Zillow®

Regression Modeling

Amanda Gomez
Junior Data Scientist

March 24, 2021

Agenda

ES

Executive Summary

Discuss the affects of features to our home value target, goals, and recommendations to minimize error in predictive models.

DA

Data Analysis

Review findings discovered while building a predictive model for home values, and how beneficial the use of my model will be.

C

Conclusion

Wrap up analysis with reflections on the predictive model.

A

Appendix

Executive Summary

Goals

Provide insight into the creation of my home value predictive value for single unit properties, data was based on properties that sold during the realty "hot" months.

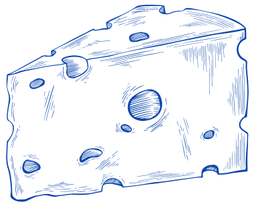
Use features to predict the future!

Implementation of valuable features greatly improve models!

Predicted Error DECREASE

My final predictive model decreased the amount of errors by 77%!

Big Data



We have a big data set, and some equally sized holes.

Data Acquisition

My data was gathered from our Zillow SQL database.

- I filtered my query using:
  - `transactiondate between "2017-05-01" and "2017-08-31"`
  - `propertylandusetypeid IN ('260', '261', '263', '264', '265', '266', '269', '275')`
  - `unitcnt = 1`

Data

Data Acquisition

- I also filtered my query's columns as requested to :
  - `taxvaluedollarcnt`
  - `calculatedfinishedsquarefeet`
  - `bedroomcnt`
  - `bathroomcnt`
    - `parcelid`

Nulls

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24923 entries, 0 to 24922
Data columns (total 5 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   parcelid                      24923 non-null  int64
 1   calculatedfinishedsquarefeet  24921 non-null  float64
 2   bedroomcnt                    24923 non-null  float64
 3   bathroomcnt                   24923 non-null  float64
 4   taxvaluedollarcnt             24923 non-null  float64
dtypes: float64(4), int64(1)
```

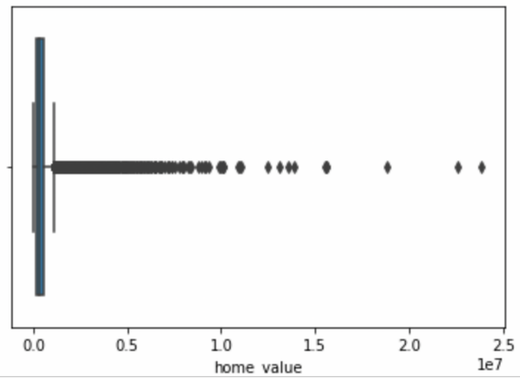
```
#check for blanks/nulls
df.isnull().sum()
```

```
parcelid                        0
calculatedfinishedsquarefeet    2
bedroomcnt                      0
bathroomcnt                     0
taxvaluedollarcnt               0
dtype: int64
```

Outliers

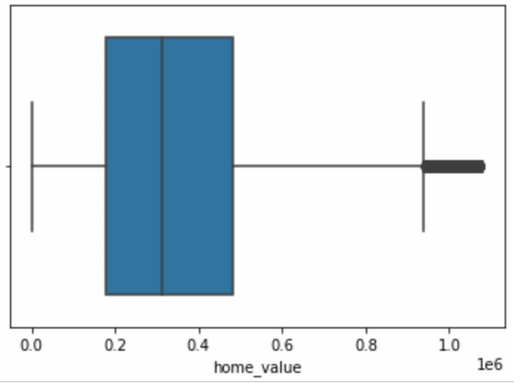
```
sns.boxplot(x="home_value", data=df)
# 5-8 major outliers skewing the data
```

```
<AxesSubplot:xlabel='home_value'>
```



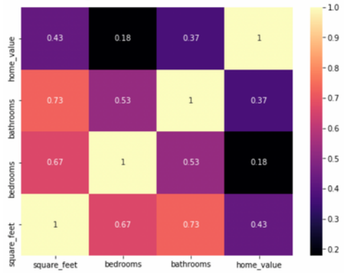

```
# Without home_value outliers
sns.boxplot(x="home_value", data=df)
```

```
<AxesSubplot:xlabel='home_value'>
```



Repercussions

Although home_value outlier removal improved the distribution of data, due to the minimal features, my correlation coefficients suffered greatly.



Exploration / Evaluation

Hypothesis Testing

Since all of my features were quantitative, I used correlation tests.

```
#assign your null hypothesis
nullh = "No correlation between square footage and home value."

#and your alternative hypothesis
alth = "There IS a correlation between square footage and home value."

#the feature you are testing
x1 = train_scaled.square_feet

#against your target
y1 = train_scaled.home_value
```

```
exp.correlation_test(nullh, alth, x1, y1)
```

```
Reject null statment: No correlation between square footage and home value.
There is a linear relationship.
Although, it is a positive weak one.

    Correlation Coefficient: 0.42675216032970326
    p: 0.0
```

Feature Engineering

Interesting findings!

```
f_feature = mo.select_kbest(X_train_scaled, y_train, 3)
f_feature
```

```
['square_feet', 'bedrooms', 'bathrooms']
```

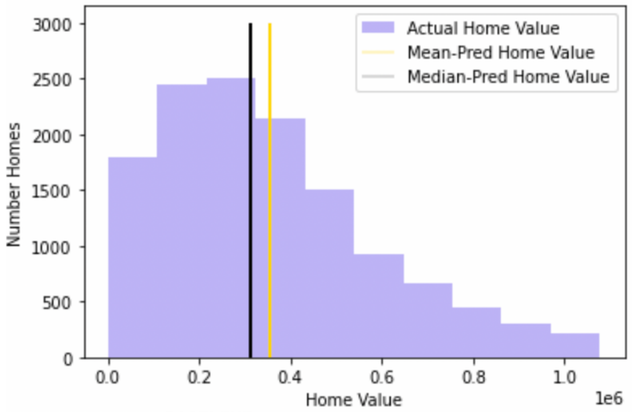
```
rfe_feature = mo.rfe(X_train_scaled, y_train, 3)
rfe_feature
```

```
['square_feet', 'bedrooms', 'bathrooms']
```

Interestingly, `bathrooms` seems to hold the most weight!

Modeling

Identifying the Baseline

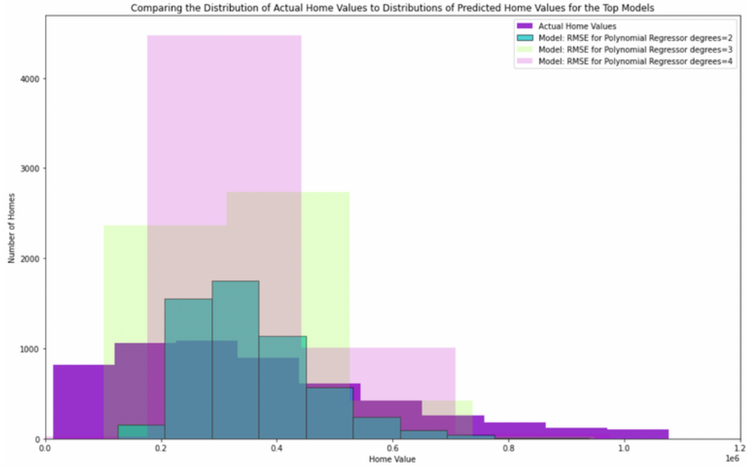



```
BASELINE:

RMSE using Median
Train/In-Sample: 236389.0
Validate/Out-of-Sample: 234776.68
```

Model Ran on Train/Validate Sets

```
BASELINE:

RMSE using Median
Train/In-Sample: 236389.0
Validate/Out-of-Sample: 234776.68

RMSE for OLS using LinearRegression

Training/In-Sample: 206116.98
Validation/Out-of-Sample: 206824.56

RMSE for LassoLars

Training/In-Sample: 206117.18
Validation/Out-of-Sample: 206816.23
```

```
RMSE for Polynomial Regressor
degrees=2

Training/In-Sample: 204944.07
Validation/Out-of-Sample: 205527.11

RMSE for Polynomial Regressor
degrees=3

Training/In-Sample: 204511.14
Validation/Out-of-Sample: 206494.85

RMSE for Polynomial Regressor
degrees=4

Training/In-Sample: 204173.26
Validation/Out-of-Sample: 206717.81
```



Testing the Best Model

First iteration: RMSE degree 2 Polynomial Regressor

```
y_test = pd.DataFrame(y_test)

#predict on test
y_test['home_value_pred_lm2'] = lm2.predict(X_test_degree2)

# evaluate: rmse
rmse_test = mean_squared_error(y_test.home_value, y_test.home_value_pred_lm2)**(0.5)

print(f"""
    RMSE for Polynomial Regressor degrees=2

    Test/Out-of-Sample Performance: {round(rmse_test, 2)}
""")
```

```
    RMSE for Polynomial Regressor degrees=2

Test/Out-of-Sample Performance: 201740.36
```

This amount of error definitely calls for a second iteration!

Second Iteration

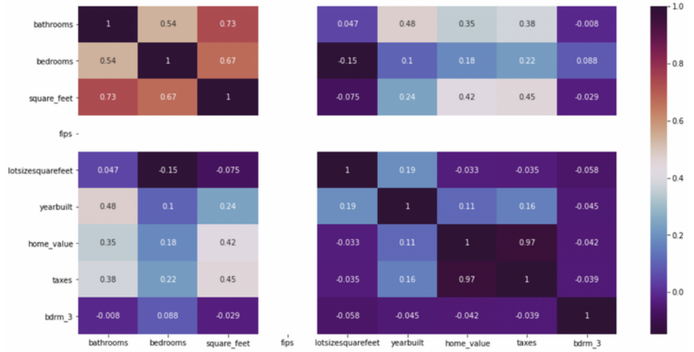
Data Acquisition

I brought in a few more features for the second iteration, and created a new feature `bdrm_3` to represent 3 bedroom homes.

parcelid	bathrooms	bedrooms	square_feet	fips	lotsizesquarefeet	yearbuilt	home_value	taxes	bdrm_3
11721753	2.0	3.0	1316.0	6037	5672.0	1923	205123.0	2627.48	1
11289917	2.0	3.0	1458.0	6037	8284.0	1970	136104.0	2319.90	1

Exploration

As I introduced more features, my correlation coefficients increased



Feature Engineering

Not too surprising.

```
f_feature = mo.select_kbest(X_train_scaled, y_train, 3)
f_feature
```

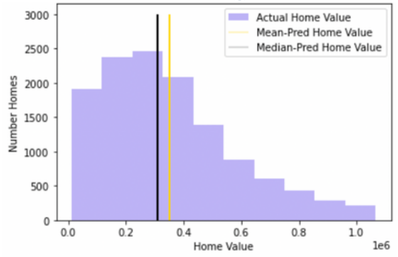
```
['bathrooms', 'square_feet', 'taxes']
```

```
rfe_feature = mo.rfe(X_train_scaled, y_train, 3)
rfe_feature
```

```
['bathrooms', 'bedrooms', 'taxes']
```

Identifying the Baseline

Median has less errors again and will be used.




```
BASELINE:

RMSE using Median
Train/In-Sample: 232113.75
Validate/Out-of-Sample: 235030.88
```

Model Ran on Train/Validate Sets

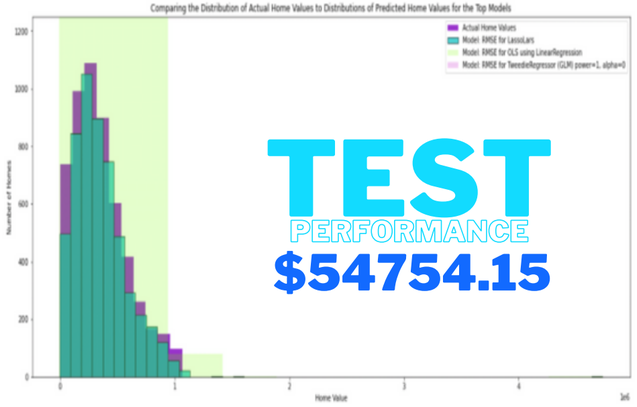
```
BASELINE:

RMSE using Median
Train/In-Sample: 232113.75
Validate/Out-of-Sample: 235030.88
```

```
RMSE for OLS using LinearRegression

Training/In-Sample: 57184.28
Validation/Out-of-Sample: 82442.8
```

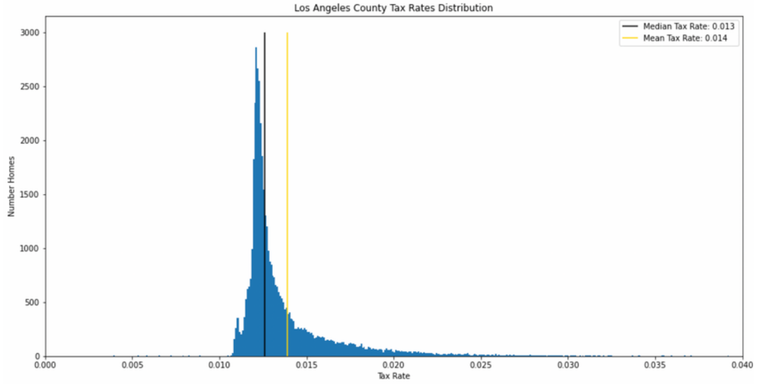
```
RMSE for LassoLars

Training/In-Sample: 57186.15
Validation/Out-of-Sample: 82413.04
```

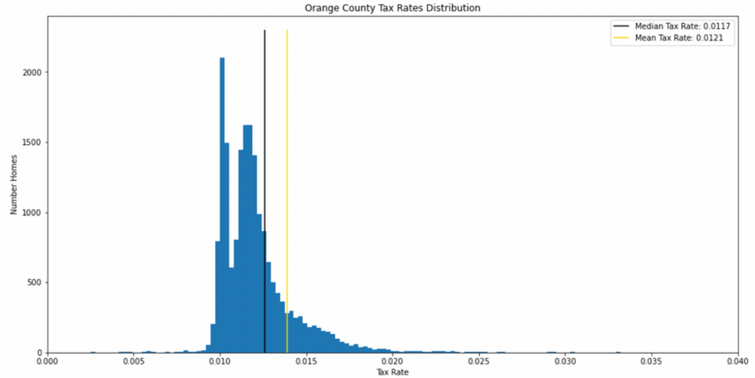
```
RMSE for TweedieRegressor (GLM)
power=1, alpha=0

Training/In-Sample: 228276.06
Validation/Out-of-Sample: 230762.6
```

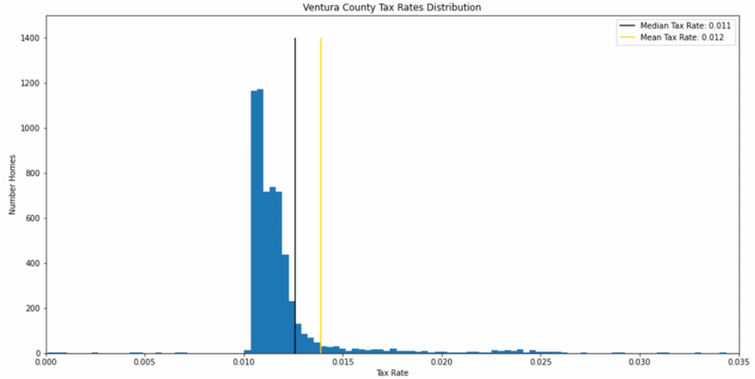
```
RMSE for Polynomial Regressor
degrees=2

Training/In-Sample: 1047628.59
Validation/Out-of-Sample: 1044229.95
```



County Tax Rates







Conclusion

Performance

The model I have presented to you out performs the baseline by 77%

Next Steps

- We have the technology!
    - Given more time and resources, an even more accurate predictive model can be created to make our Zestimates® reign over our competitors.

Appendix

GitHub

https://github.com/o0amandagomez0o/regression_project

Data Dictionary

column_name	description	key	dtype
`bathrooms` / `bathroomcnt`	Number of bathrooms in home including fractional bathrooms		float64
`bedrooms` / `bedroomcnt`	Number of bedrooms in home		float64
`square_feet` / `calculatedfinishedsquarefeet`	Calculated total finished living area of the home		float64
`fips`	Federal Information Processing Standard code - see https://en.wikipedia.org/wiki/FIPS_county_code for more details		float64 / int64
`lotsizesquarefeet`	Area of the lot in square feet		float64
`parcelid`	Unique identifier for parcels (lots)		int64
`yearbuilt`	The Year the principal residence was built		float64 / int64
`home_value` / `taxvaluedollarcnt`	The total tax assessed value of the parcel		float64
`taxes` / `taxamount`	The total property tax assessed for that assessment year		float64
`county`	The county the property is located.		object
`state`	The state the property is located.		object
`bdrm_3`	Identifies if property is a 3 bedroom home.	1: 3 bedroom home 0: not a 3 bdrm home	int64
`tax_rates`	Calculated tax rate = `taxes` / `home_value`		float64

Thank you for your time!

Questions?