HOUSE PRICE PREDICTION

HANQING ZHANG

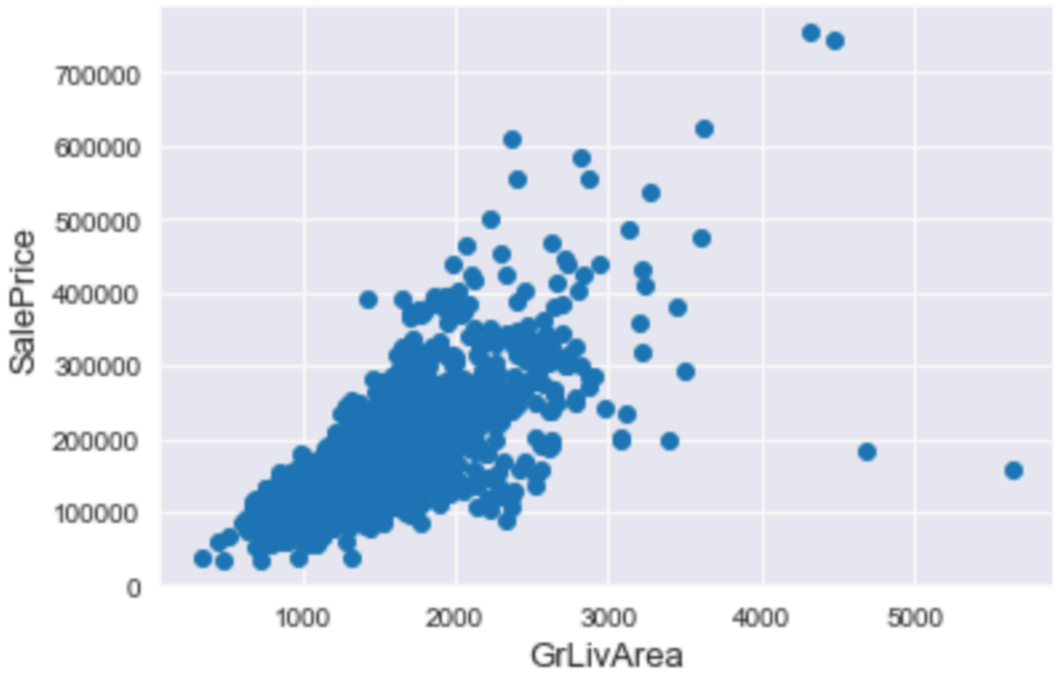
INTRODUCTION

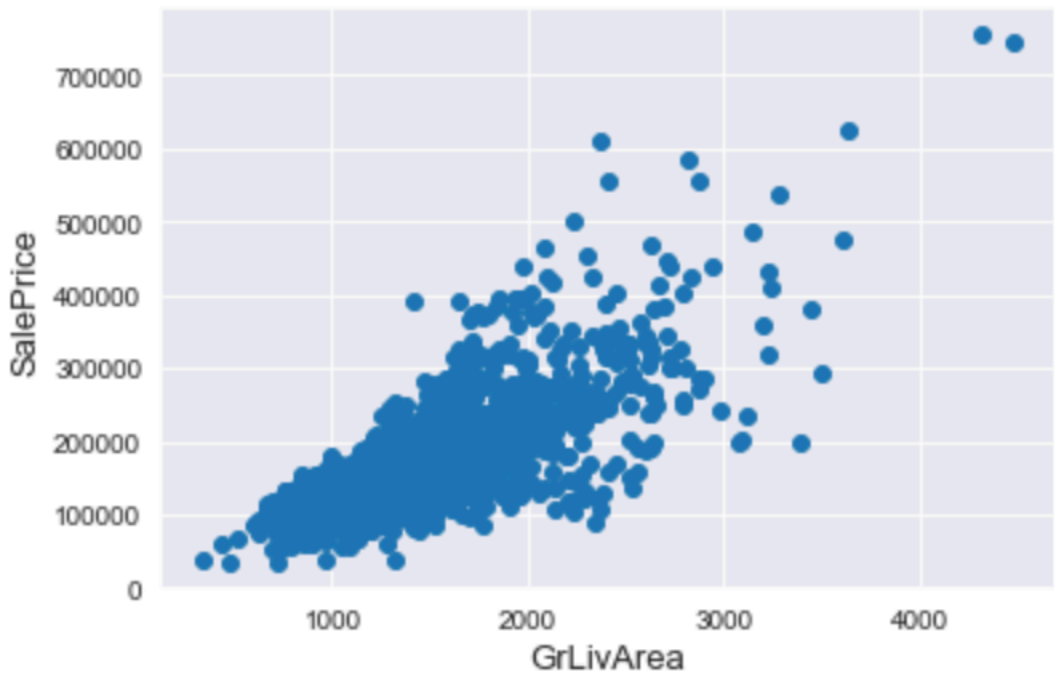
Purchasing a home is one of the most decisions people make. It is pivotal that a prospective home buyer makes this purchase at the correct price. However, when facing a decision of such financial magnitude, people may consider that they are paying more for the house than it’s worth.

DATA

- Kaggle Dataset
- https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data
- 1460 observations, 79 features
- 51 categorical, 28 continuous

DATA PREPROCESSING– OUTLIERS

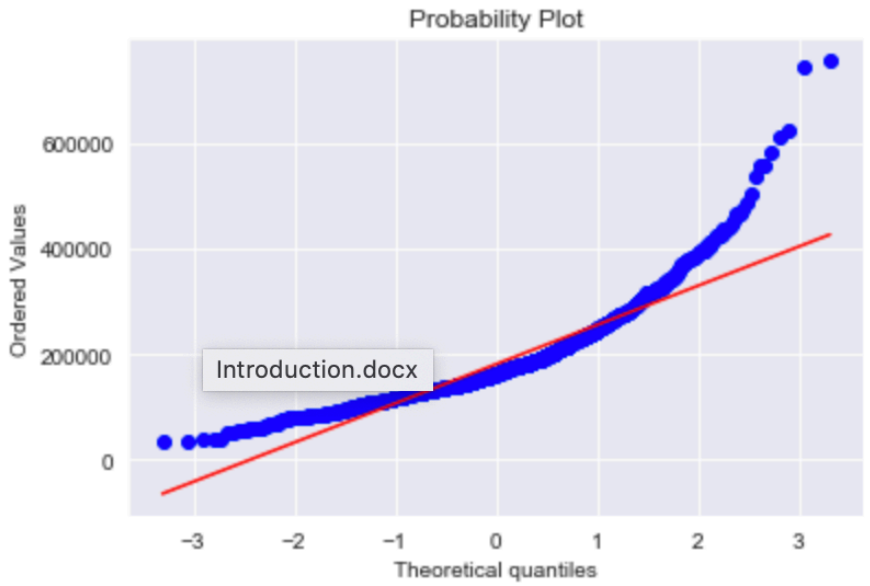




DATA PREPROCESSING

- The target variable Saleprice looks right skewed

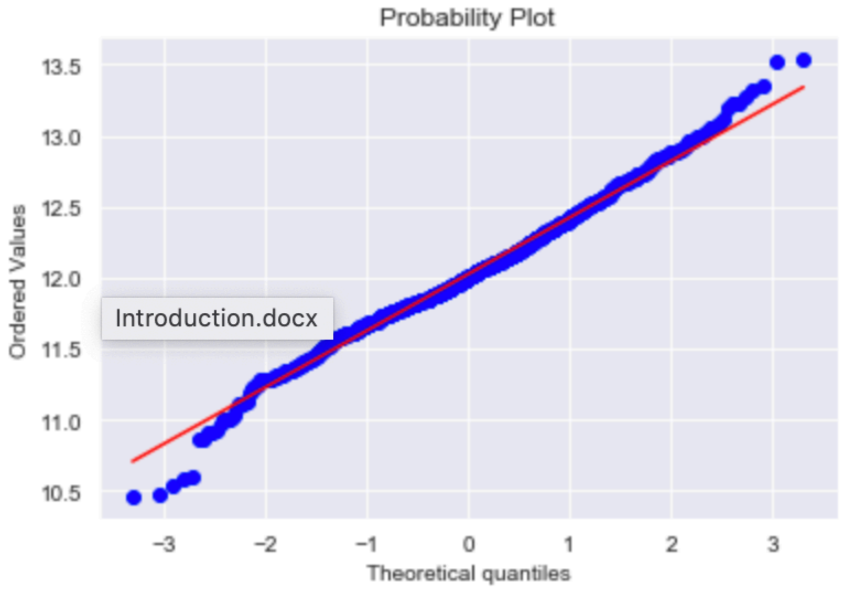
```
mu = 180932.92 and sigma = 79467.79
```



DATA PREPROCESSING

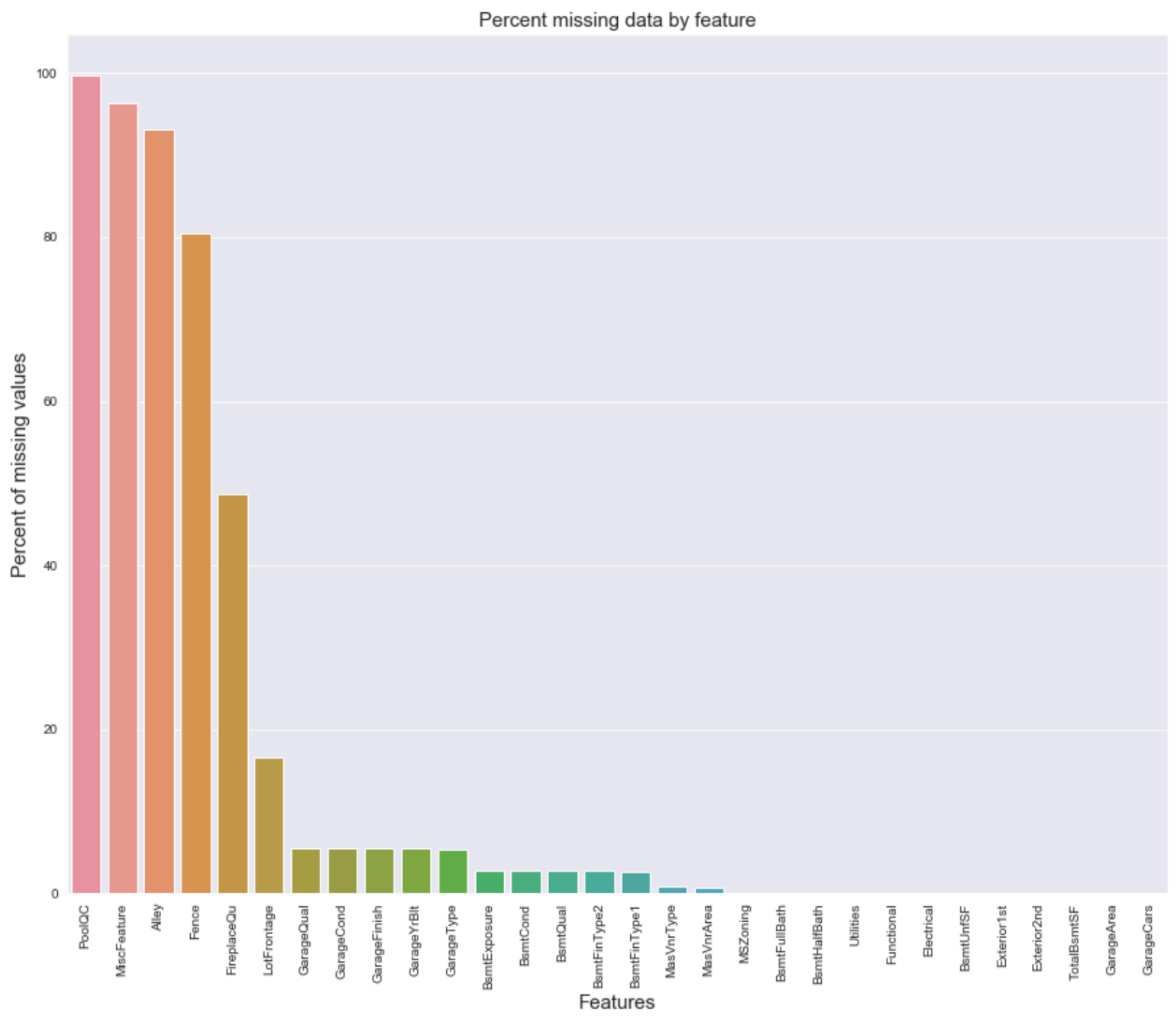
- Log transformation is applied to Saleprice

mu = 12.02 and sigma = 0.40



FEATURE ENGINEER

- The plot shows the missing rate for each variable. According to different missing rates combined with data description, we use different ways to impute the missing values.



FEATURE ENGINEER

- Investigating skewness for each feature, and use a box-cox method to transform the data

	Skew
MiscVal	21.939672
PoolArea	17.688664
LotArea	13.109495
LowQualFinSF	12.084539
3SsnPorch	11.372080
LandSlope	4.973254
KitchenAbvGr	4.300550
BsmtFinSF2	4.144503
EnclosedPorch	4.002344
ScreenPorch	3.945101

METHOD

- We basically use four models to initialize the data, here are the tuning parameters.

```
lasso = make_pipeline(RobustScaler(),
                      Lasso(alpha =0.0011, random_state=1))
```

```
ENet = make_pipeline(RobustScaler(),
                     ElasticNet(alpha=0.0018,
                                l1_ratio=1.1,
                                random_state=9))
```

```
KRR = KernelRidge(alpha=0.2, kernel='polynomial', degree=2, coef0=5.0)
```

```
GBoost = GradientBoostingRegressor(n_estimators=3000, learning_rate=0.1,
                                   max_depth=4, max_features='sqrt',
                                   min_samples_leaf=11, min_samples_split=10,
                                   loss='huber', random_state =5)
```

MODEL AVERAGING

- Model averaging is an approach to ensemble learning where each ensemble member contributes an equal amount to the final prediction. In the case of regression, the ensemble prediction is calculated as the average of the member predictions. We define a class first to allow averaging models. We first define clones of the original four models we preciously have to fit the data in; Then train the cloned base models; Finally make predictions for cloned models and average the predictions got from each model.

STACKING

- What we do is to define the class, and first fit the data on clones of the original models; Train cloned base models then create out-of-fold predictions that are needed to train the cloned meta-model; Then train the cloned meta-model using the out-of-fold predictions as a new feature; Finally make the predictions of all base models on the test data and use the average predictions as meta-features for the final prediction which is done by the meta-model.

Algorithm	score
Lasso	0.1139
Kernel Ridge	0.1190
ElasticNet	0.1187
Gradient Boosting	0.1195

Algorithm	score
Averaged base models sore	0.0773

Algorithm	score
Stacking averaged models sore	0.0722

RESULTS

THANK YOU