Zillow Zestimate

Capstone #1 - Theo Pujianto

Introduction

Objectives:

Utilize machine learning technique to predict logerror from Zestimate compare to actual sales price.

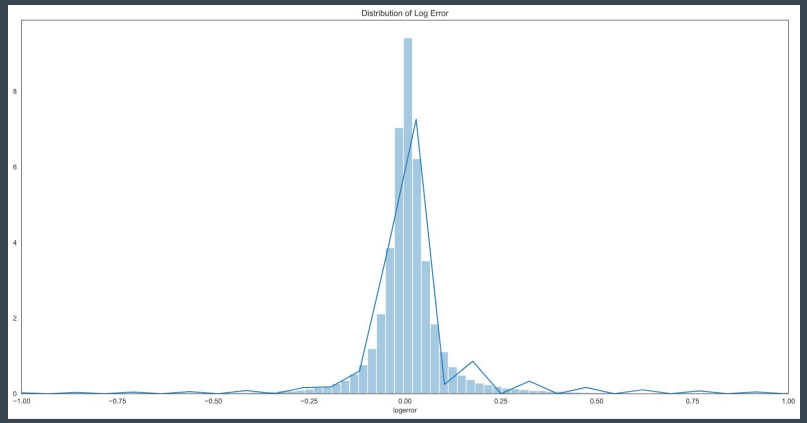
logerror=log(Zestimate)−log(SalePrice)

Data

Kaggle provided real estate data from three counties in and around Los Angeles, CA. Each observation had 56 features; no additional features from outside data sources were allowed in the analysis. The data was split into two files: a training set with the actual logerror and feature information for 90,725 properties and a prediction set with only the feature information for 2,985,217 properties.

EDA - Log Error Distribution

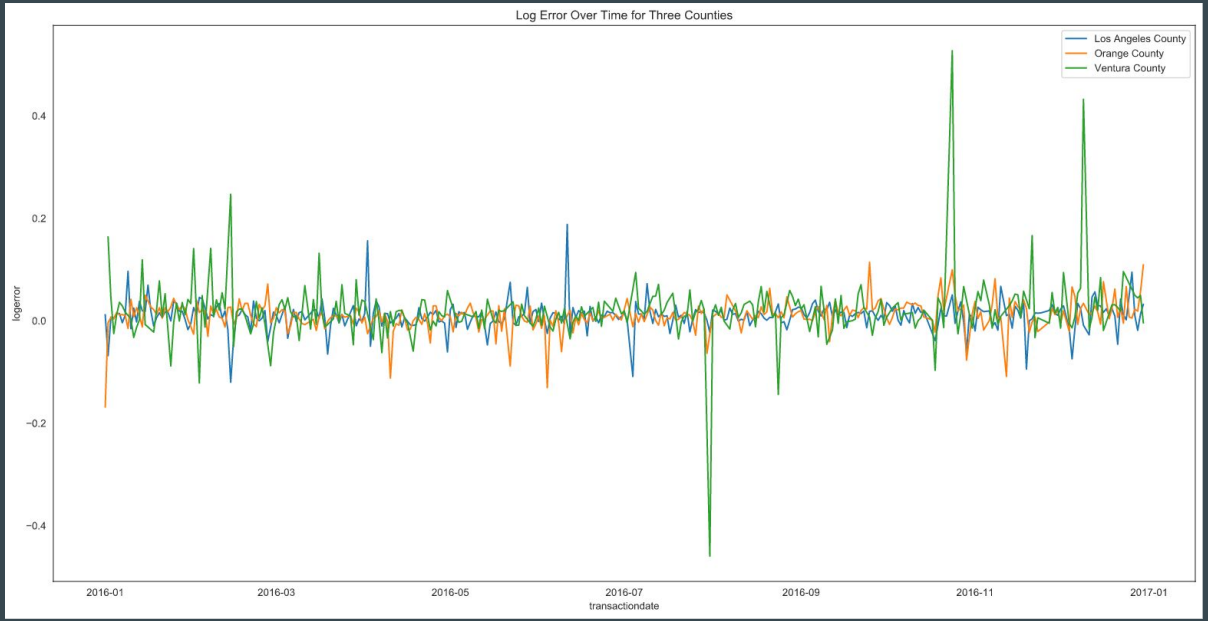
Firstly, let's check the distribution of the log error data. We would like to ensure if it has normal distribution.



We observed normal distribution to the log error; this means that the random sample we use to test our model will have the same distribution as our overall data, and we can guard against overfitting.

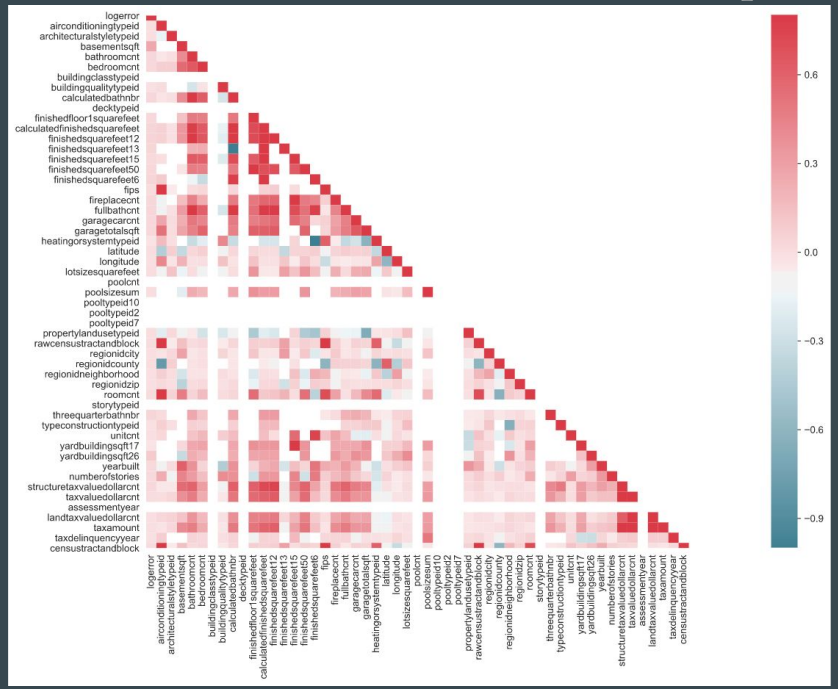
EDA - Log Error vs. Time

Ventura County recorded highest error compare to the other 2 counties.



EDA - Correlation

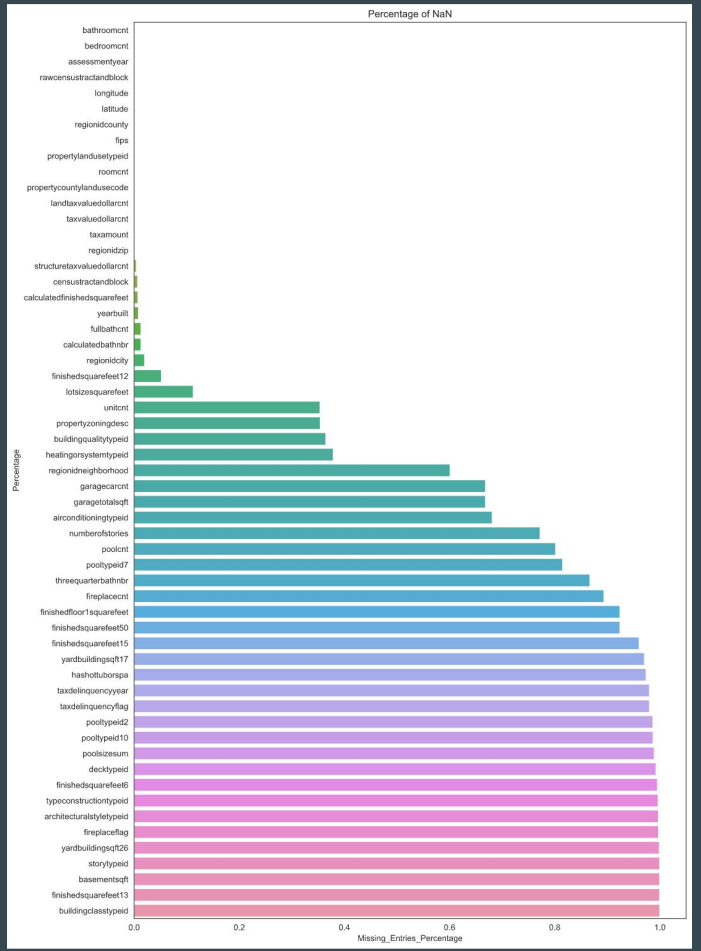
There is no features strongly correlated with ‘logerror’. However, there are group of features strongly correlated to each other. We would later deep dive into each group closely.



Feature Engineering - Data Preparation

Visualize missing values percentage over all features.

- There are a lot of features have more than 80% null entries



Feature Engineering - Imputation Strategy

1. Features to be grouped per category, e.g. pool, tax, square-footage, etc.
2. Eliminate and/or combine features containing similar information, i.e. highly correlated.
3. Impute the null values with 0 or mean value for the numerical features. Deep dive for the categorical features.

Modeling - PyCaret

In this project, we are using pycaret modules to solve our regression problem. Below are the list of steps :

1. Data preparation - remove outlier
2. Setup regression model and compare model - here we were looking at Linear Regression, Ridge Regression, Lasso Regression, Light Gradient Boosting, Bayesian Ridge, and Catboost.
3. Pick a model that is appropriate for the project from the comparison we've done.
4. Tune the model
5. Interpret
6. Make predictions

Modeling - Model Comparison

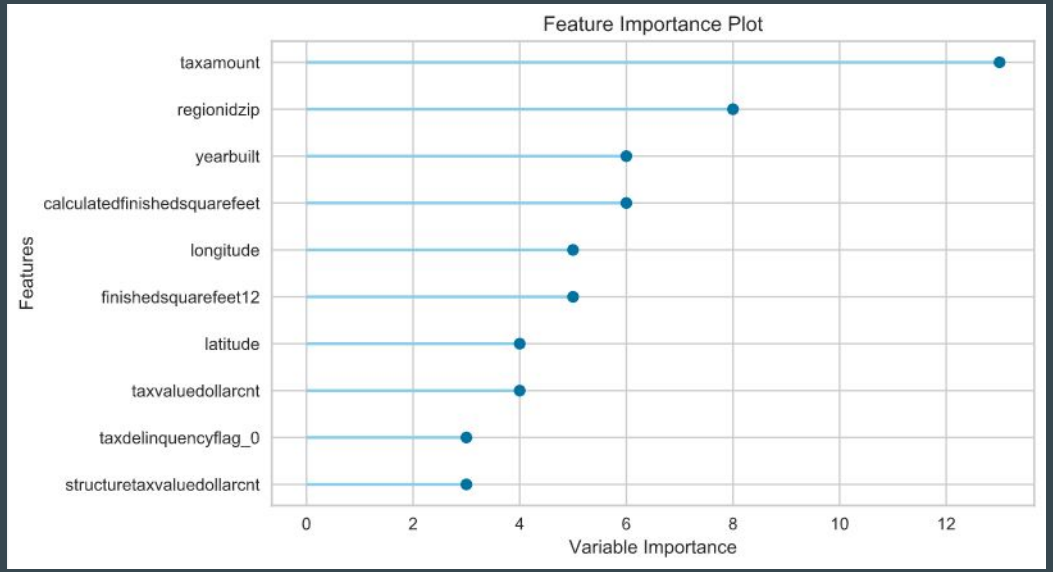
After setting up the pycaret regression, all regression models can be compared. For our project, we chose Linear Regression, Ridge Regression, Lasso Regression, Light Gradient Boosting, Bayesian Ridge, and Catboost.

Lightgbm performed better than other models.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
0	Light Gradient Boosting Machine	0.0524	0.0066	0.0813	0.0229	0.0675	-0.1214	6
1	CatBoost Regressor	0.0524	0.0066	0.0813	0.0226	0.0677	-0.1261	20.76
2	Bayesian Ridge	0.0525	0.0067	0.0818	0.01	0.0697	-0.149	49.48
3	Ridge Regression	0.0534	0.0068	0.0822	0.0004	0.0679	-0.1005	7.034
4	Lasso Regression	0.0528	0.0068	0.0822	-0.0002	0.0707	-0.2381	2.806
5	Linear Regression	1.678e+07	2.622e+17	4.927e+08	-3.916e+19	2.181	-1.102e+08	17.2

Modeling - Feature Importance

Once we tuned our model, we can look at the feature value on our model.



Modeling - Feature Importance

‘taxamount’ - represent the tax influence our model the most.

‘regionidzip’ - represent the location is the 2nd highest feature value.

Tax and location are arguably what people look at the most when looking for a house. Those features, unfortunately, might change their influence to the price of the house.

For example, a certain area used to be expensive but due to recent natural disaster nearby the area, the house price plummetted.

Modeling - Final Result

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Light Gradient Boosting Machine	0.0527	0.0068	0.0822	0.0157	0.0701	-0.1714