supporting fractional investment & tokenized real estate: multiple regression analysis

presented by: marvin lee

final model 00

background 01

data + eda 02

modeling 03

conclusion 04

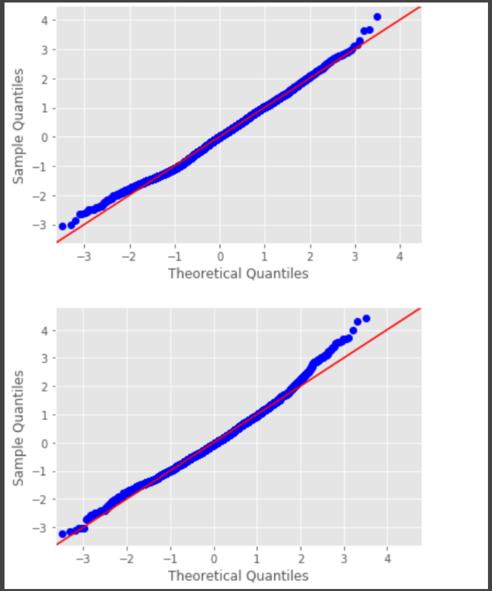
final model:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.821
Model:	OLS	**Adj. R-squared:**	0.818
Method:	Least Squares	**F-statistic:**	324.9
Date:	Sun, 01 Nov 2020	**Prob (F-statistic):**	0.00
Time:	20:02:45	**Log-Likelihood:**	416.17
No. Observations:	4320	**AIC:**	-710.3

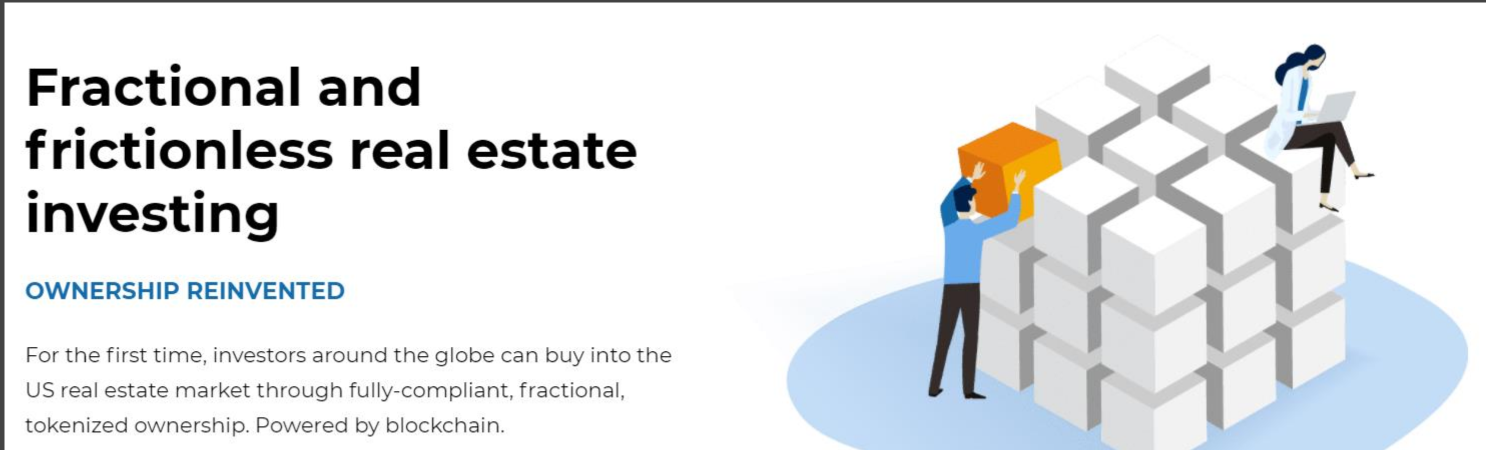
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.6736	0.094	60.336	0.000	5.489	5.858
sqft_above	0.5807	0.012	47.327	0.000	0.557	0.605
has_basement	0.1750	0.008	22.228	0.000	0.160	0.190
sqft_living15	0.3375	0.017	20.112	0.000	0.305	0.370
zip_98004	1.1104	0.030	37.429	0.000	1.052	1.169
zip_98005	0.7246	0.038	18.822	0.000	0.649	0.800

Keys: log-transformed price, zip code dominance, log+scale of sqft_above



background: decentralized, trustless systems for real estate asset management need accurate price predictions

- collateral
- home upgrades
- [insert need here]

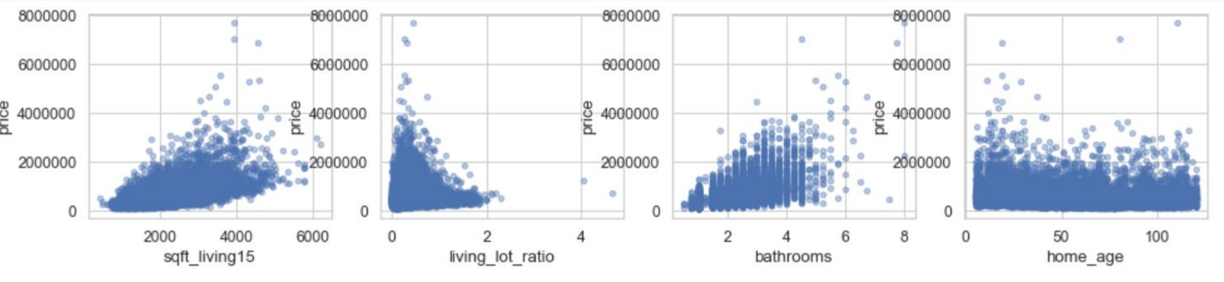


data

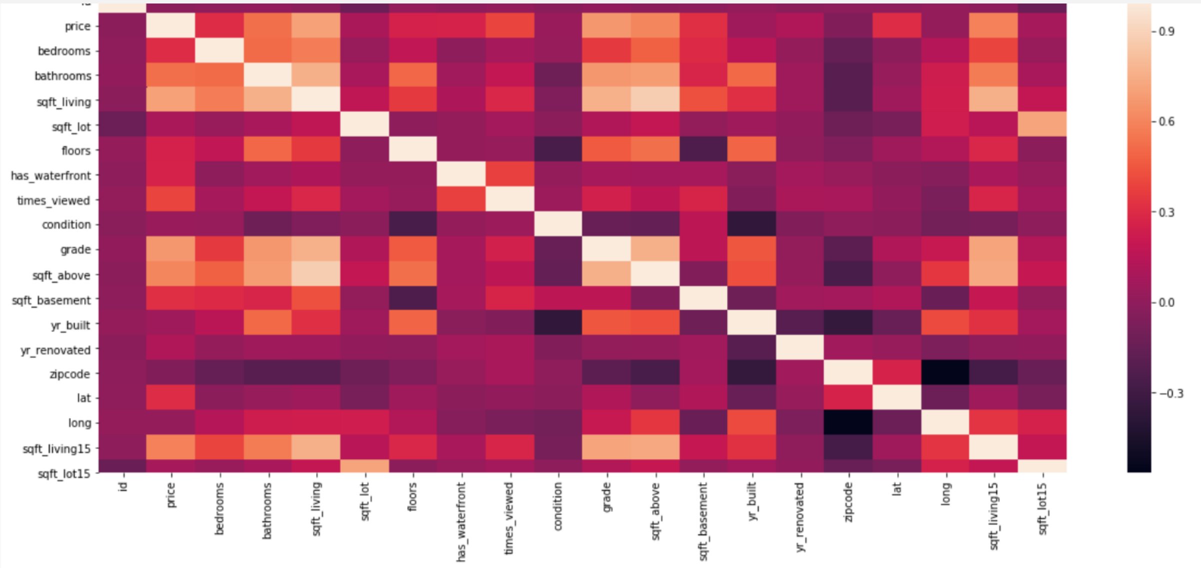
- KC house data set (21,000+ items)
- Output “cleaned” data set to a CSV file
- Most variables were turned into categorical variables
- **Primary goal:** improve R2 / **Secondary goal:** a trustless model
- Used stepwise method for initial “automatic” add/drop statistical analysis
- Log-transformed + scaled continuous variables / Log-transformed target

Created the following features:

home_age, living_lot_ratio, baths_with_reno



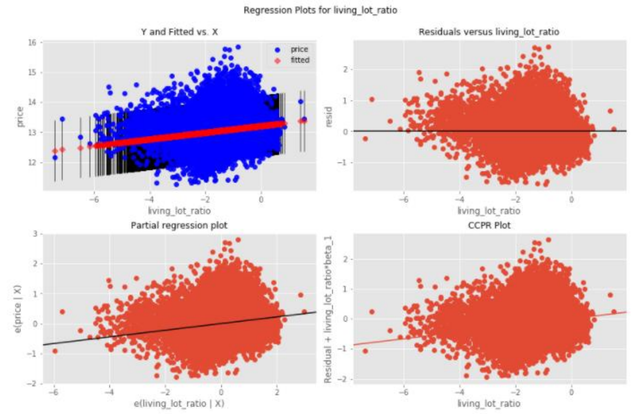
eda



checked for:

- multicollinearity
- subjectivity
- redundancy
- mostly categoricals

modeling




```python

outcome3 = 'price'
predictors3 = train.drop('price',axis=1)
pred_sum3 = '+'.join(predictors3[result_final])
formula3 = outcome3 + '~' + pred_sum3

model3 = ols(formula=formula3, data=train).fit()
model3.summary()
```

Out[108]:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.856
Model:	OLS	**Adj. R-squared:**	0.855
Method:	Least Squares	**F-statistic:**	1041.
Date:	Sun, 01 Nov 2020	**Prob (F-statistic):**	0.00
Time:	15:59:47	**Log-Likelihood:**	3304.7
No. Observations:	17276	**AIC:**	-6411.

Living lot ratio proved to have homoscedasticity and was therefore unfit to be included in the final model.

Log-transformation of the target variable + stepwise addition led to the best R-squared value

features

stepwise

zips + minimum other features

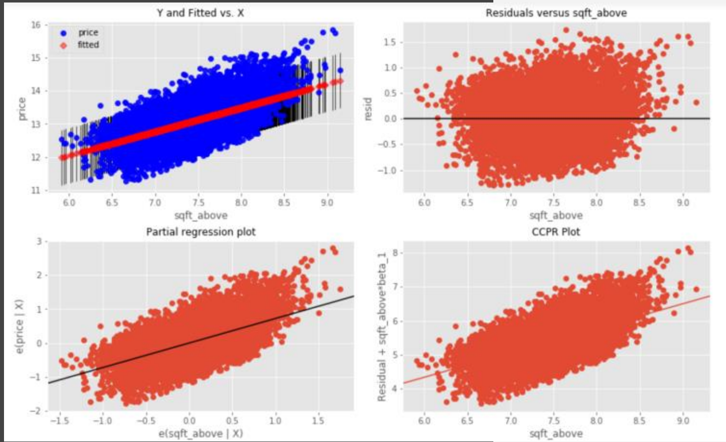
In [129]:

```
include1 = [i for i in train if 'zip' in i]
include2 = ['sqft_above','has_basement','home_age','living_lot_ratio']

outcome3 = 'price'
predictors3 = train.drop('price',axis=1)
pred_sum3 = '+'.join(predictors3[include1+include2])
formula3 = outcome3 + '~' + pred_sum3

model3 = ols(formula=formula3, data=train).fit()
model3.summary()
```

Out[129]:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.823
Model:	OLS	**Adj. R-squared:**	0.822
Method:	Least Squares	**F-statistic:**	1094.
Date:	Sun, 01 Nov 2020	**Prob (F-statistic):**	0.00
Time:	17:19:55	**Log-Likelihood:**	1517.0
No. Observations:	17276	**AIC:**	-2886.
Df Residuals:	17202	**BIC:**	-2312.
Df Model:	73		
Covariance Type:	nonrobust		



R-squared of 0.823

2 variables with p-value > 0.05

conclusion + the future

base collateral levels on:
location + sqft_above

segmented modeling
high prices vs. lower prices

study grading
subjective variables

features didn't pass
but may still hold insights

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.607e+05	8.44e+04	-3.089	0.002	-4.26e+05	-9.53e+04
has_waterfront	8.138e+05	1.62e+04	50.226	0.000	7.82e+05	8.46e+05
sqft_above	206.4688	3.368	61.304	0.000	199.867	213.070
has_basement	9.21e+04	3554.150	25.913	0.000	8.51e+04	9.91e+04
sqft_living15	85.4519	3.169	26.968	0.000	79.241	91.663
home_age	198.0654	77.734	2.548	0.011	45.698	350.433
living_lot_ratio	-8.247e+04	7794.683	-10.581	0.000	-9.78e+04	-6.72e+04

04

thank you

www.github.com/emel333