Zillow's Price Data Exploration

Chan Yu Yankai Liu Yifei Bi

OverView

1.Program Background
2.Trainset Exploration
3.Property Exploration
4.Geograhic&correlation

Reporter : Yankai Liu

1

Program background

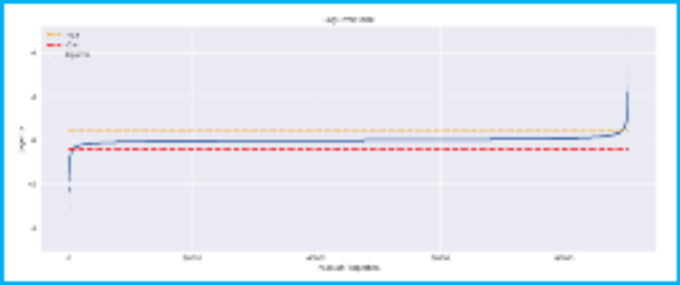
Question1:What is zillow and Zestimates?
Zillow: Real estate price evaluation company
Zestimates: Using algorithms and data to estimate the price.

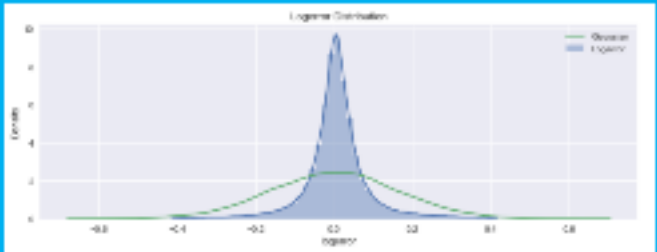
Question2:Why is Zestimates so important?

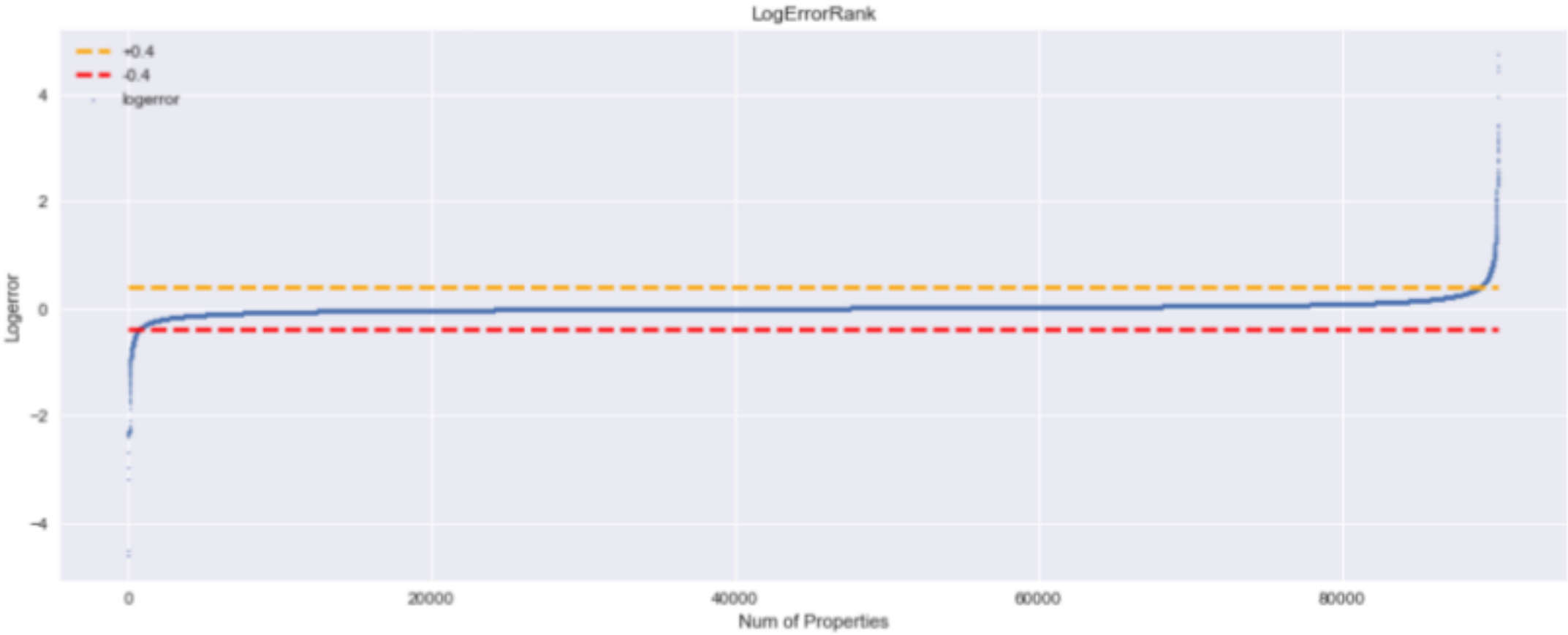
Zestimates can evaluate the price of real estate. But Zestimates is based on statistical and machine learning models. So the efficiency and accuracy of the model are very important.

2

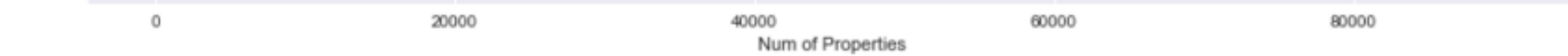
Trainset exploration

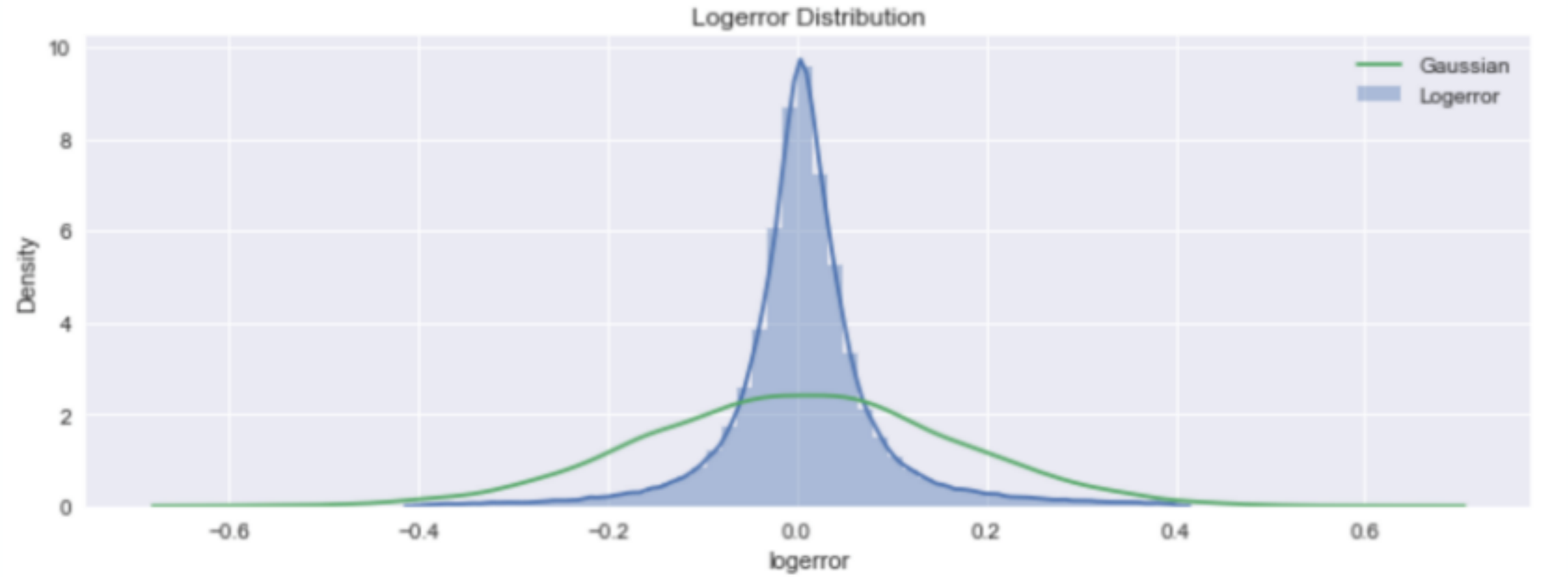


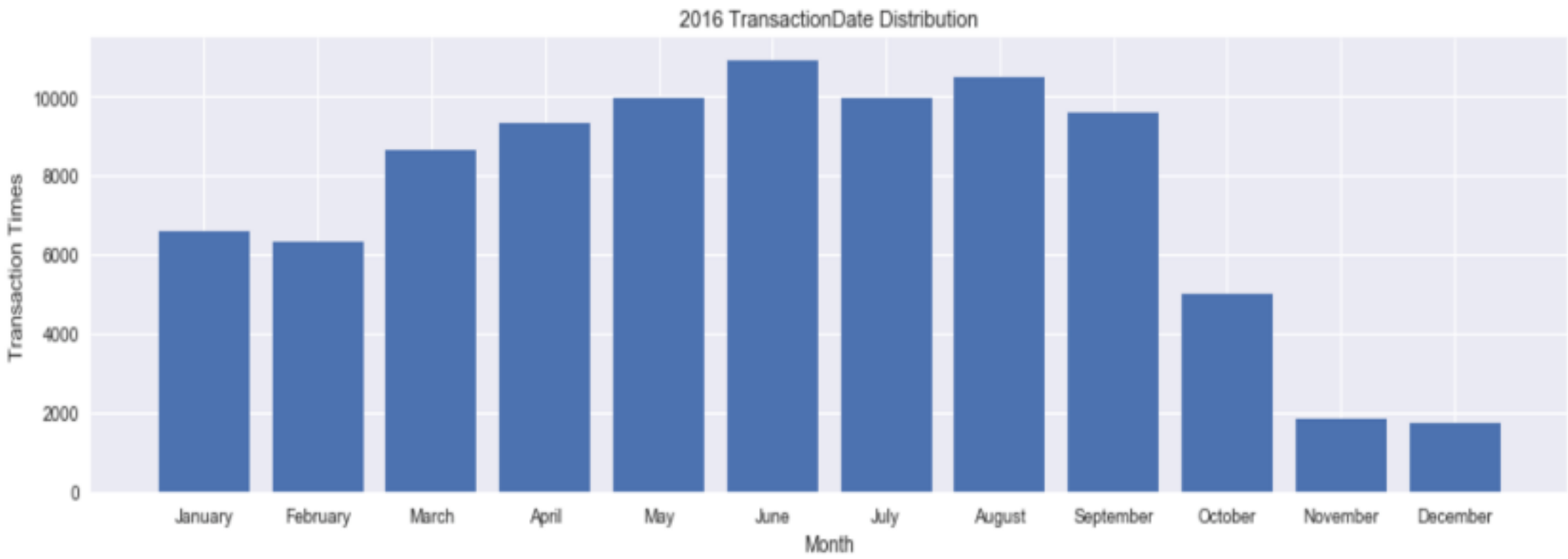



LogErrorRank
+0.4
-0.4
logerror
4
2
0
−2
−4
Logerror
0
20000
40000
60000
80000
Num of Properties


Logerror Distribution
10

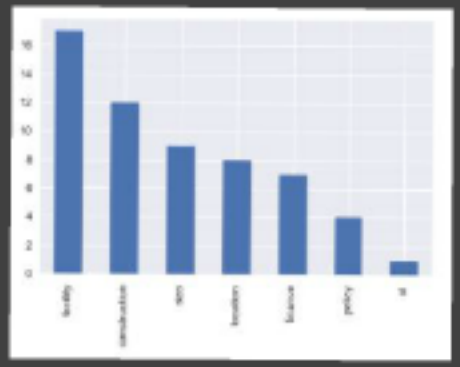

0
20000
40000
60000
80000
Num of Properties

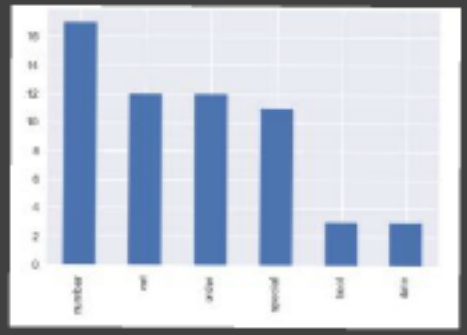

Logerror Distribution
Gaussian
Logerror
Density
10
8
6
4
2
0
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
logerror

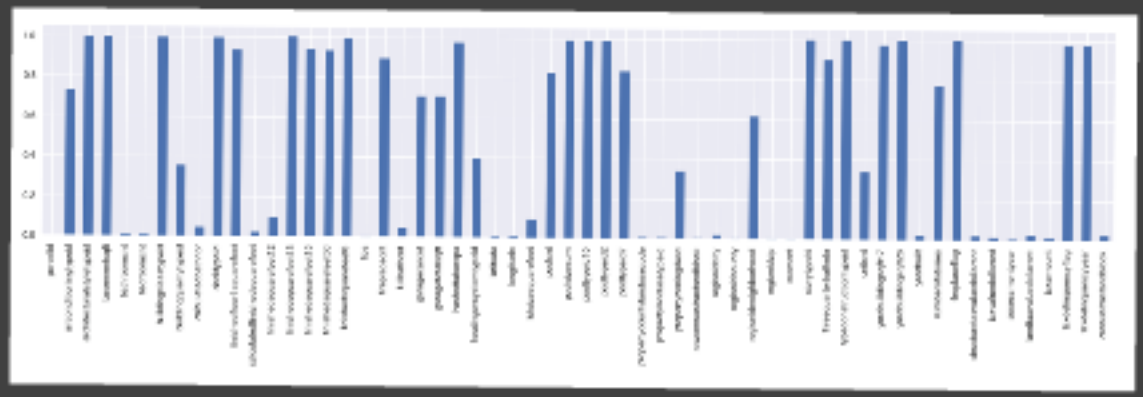

2016 TransactionDate Distribution
Transaction Times
0
2000
4000
6000
8000
10000
January
February
March
April
May
June
July
August
September
October
November
December
Month

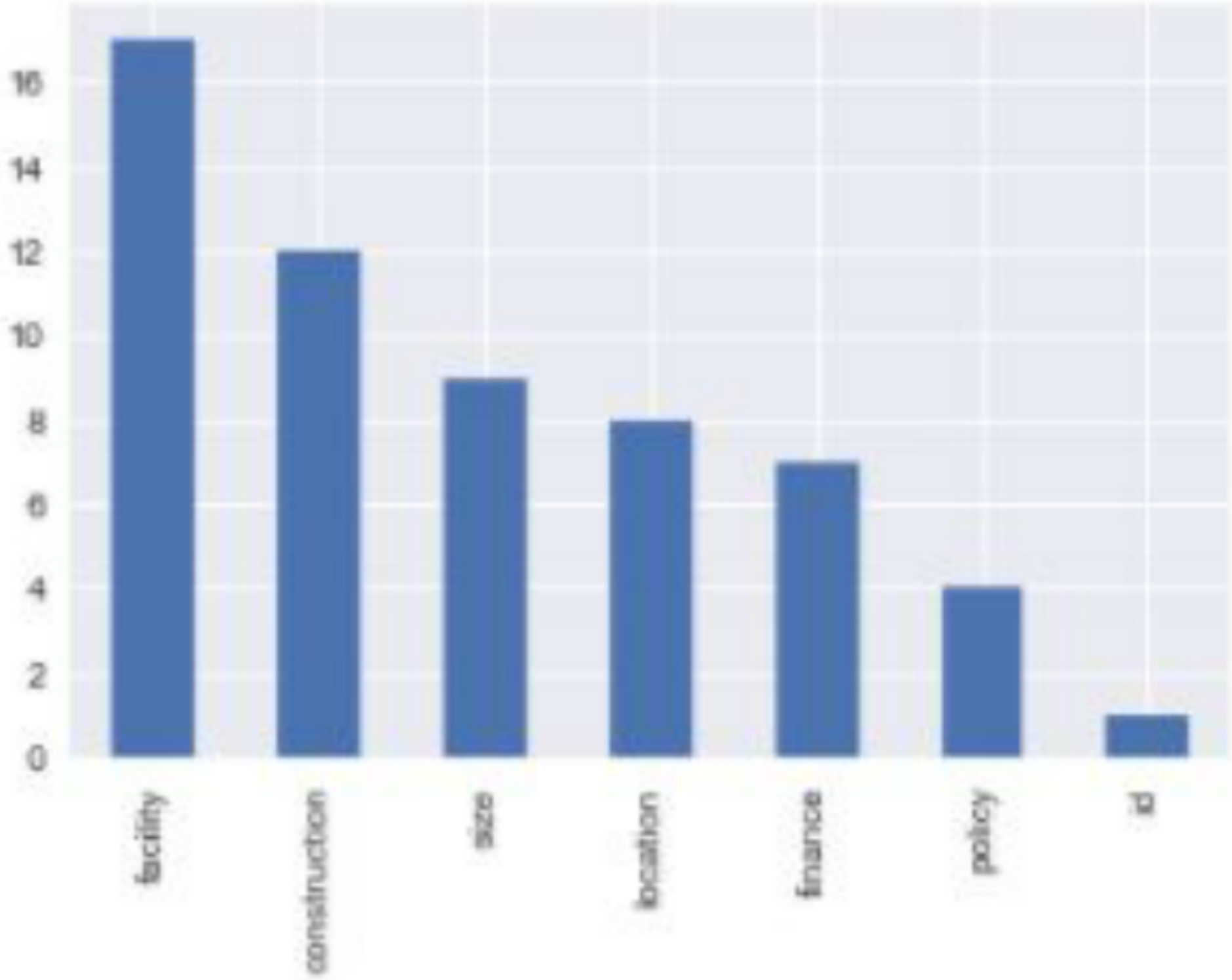
3

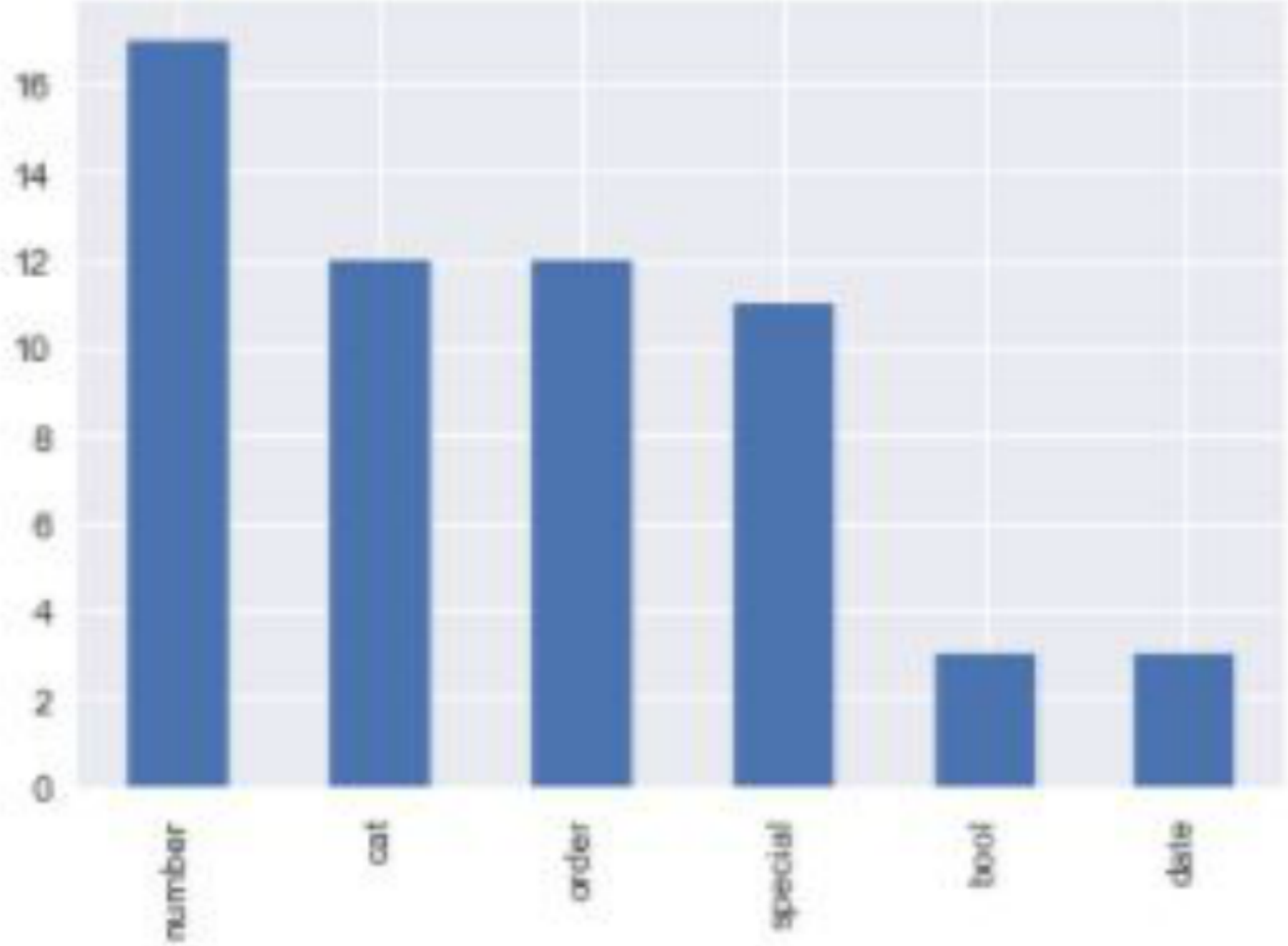
Properties exploration

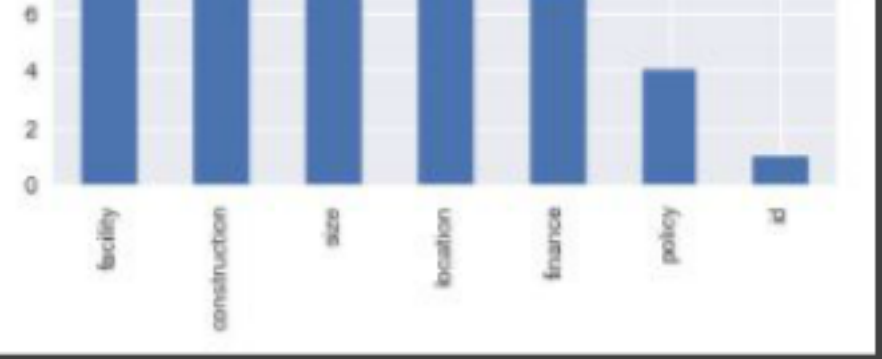


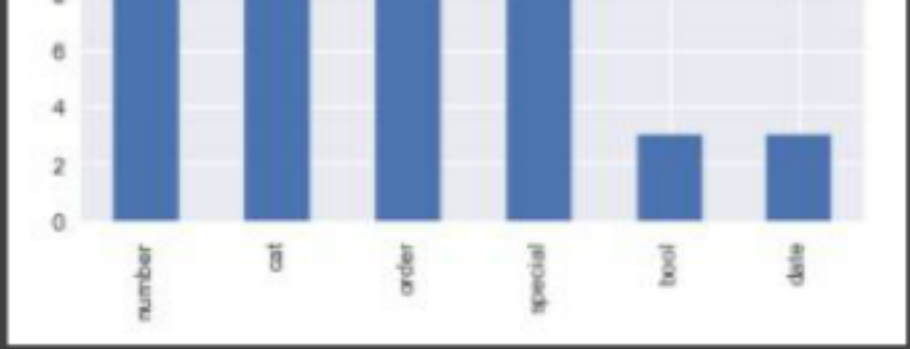


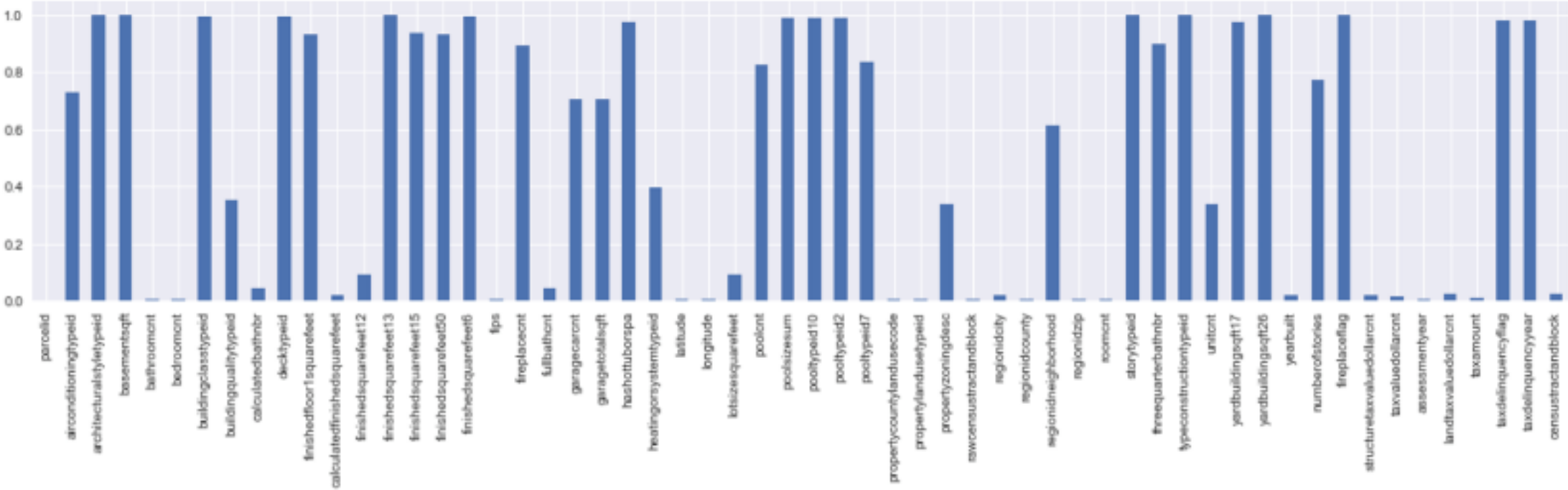



16
14
12
10
8
6
4
2
0
facility
construction
size
location
finance
policy
id

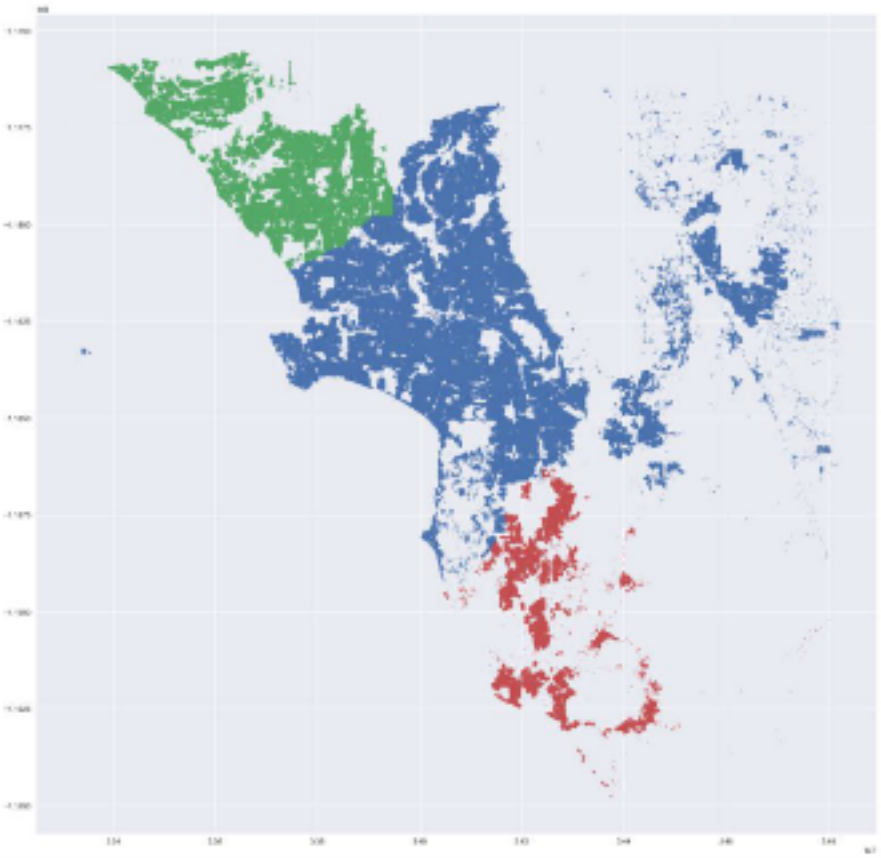

16
14
12
10
8
6
4
2
0
number
cat
order
special
bool
date

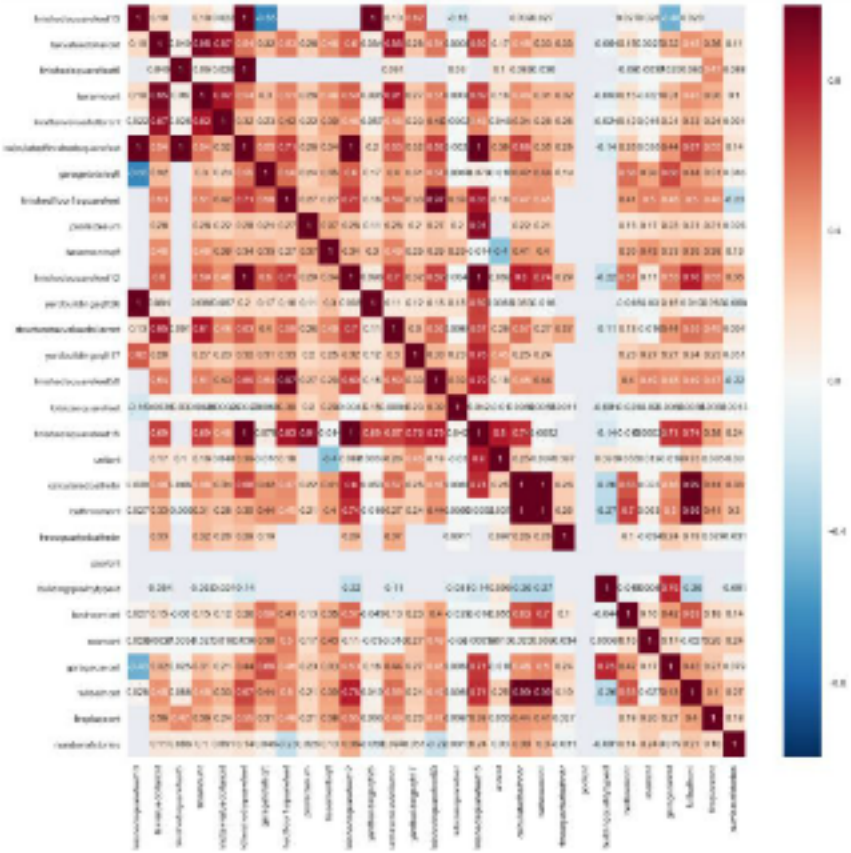

0
2
4
6
facility
construction
size
location
finance
policy
id


0
2
4
6
number
cat
order
special
bool
date


0.0
0.2
0.4
0.6
0.8
1.0
parcelid
airconditioningtypeid
architecturalstyletypeid
basementsqft
bathroomcnt
bedroomcnt
buildingclasstypeid
buildingqualitytypeid
calculatedbathnbr
decktypeid
finishedfloor1squarefeet
calculatedfinishedsquarefeet
finishedsquarefeet12
finishedsquarefeet13
finishedsquarefeet15
finishedsquarefeet50
finishedsquarefeet6
fips
fireplacecnt
fullbathcnt
garagecarcnt
garagetotalsqft
hashottuborspa
heatingorsystemtypeid
latitude
longitude
lotsizesquarefeet
poolcnt
poolsizesum
pooltypeid10
pooltypeid2
pooltypeid7
propertycountylandusecode
propertylandusetypeid
propertyzoningdesc
rawcensustractandblock
regionidcity
regionidcounty
regionidneighborhood
regionidzip
roomcnt
storytypeid
threequarterbathnbr
typeconstructiontypeid
unitcnt
yardbuildingsqft17
yardbuildingsqft26
yearbuilt
numberofstories
fireplaceflag
structuretaxvaluedollarcnt
taxvaluedollarcnt
assessmentyear
landtaxvaluedollarcnt
taxamount
taxdelinquencyflag
taxdelinquencyyear
censustractandblock

Geograhic&correlation

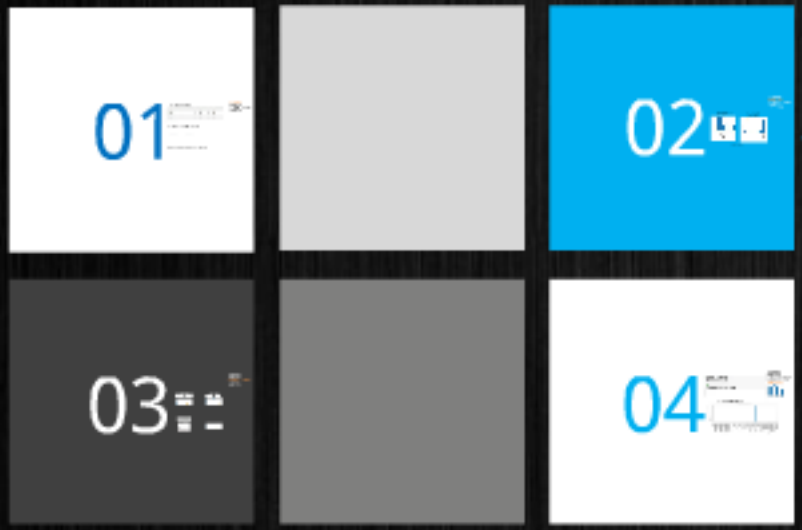




Zillow's Price Feature Engineering

Chan Yu Yankai Liu Yifei Bi

Procedures Guide



1.Preparation
2.Impute Data 1
3.Creating New Features
4.Impute Data 2

Reporter : Chan Yu

1

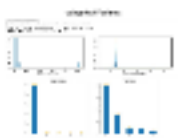
1.Preparation
2.Impute Data 1
3.Creating New Features
4.Impute Data 2

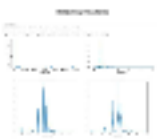
1.Convert Float64 to Float32

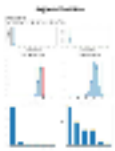
	parcelid	airconditioningtypeid	architecturalstyletypeid	basementsqft	bathroomcnt	bedroomcnt	buildingclasstypeid
0	10754147	1.0	7.0	535.0	0.0	0.0	4.0
1	10759547	1.0	7.0	535.0	0.0	0.0	4.0
2	10843547	1.0	7.0	535.0	0.0	0.0	5.0
3	10859147	1.0	7.0	535.0	0.0	0.0	3.0
4	10879947	1.0	7.0	535.0	0.0	0.0	4.0

5 rows x 54 columns

2.Group Features into Different Groups



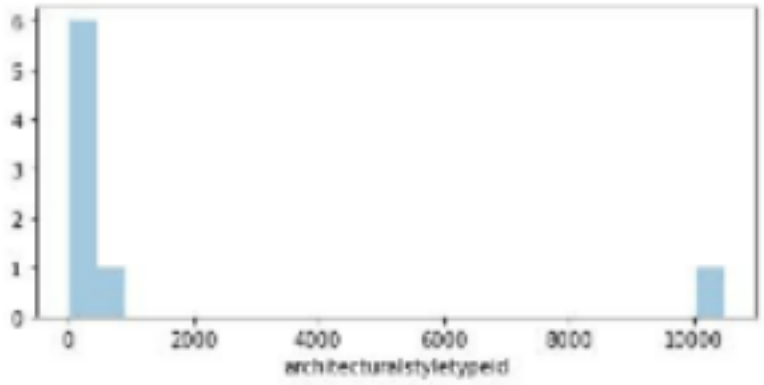


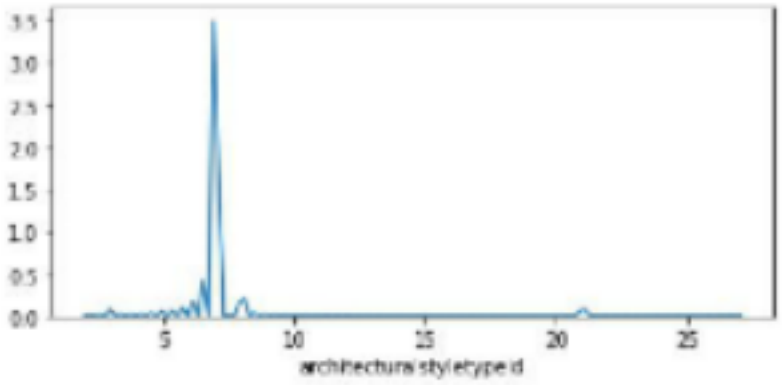


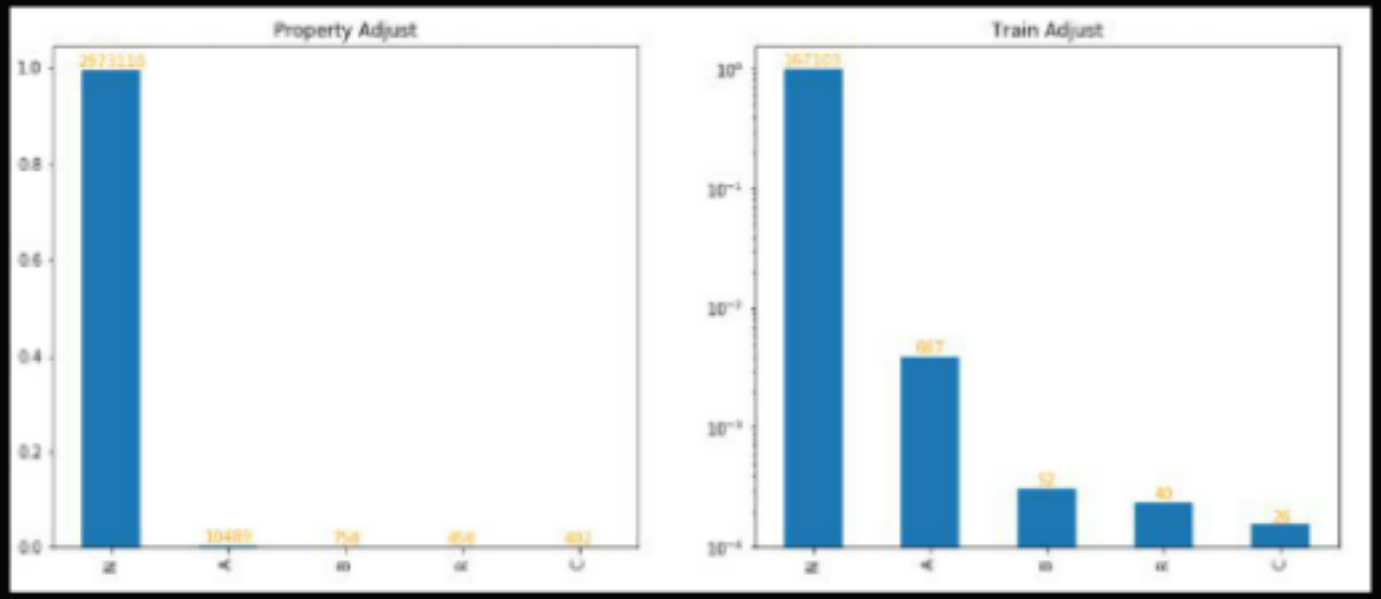
3.Prepare Functions For Corresponding Group

Categorical Features

architecturalstyletypeid
Full NaN : 99.59%: Full NumOfCat : 9 Full CatRatio : 0.07%
Train NaN : 99.53% Train NumOfCat : 8 Train CatRatio : 1.11%
[10489L, 758L, 402L, 300L, 116L, 38L, 2L, 2L]

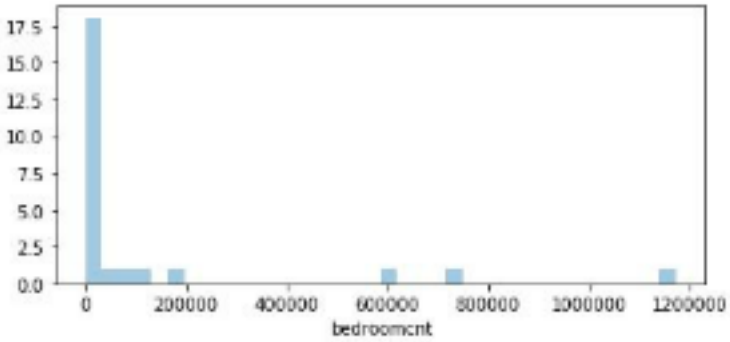


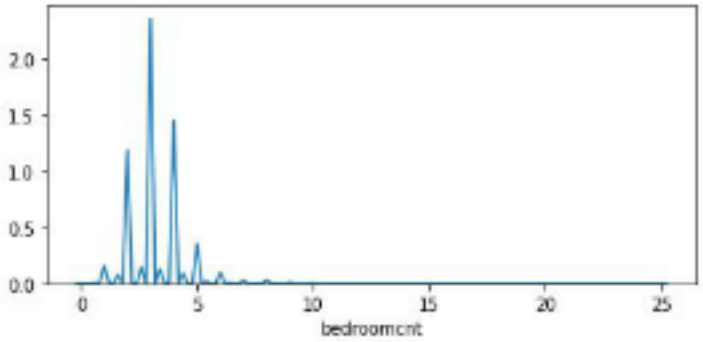


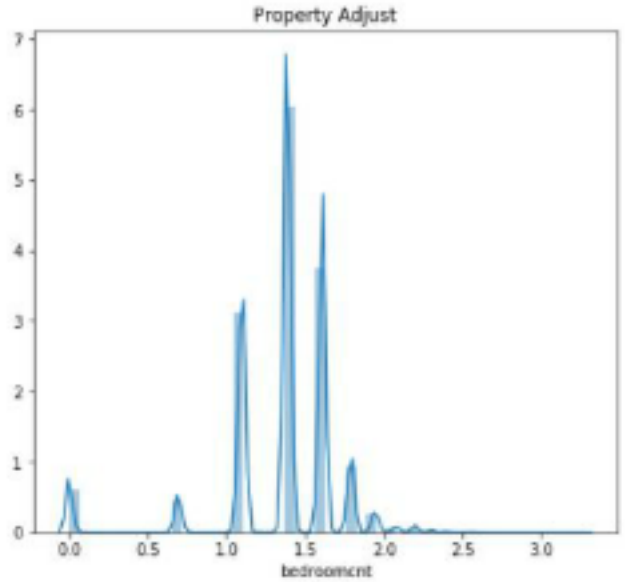


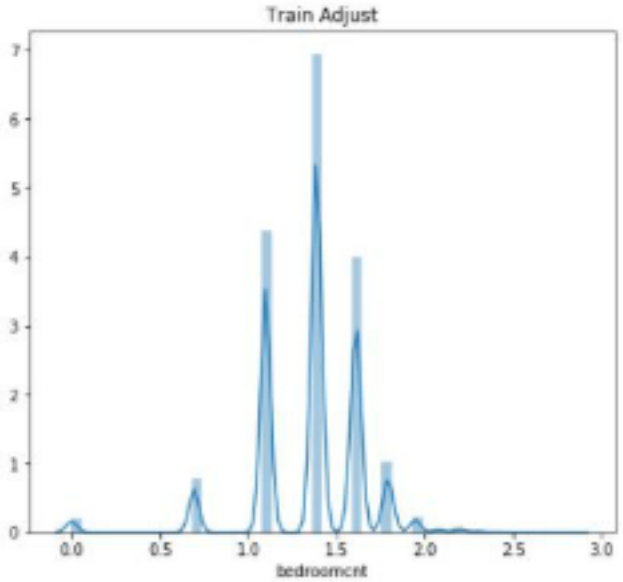
Numerical Features

```
bedroomcnt
Full NaN : 0.10%:   Full NumOfCat : 26   Full CatRatio : 0.00%
Train NaN : 0.03%   Train NumOfCat : 18   Train CatRatio : 0.01%
[1172760L, 731476L, 606787L, 182768L, 118705L, 85942L, 48915L, 13542L, 12763L, 4279L]
```

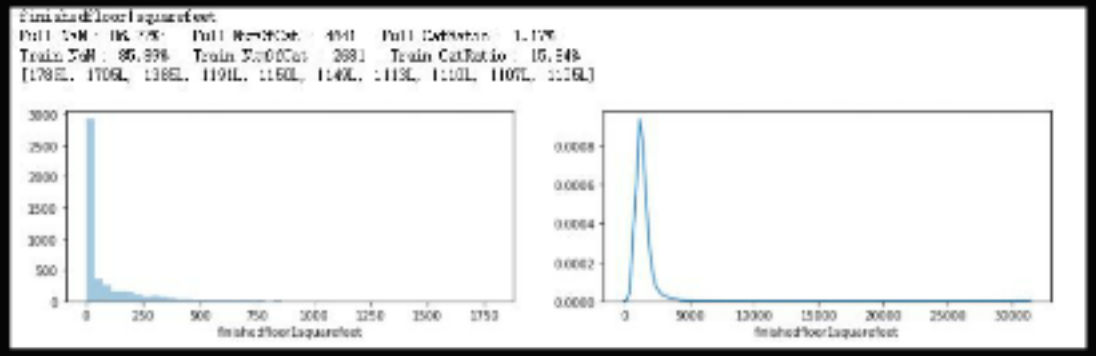


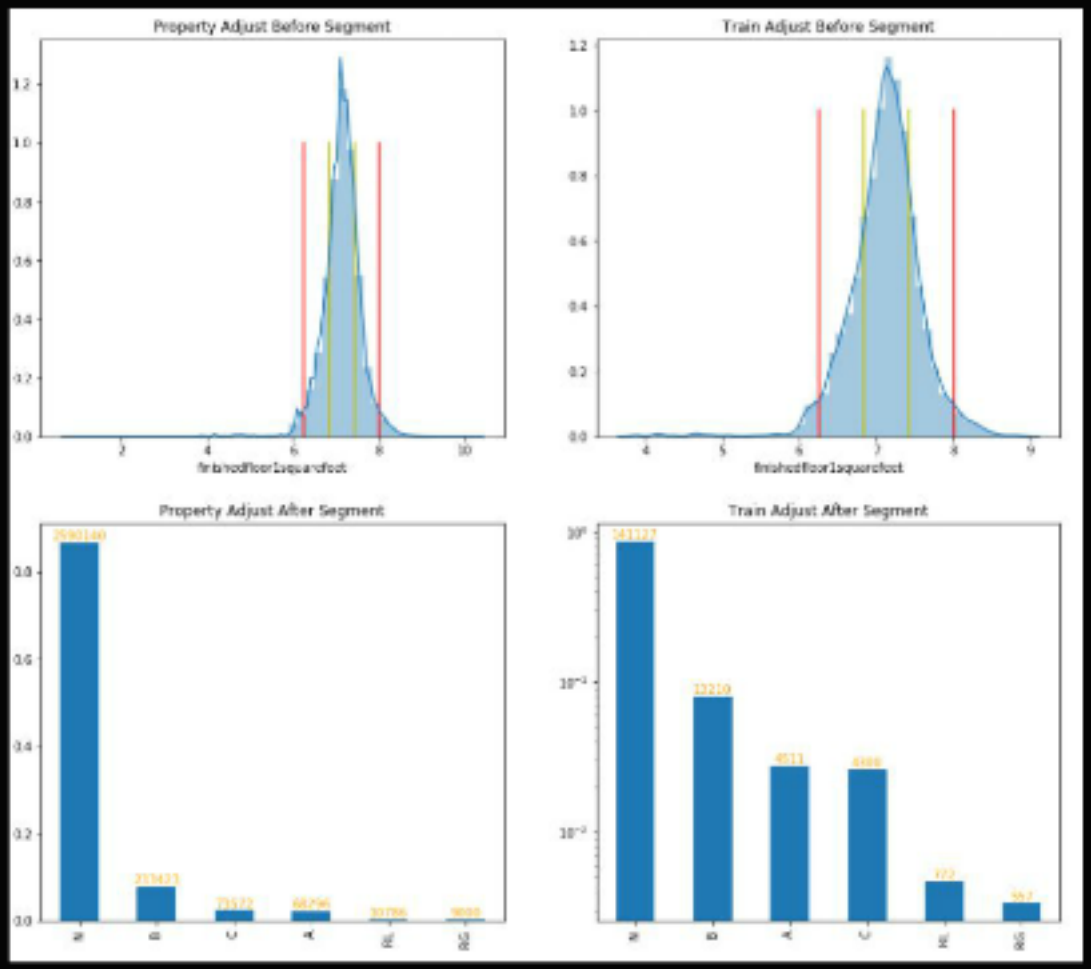






Segment Features

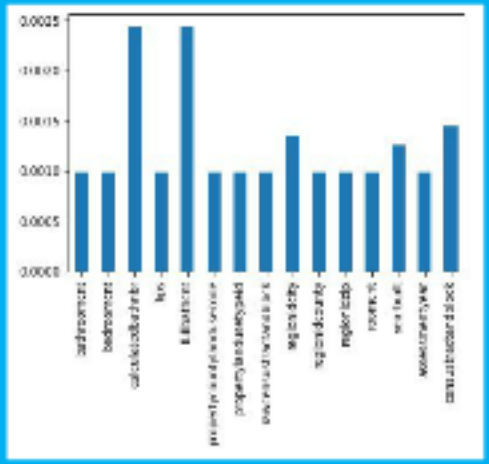




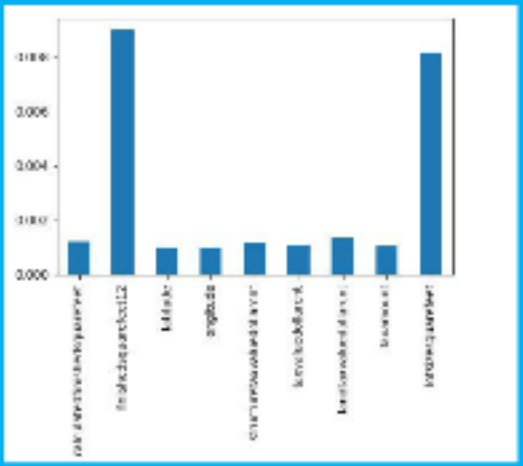
2

1.Preparation
2.Impute Data 1
3.Creating New Features
4.Impute Data 2

Impute By Median



Impute By Mode

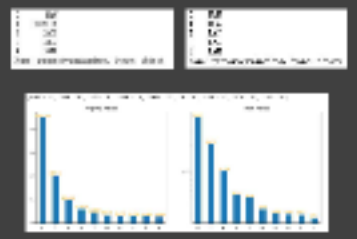


Missing ratio < 10%

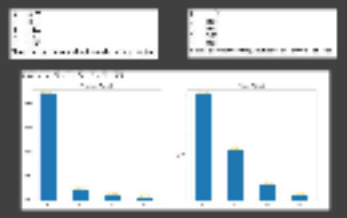
3

1.Preparation
2.Impute Data 1
3.Creating New Features
4.Impute Data 2

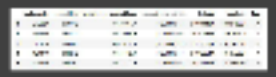
propertyzoningdesc



propertycountylandusecode

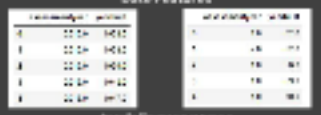


Cluster Geo Info



7 algrithms and each for 8 classes

Date Features

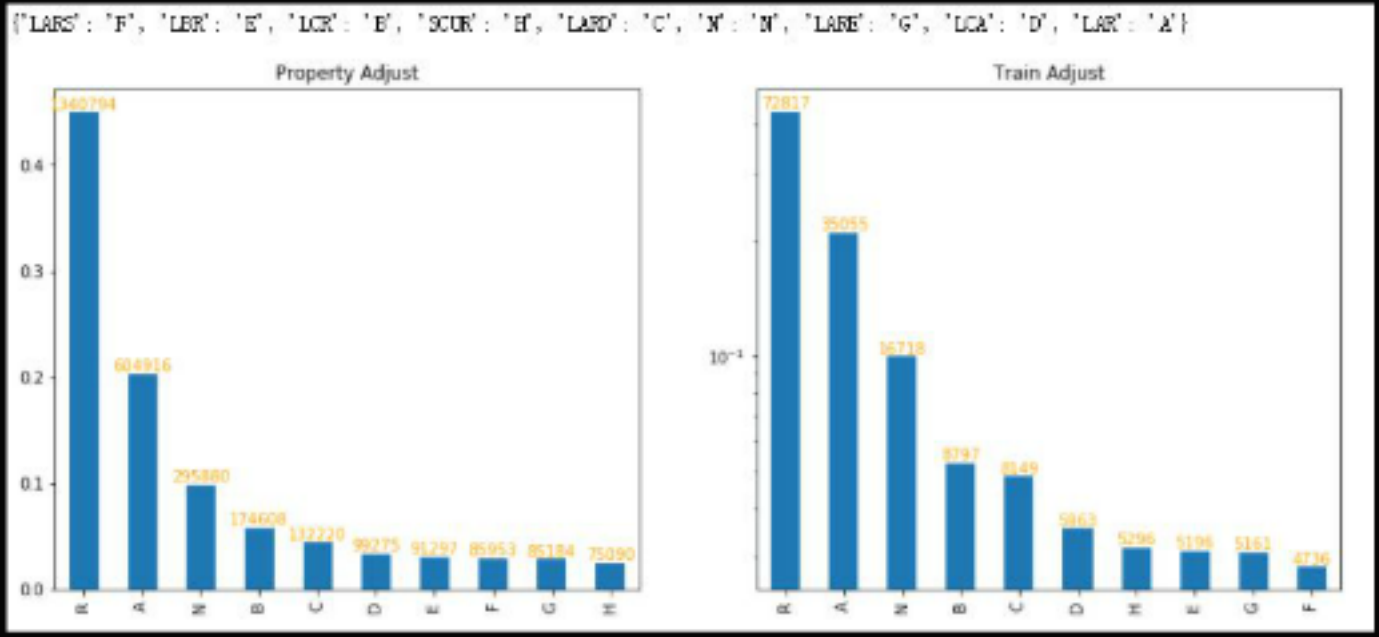


taxdelinquencyyear

propertyzoningdesc

```
0        NaN
1    LCA11*
2      LAC2
3      LAC2
4      LAM1
Name: propertyzoningdesc, dtype: object
```

```
0    NaN
1    LCA
2    LAC
3    LAC
4    LAM
Name: propertyzoningdesc, dtype: object
```

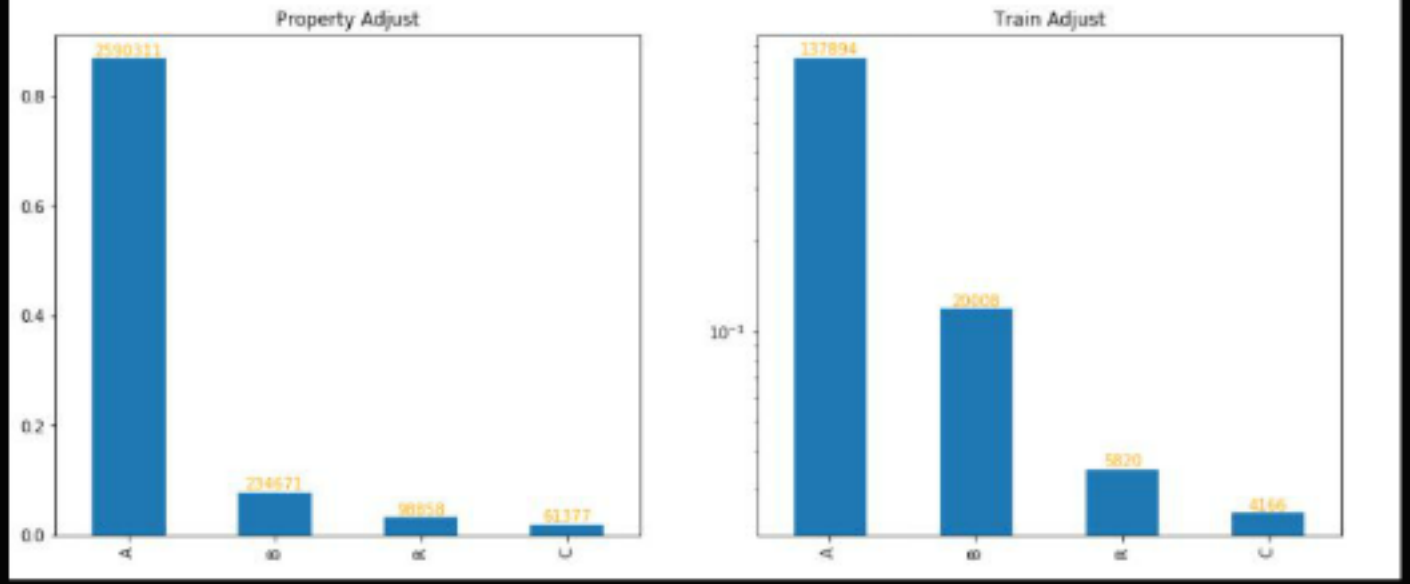


propertycountylandusecode

```
0    010D
1    0109
2    1200
3    1200
4    1210
Name: propertycountylandusecode, dtype: object
```

```
0       D
1    NAN1
2    NAN1
3    NAN1
4    NAN1
Name: propertycountylandusecode, dtype: object
```

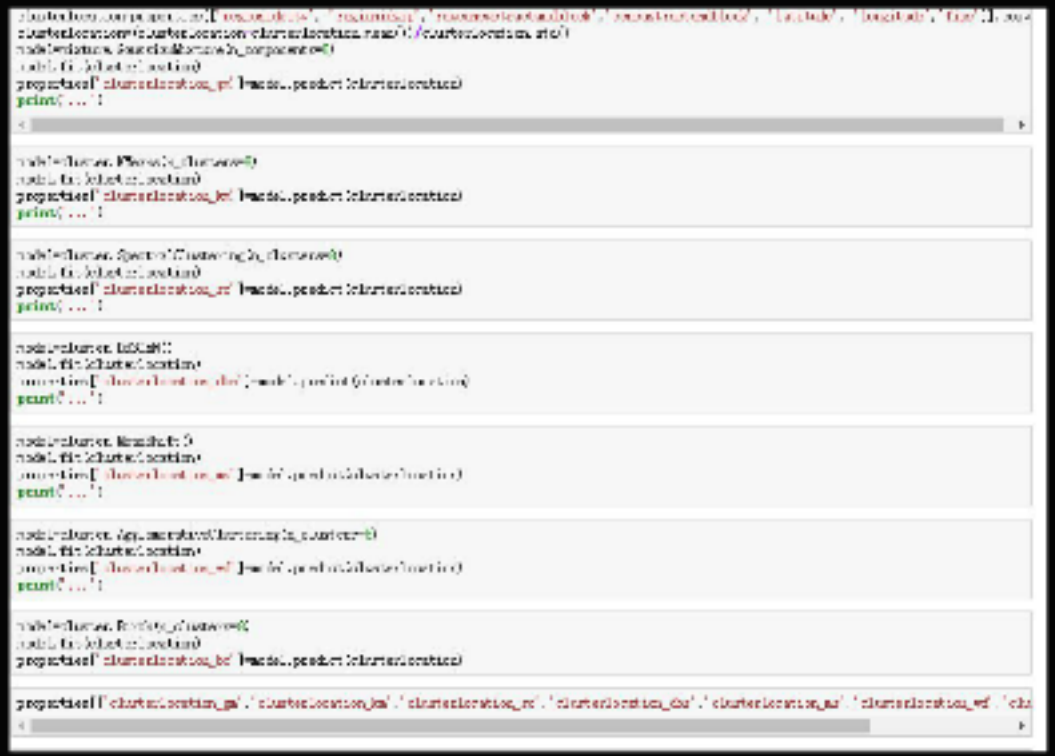
```
{'NAN1': 'A', 'C': 'B', 'D': 'C', 'N': 'N'}
```



Cluster Geo Info

	regionidcity	regionidzip	rawcensustractandblock	censustractandblock	latitude	longitude	fips
0	37688.0	96337.0	60378004.0	6.059032e+13	34144440.0	-118654080.0	A
1	37688.0	96337.0	60378000.0	6.059032e+13	34140432.0	-118625360.0	A
2	51617.0	96095.0	60377032.0	6.059032e+13	33989360.0	-118394632.0	A
3	12447.0	96424.0	60371412.0	6.059032e+13	34148864.0	-118437208.0	A
4	12447.0	96450.0	60371232.0	6.059032e+13	34194168.0	-118365816.0	A

7 algrithms and each for 8 classes

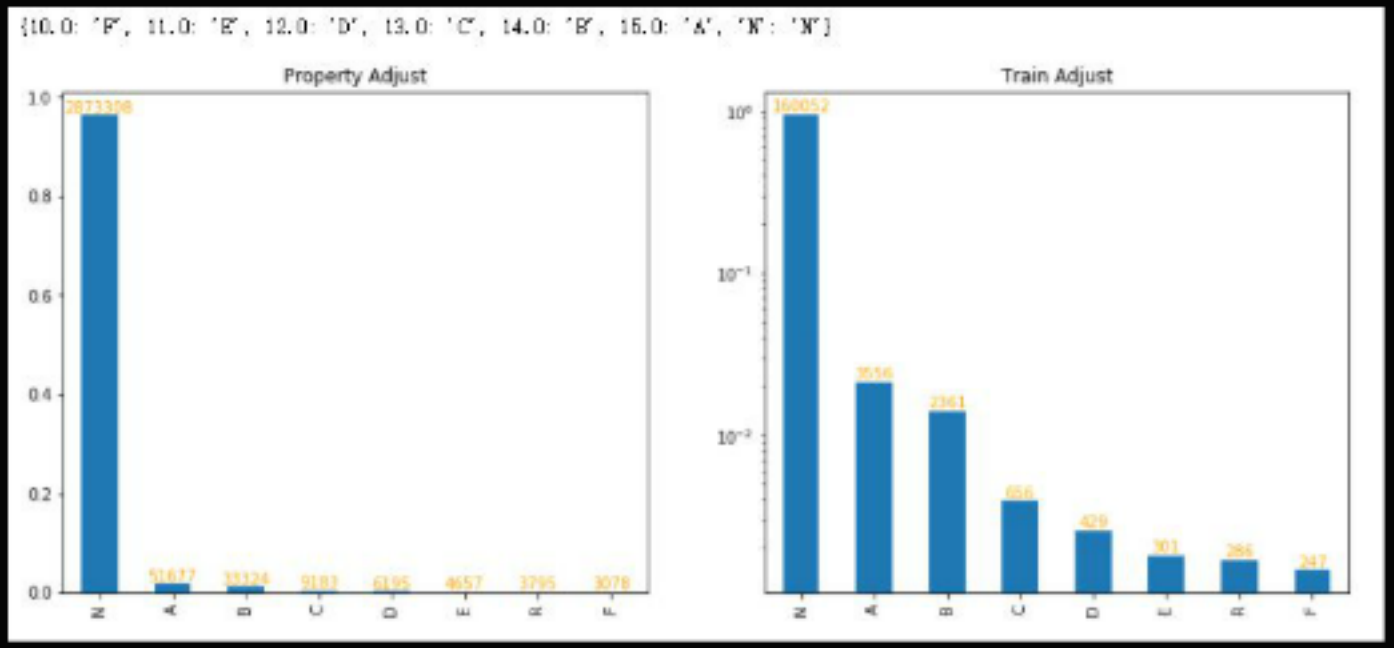


Date Features

	assessmentyear	yearbuilt
0	2016.0	1963.0
1	2015.0	1963.0
2	2016.0	1959.0
3	2016.0	1948.0
4	2016.0	1947.0

	assessmentyear	yearbuilt
0	1.0	62.0
1	2.0	62.0
2	1.0	58.0
3	1.0	69.0
4	1.0	70.0

taxdelinquencyyear

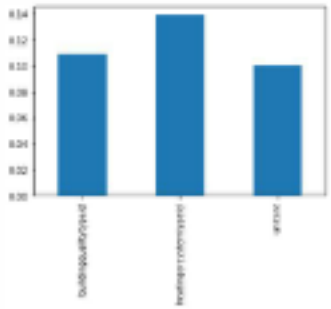


4

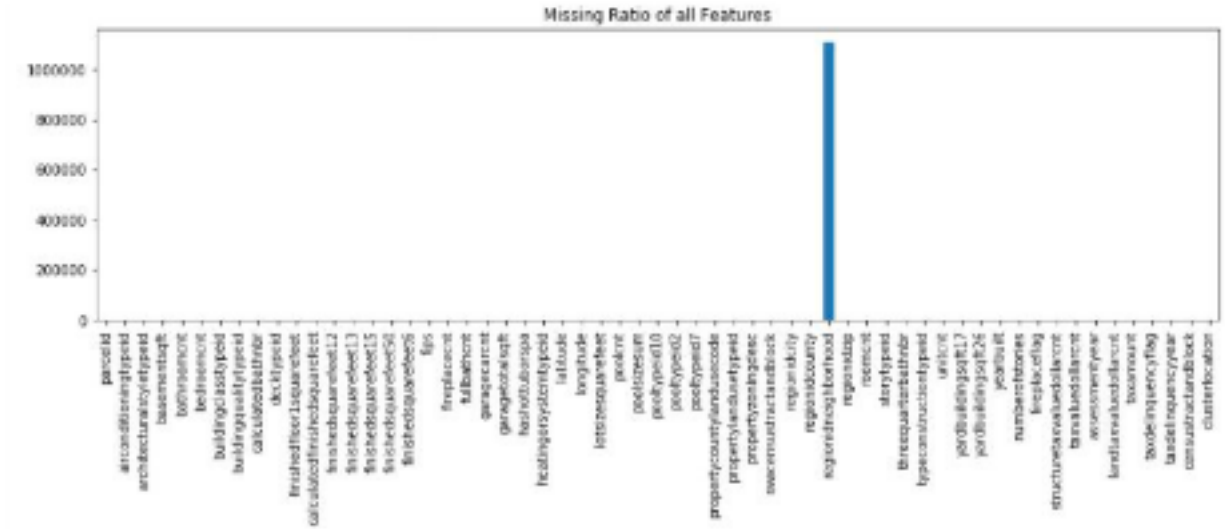
1.Preparation
2.Impute Data 1
3.Creating New Features
4.Impute Data 2

```
imputedatasets=imputefullsets[complete_featurel.copy()
for f in ['buildingqualitytypeid','heatingorsystemtypeid','unitcnt']:
    print(f)
    train_x=imputedatasets[imputefullsets[f].notnull()]
    train_y=imputefullsets[f][imputefullsets[f].notnull()]
    impute_x=imputedatasets[imputefullsets[f].isnull()]
    model = ensemble.RandomForestClassifier()
    model.fit(train_x,train_y)
    print('ok')
    scores=model.score(train_x,train_y)
    print('R2 Score : '+str(scores.mean()))
    imputefullsets.ix[imputefullsets[f].isnull().f]=model.predict(impute_x)
```

```
buildingqualitytypeid
ok
R2 Score : 0.988157806080
heatingorsystemtypeid
ok
R2 Score : 0.989199035403
unitcnt
ok
R2 Score : 0.996027609477
```



10% < Missing ratio < 30%



Zillow's Price Training

Chan Yu Yankai Liu Yifei Bi

Procedures Guide

01

02

03

04

1.XGBoost
2.LightGBM
3.Neural Network
4.Linear Regression

Reporter : Chan Yu

1

XGBoost - Extreme Gradient Boosting

- Regularization
- Parallel Processing
- High Flexibility
- Handling Missing Values
- Tree Pruning

1.XGBoost
2.LightGBM
3.Neural Network
4.Linear Regression

XGBoost - First

Combined XGBoost Predictions

XGBoost - First

Set XGBoost Hyper Parameters

```
xgb_params = {
    'eta': 0.037,
    'max_depth': 5,
    'subsample': 0.80,
    'objective': 'reg:linear',
    'eval_metric': 'mae',
    'lambda': 0.8,
    'alpha': 0.4,
    'base_score': y_mean,
    'silent': 1
}
num_boost_rounds = 250
```

```
First XGBoost predictions:
          0
0  0.003354
1  0.008021
2 -0.002644
3  0.009713
4 -0.001039
```

Validate For First XGBoost:

0.06695654820228315

Combined XGBoost Predictions

```
Combined XGBoost predictions:
           0
0  0.003386
1  0.008179
2 -0.002297
3  0.009634
4 -0.000765
```

XGB1_WEIGHT = 0.8000
XGB2_WEIGHT = 0.2000

Validate For Combined XGBoost:

0.06694858812380969

2

LightGBM

- Optimization in Speed and Memory Usage
- Optimization in Accuracy
- Optimization in Network Communication
- Optimization in Parallel Learning

```
Unadjusted LightGBM predictions:
          0
0  0.021860
1  0.022435
2  0.022740
3  0.021015
4  0.022069
```

Validate For LightGBM

0.06942165970198744

3

Fitting neural network model...

0.0699696842884

Neural Network predictions:

	parcelid	201607	201608	201609	201707	201708	201709
782	14677191	0.004404	0.004402	0.004400	0.004405	0.004403	0.004401
968	11183209	0.004399	0.004403	0.004407	0.004400	0.004403	0.004407
1165	11554091	0.004373	0.004375	0.004378	0.004376	0.004377	0.004379
1351	11742566	0.004386	0.004388	0.004393	0.004387	0.004389	0.004394
1609	14667297	0.004405	0.004404	0.004403	0.004406	0.004405	0.004404

Validate For NN

0.06760387642306702

1.XGBoost
2.LightGBM
3.Neural Network
4.Linear Regression

Model Structure

Input
400 hidden units
relu
dropout 0.4
160 hidden units
relu
dropout 0.6
64 hidden units
relu
dropout 0.5
26 hidden units
relu
dropout 0.6
1 hidden units

4 Linear Regression

1.XGBoost
2.LightGBM
3.Neural Network
4.Linear Regression

OLS predictions:

	parcelid	201607	201608	201609	201707	201708	201709
782	14677191	-0.006553	-0.008514	-0.010476	-0.002206	-0.004168	-0.006130
968	11183209	0.013480	0.011519	0.009557	0.017826	0.015865	0.013903
1165	11554091	-0.005324	-0.007286	-0.009248	-0.000978	-0.002940	-0.004901
1351	11742566	0.023170	0.021208	0.019246	0.027516	0.025554	0.023593
1609	14667297	-0.017631	-0.019592	-0.021554	-0.013284	-0.015246	-0.017208

Validate For OLS

0.068008479612021 1

Combination

- Combining XGBoost, LightGBM, and baseline predictions...
- XGB_WEIGHT = 0.6200
- BASELINE_WEIGHT = 0.0100
- OLS_WEIGHT = 0.0620
- NN_WEIGHT = 0.0800

```
Combined XGB/LGB/baseline predictions:
           0
0   0.007198
1   0.010301
2   0.003876
3   0.010879
4   0.004672
```

Combination

- Combining with XGB/LGB/NN/OLS/baseline predictions...
- XGB_WEIGHT = 0.6200
- BASELINE_WEIGHT = 0.0100
- OLS_WEIGHT = 0.0620
- NN_WEIGHT = 0.0800

```
Combined XGB/LGB/NN/baseline/OLS predictions:
      parcelid  201607  201608  201609  201707  201708  201709
782   14677191  0.0080  0.0079  0.0077  0.0083  0.0082  0.0080
968   11183209  0.0129  0.0127  0.0126  0.0132  0.0130  0.0129
1165  11554091  0.0044  0.0042  0.0041  0.0047  0.0045  0.0044
1351  11742566  0.0142  0.0141  0.0139  0.0145  0.0144  0.0142
1609  14667297  0.0044  0.0043  0.0041  0.0047  0.0046  0.0044
```