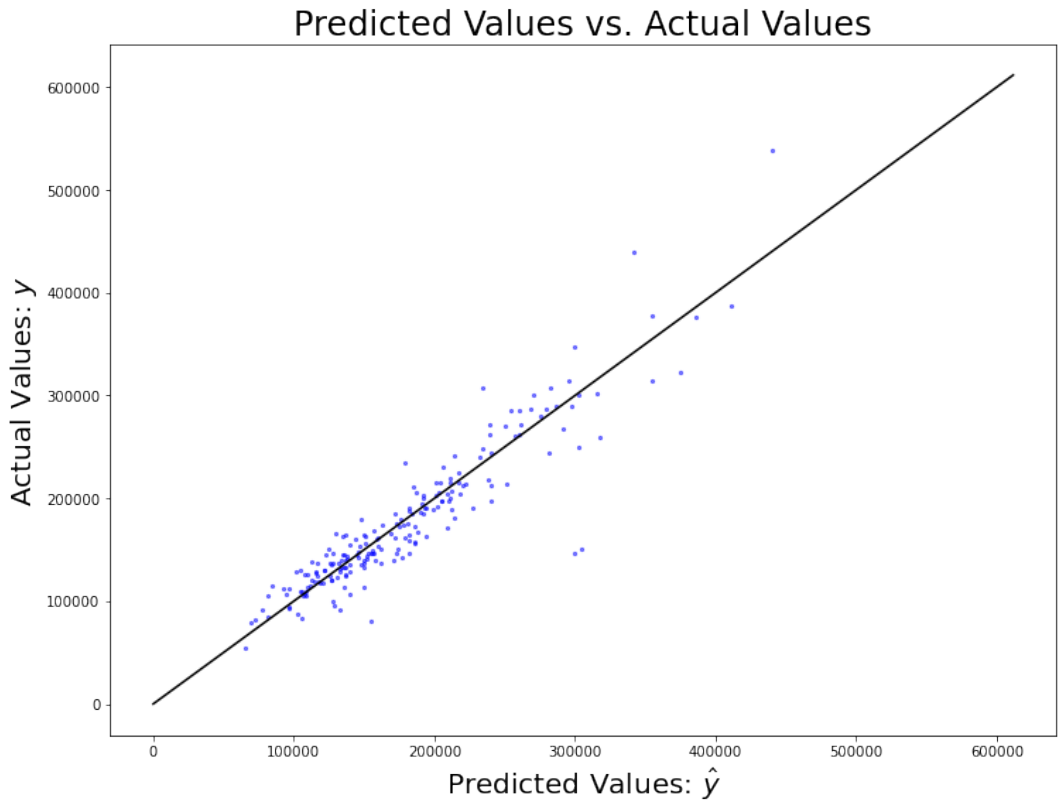
R2 on training data = 0.90

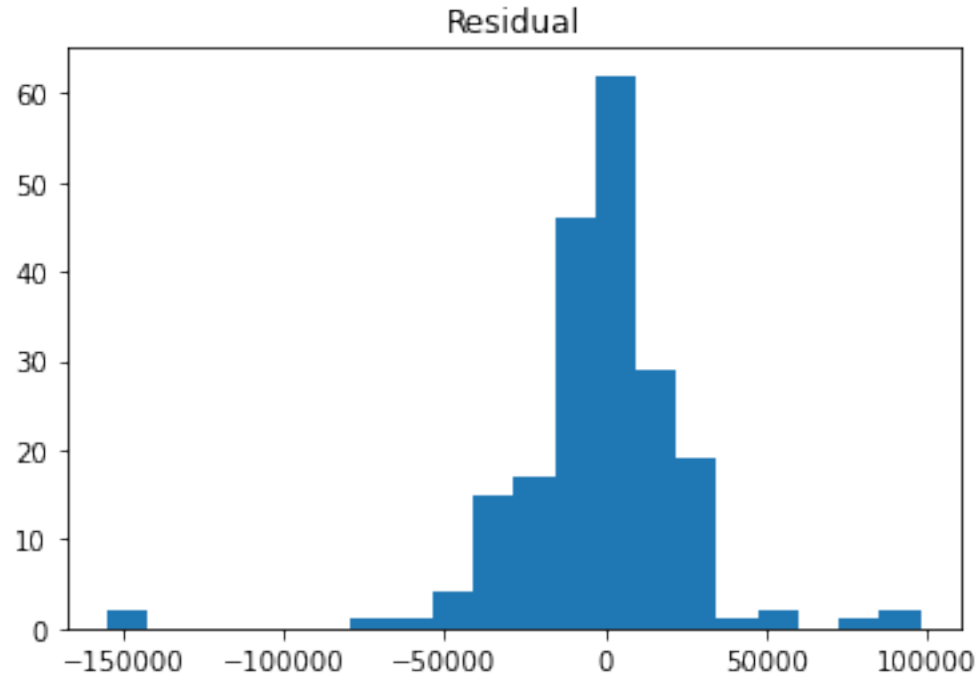
R2 on testing data = 0.85

- Alpha' for Lasso decreases from model 1
- Accuracy = 90% on training data, and 85% on testing data.
- The model still has the overfit condition.

Model 2 Evaluation

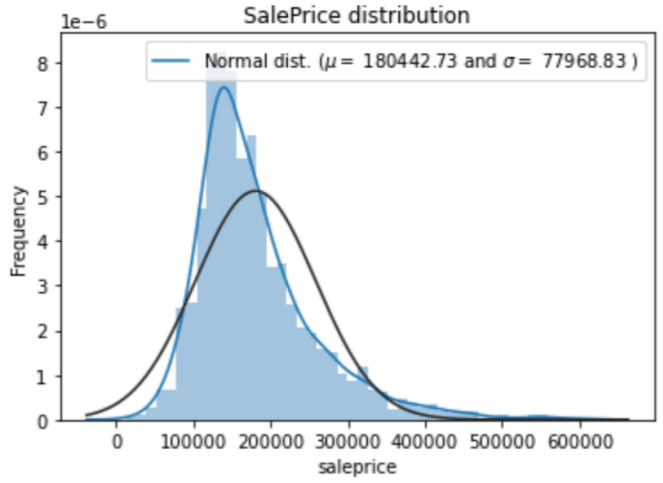
From Lasso model, the scatter plot shows the prediction is linear, but the distribution of residual is quite different from model 1.

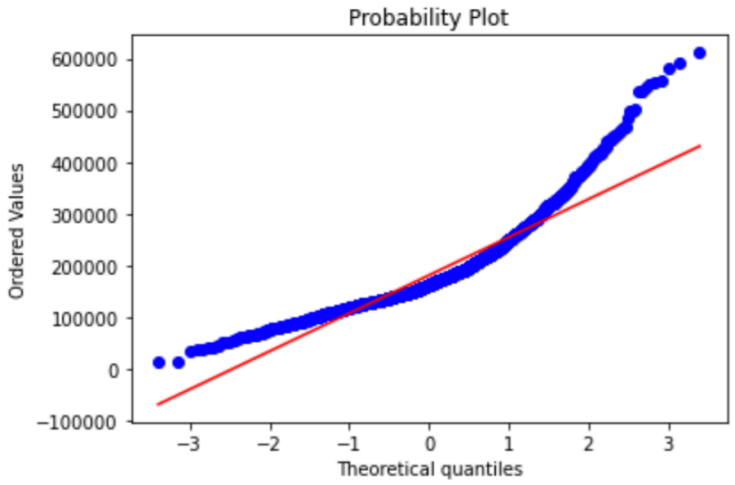




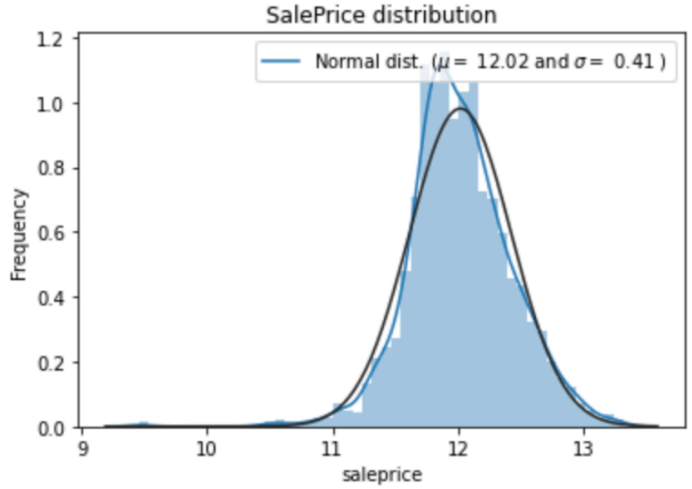
Model 3 – Log transformation

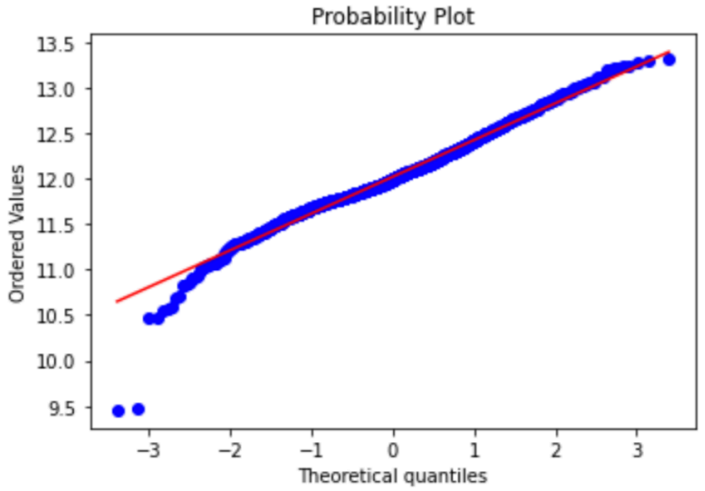
Original sale price distribution





Log transformation





- As the assumption of linear regression, log transformation is used to adjust the sale price value to be more linear and normal distribution.

Model 3 – Log transformation

```
features_3 = features_2
```

features

```
y_3 = np.log(df_train_model['saleprice'])
X_3 = df_train_model[features_3]
X_df_test_3 = df_test_model[features_3]
```

Ridge

alpha = 10

R2 on training data = 0.84

R2 on testing data = 0.76

Lasso

alpha = 0.002

R2 on training data = 0.84

R2 on testing data = 0.76

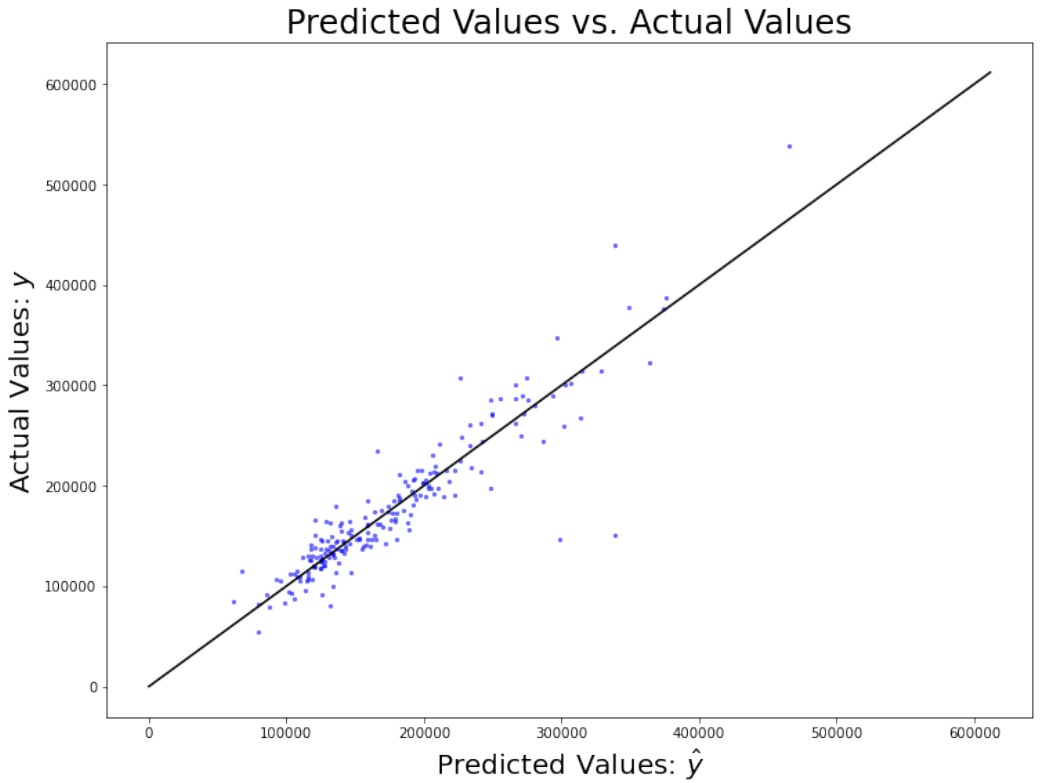
```
model(X_train_sc_3,y_train_3, X_test_sc_3, y_test_3)
```

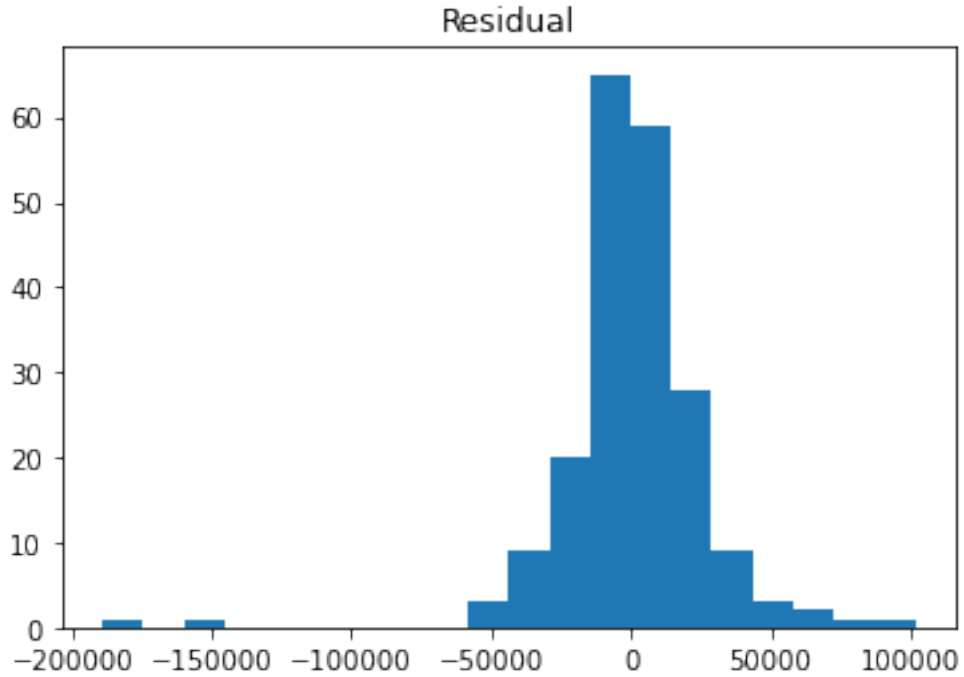
```
Linear Regression
R2 on training data:  0.9076880295169286
Linear R2 on testing data:  0.8343412452403224
Linear rmse:  364684626168.78876

Ridge Regression
R2 on training data:  0.907711430199991
Ridge R2 on testing data:  0.8340577306946904
Ridge rmse:  0.1440645061409937

Lasso Regression
R2 on training data:  0.9064128728186647
Lasso R2 on testing data:  0.8398726690740876
Lasso rmse:  0.1426081447109504
```

- Alpha's Lasso decreases from model 2.
- Accuracy = 84% on training data, and 76% on testing data.
- The model still has the overfit condition.
- The score on Kaggle is 23706.44408.
- Even if the accuracy decreases, the score on Kaggle is better than the previous models.

Model 3 Evaluation

From Lasso model, the scatter plot shows the prediction is linear, and the distribution of residual is nearly 0.





log(saleprice) = 12.02 + 1.125026(gr liv area) + 1.098346(overall qual) + 1.055842(total bsmt sf) + 1.035671(year built) + 1.032073(garage area) + ...

Answer to problem statement

- The top 5 features that make more value of house are following the table. These features can increase the price follow the equation (1).

  Equation(1):
  log(saleprice) = 12.02 + 1.125026(gr liv area) + 1.098346(overall qual) + 1.055842(total bsmt sf) + 1.035671(year built) + 1.032073(garage area) + ...

Feature	Meaning	Coefficient
gr liv area	*Above grade (ground) living area square feet*	1.125026
overall qual	*Rates the overall material and finish of the house*	1.098346
total bsmt sf	*Total square feet of basement area*	1.055842
year built	*Original construction date*	1.035671
garage area	*Size of garage in square feet*	1.032073

Answer to problem statement

- The 5 things that should be avoid if you do not want to drop the house price are in this table.

Feature	Meaning	Coefficient
exter cond_Po	*Evaluates the present condition of the material on the exterior (Poor)*	0.978867
ms zoning_C (all)	*Identifies the general zoning classification of the sale (Commercial)*	0.976815
central air_N	*Central air conditioning (No)*	0.976480
fireplace qu_None	*Fireplace quality (None)*	0.973895
ms zoning_RM	*Size of garage in square feet (Residential Medium Density)*	0.970536

- The houseowner should improve the exterior condition and maintenance the fireplace.
- The central air should be set up to avoid the dropping of price.
- When the estate company want to make a new project, the company should consider about the zone of location.

Answer to problem statement

- This model cannot predict the house price in other city because the training data was collected from Ames Iowa only.
- Some features need to be adjust and remove.
- For example, neighborhood in the training data is the specific areas are found in Ames Iowa, so the neighborhood might be change to the environment around the house i.e. school, shooping mall and bus stop.
- The location of house in term of longitude and latitude also gives the advantage because everywhere has this data and this data will help to specific the location of house.
- This has more accuracy than zoning of house.