Libby Amir
STA160
7 June 2025

Zestimate: The Price Is (Not Always) Right

Introduction

Zillow's Zestimate is widely used to estimate home values, but in some cases it differs significantly from the actual selling value of houses. While the existing algorithm is "based on 7.5 million statistical and machine learning models that analyze hundreds of data points on each property," Zillow hosted a Kaggle competition with the goal of further improving their algorithm by understanding what contributes most to its error.

I find this topic both interesting and important because the real estate market in California is so expensive and volatile, such that small errors in home evaluation can mean tens or even hundreds of thousands of dollars. Accurate pricing is both economically and socially critical in this market, and inaccurate estimates can mislead buyers, sellers, and real estate professionals. Housing in California is a notorious topic because everyone wants to be a homeowner, but there are many hoops to jump through in order for that to be possible, and even then it's not possible for everyone. Having a basic understanding of property value and real estate is obligatory for any young adult hoping to one day own a house in California, so I truly made this project a learning opportunity for my future self. Zillow is one of the most commonly used websites for people looking to buy houses, and I wanted to understand its reliability as well as its pitfalls.

In this analysis, I set out to explore which property features and conditions are most associated with high prediction errors in the Zestimate. Using Zillow's 2016–2017 data from the Kaggle competition, I combined property and transaction records, ran exploratory data analysis, and built a regression model to better understand where and why the Zestimate tends to fail.

Data Overview and Exploratory Data Analysis (EDA)

My analysis utilizes the Kaggle Zestimate Prediction dataset, including two key components:

1.) Training data from 2016 and 2017, with actual sale prices and Zillow's predicted values (to calculate log error)

2.) Property characteristics (ie. square footage, year built, num of beds/baths, tax value etc.)

After merging the training and property data by the unique identifier "parcelid", I obtained a combined dataset with over a million observations and 60 attribute variables.

Summary of preprocessing steps for data preparation and cleaning:

The data's size, detail, and complexity are promising for gaining meaningful understanding about real world patterns, but also poses challenges in terms of data modeling and manipulation. Thus, I began by organizing and cleaning the dataset using the following steps and strategies:

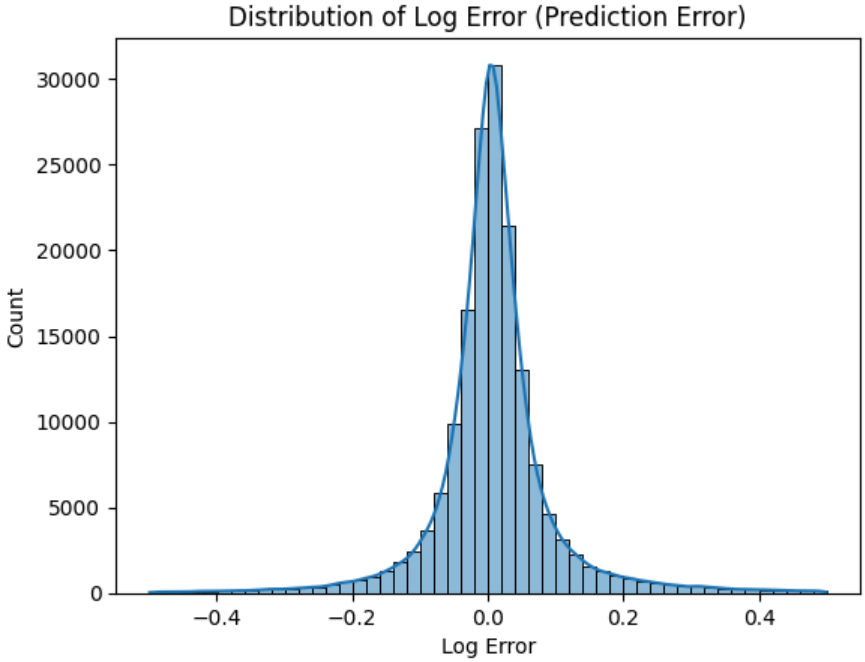
- Merged training and property data by parcel ID (left join because the properties are only meaningful if both their sale value and Zestimate are available)

- I tried filtering to remove extreme outliers (ie. homes with over 10 bedrooms or 10,000 sqft) because EDA showed that these outliers may create unnecessary noise
    - However, I decided to keep most of the data because the unique properties are part of the real-world challenge at hand, and I wanted to understand if unusual properties are more prone to error in the Zestimate

- Handled missing values by dropping rows with NA values within key columns needed for analysis (only appropriate given the size of the data: dropping a couple observations had a negligible impact on the overall shape and distribution of the data)

- Encoded categorical variables for modeling using one-hot encoding ("regionidzip" and "propertylandusetypeid" are two examples of variables that needed to be encoded, as they are numerical yet categorical variables, and need to be treated as such)

- Created some new attributes, such as home age (year - year built), and most importantly, the response variable of absolute log error (abs(logerror)



Before taking the absolute value of the log error, I wanted to understand more about the shape and spread of the Zestimate error in general. As seen in the graph above, the log error has an approximately normal distribution centered around 0. This is important to my analysis because it suggests that Zestimate is unbiased (on average), and does not lean towards over/underestimating a majority of the time. It further indicates that the errors are generally small and random, which bodes well for the model's reliability.

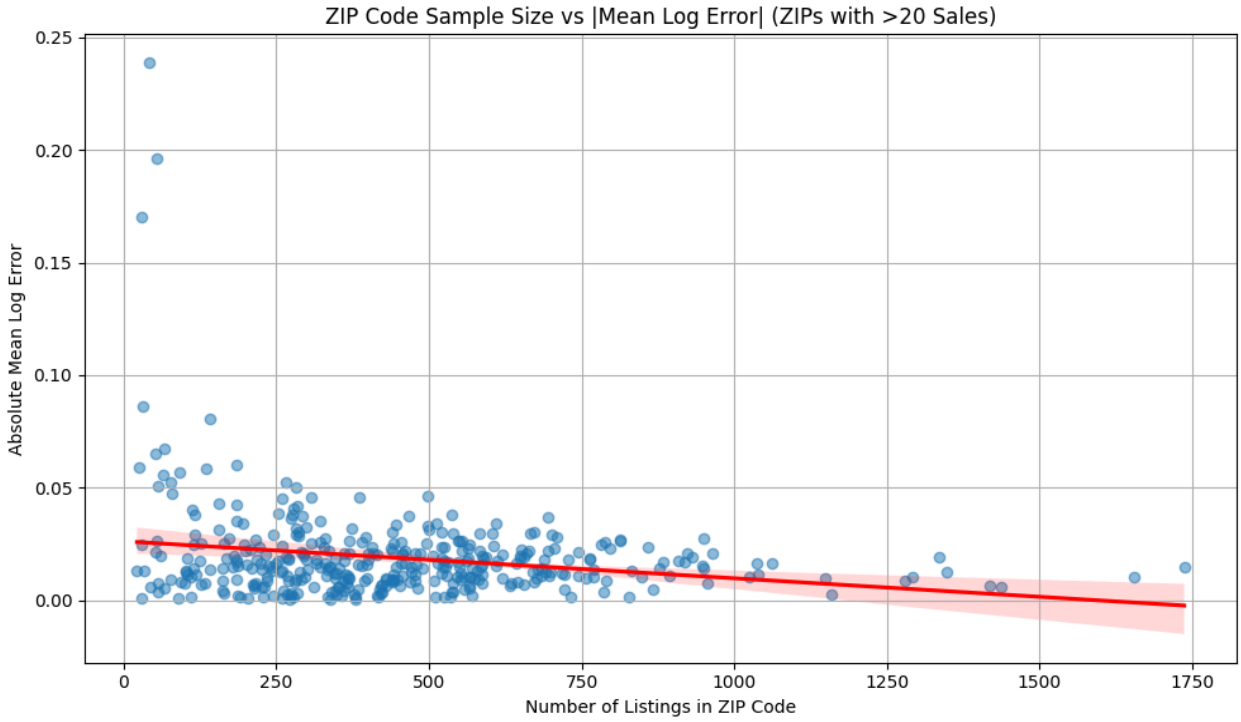
Some of the key patterns and relationships discovered throughout EDA:

I explored several questions and hypotheses through targeted EDA using statistical summaries and data visualizations to highlight patterns and anomalies in the data. Most notably, I discovered:

- Property type/size vs. error (with a combination of boxplots and scatterplots to visualize the distribution and how these variables relate to absolute log error)

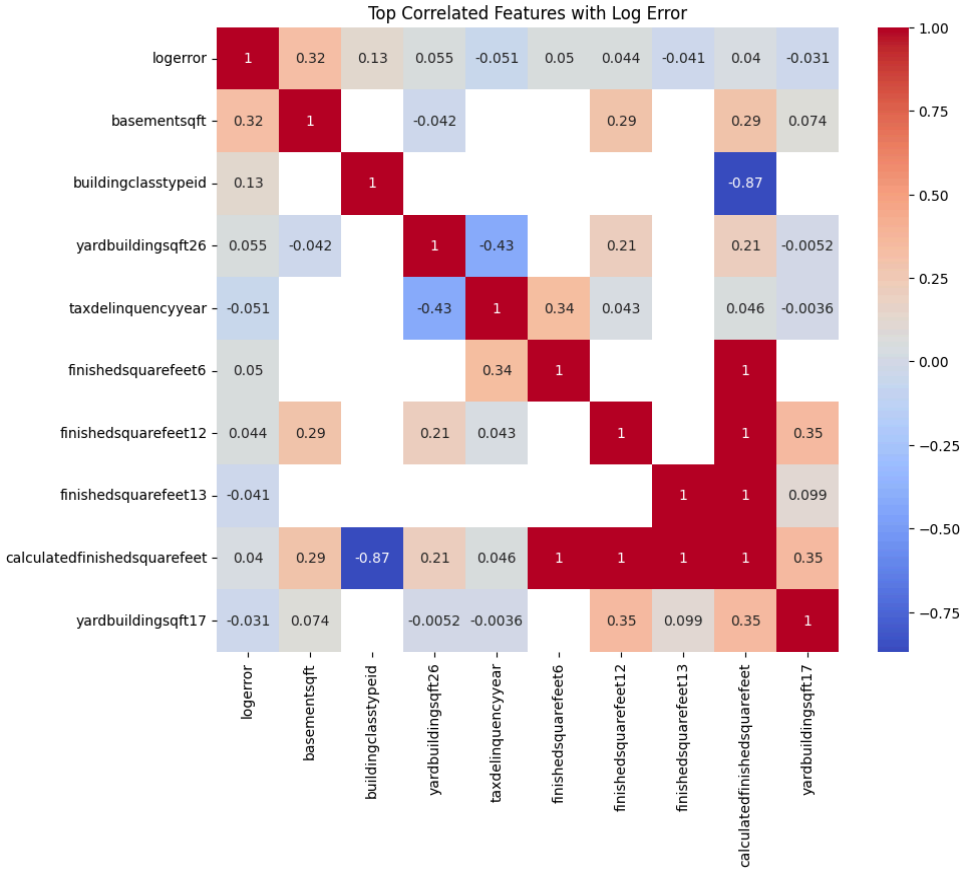
- Impact of age of homes and unusual bedroom/bathroom counts on error in prediction (regression and box plots to identify if there is a significant relationship)

- Zip codes and regions with high error (using mostly bar plots and scatter plots)



The plot above shows the relationship between the number of listings for a given zip code, and the mean log error for that zip code. My goal in plotting this relationship was to see if zip codes with fewer properties were more prone to error than those with many properties, and as you can see there is a slight downward trend that suggests this pattern. Furthermore, the outliers with much larger average errors are zip codes with relatively few listings which reinforces this idea.

One of the most important things I identified in my EDA was the possibility of multicollinearity, or dependencies between variables. If home attributes have strong correlations with each other, then their codependency can lead to unnecessary noise in predictive modeling, so I used a correlation matrix to observe whether or not these patterns are present in the data.



As seen above, each variable relationship is represented by a correlation coefficient which corresponds to a shade of red (positive) or blue (negative). Darker colors represent stronger correlations, which indicates a dependency between attributes. It is ideal to predict using relatively few, independent variables, so this visualization helped me omit redundancies. For example, I only need the fully calculated square footage, not all the various types of square footage measurements that are clearly not independent variables.

Methodology

Since I had already combined the 2016 and 2017 transaction data with their respective property characteristics, my merged dataset served as the foundation for both exploratory analysis and predictive modeling. In order to build, train, and test my model, I used the standard random split into a training set (80%) and a test set (20%) using train_test_split from scikit-learn. This ensures the model is evaluated on unseen data while maintaining a representative distribution of the target variable, which in this case is logerror.

For my initial model, I wanted to utilize a subset of the attributes that are most significant. Based on insights from EDA, this is the subset of features that are most strongly correlated with log error while avoiding redundancy/multicollinearity:

Numerical: bedroomcnt, bathroomcnt, calculatedfinishedsquarefeet, taxvaluedollarcnt, lotsizesquarefeet, yearbuilt

Categorical: regionidzip, propertylandusetypeid

(Missing values were dropped, and categorical features were one-hot encoded)

Initial Modeling Approach:

Based on my knowledge of different predictive models, I decided to start by training a random forest regressor. Some of the advantages of this model that I considered include:

- Ability to handle non-linear relationships and interactions
    - From the EDA, not all the variables seem to have a linear relationship with the response variable (log error). Random forest allows for more complexity, and avoids oversimplification.

- Resistance to overfitting with proper tuning
    - While I eliminated a majority of the insignificant variables through hypothesis testing and EDA, the multivariate relationship at hand can be prone to overfitting. Random forest helps to minimize this risk by utilizing many decision trees and bootstrapping the data.

- Providing interpretable feature importances
    - Random forest analysis inherently provides information about which features are most important in the prediction. By tracking how much each feature improves the model's performance, and providing a ranked list based on internal metrics, random forest results would be meaningful to my curiosity about the causes for Zestimate error.

Evaluation Metric:

I fit each model using the training data, and then ran predictions on the test set using MSE (Mean Squared Error) as the main evaluation metric. I deemed that MSE is appropriate here because it penalizes large errors more heavily, and our target variable is log error so MSLE (Mean Squared Log Error) is not really necessary.

Model Iterations:

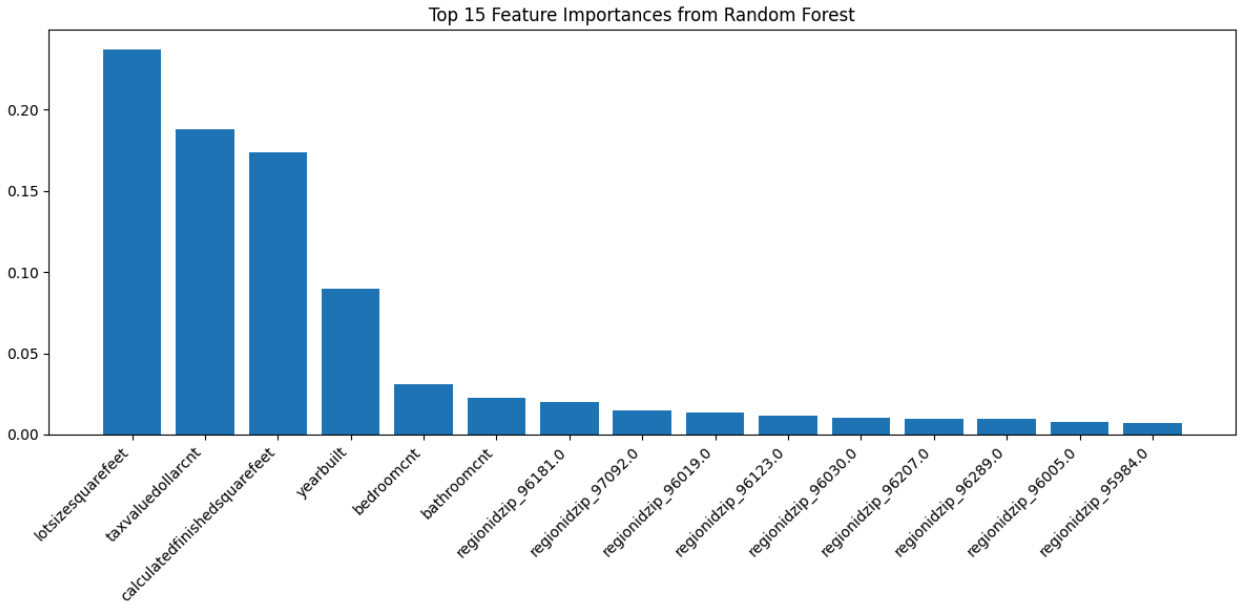
Initial Model: Random forest with the following baseline features.

- Bedroom count
- Bathroom count

- Square footage
- Tax value
- Year built (age)
- Lot size (sqft)
- Property land type

First Model - Test MSE: **0.026056**

To identify which features contributed most, I plotted the top 15 features from the initial model:



The most predictive features were:

- Square footage
- Tax value
- Lot size (sqft)
- Year built (age)

First Revision: Certain ZIP codes had strong predictive power, but not all of them, highlighting regional differences in valuation accuracy. This led me to believe that the zip code categorical feature might be adding too much noise/variance, so I fit a new random forest model but omitted the zip code variable.

Second Model - Test MSE without Zip Codes Feature: **0.026972**

Second Revision: The test MSE without including zip codes was actually higher, so this suggests that zipcodes are actually of significance in determining the amount of error in the Zestimate. However, if only certain zip codes are significant, the others might still be cluttering the accuracy of the model. Thus, I fit a third random forest model using only the top 20 features (only includes most influential zip codes and omits others). Seeing that the MSE fluctuated, I changed the number of features until I landed on the top 8 most significant features which resulted in the lowest MSE of my trials thus far. This top 8 included the important features from before, plus a couple of the most influential zip codes.

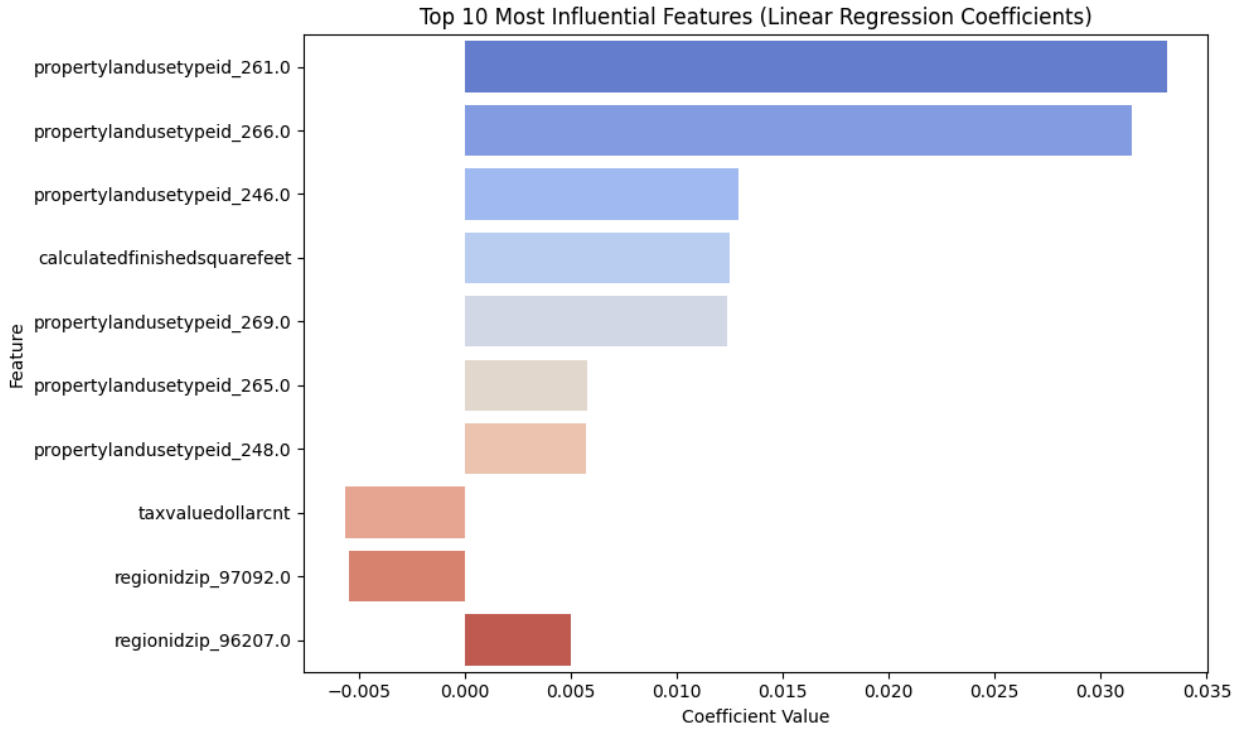
Third Model - Test MSE with Top 8 Features: **0.025806**

Third Revision: In order to assess the quality of my MSE, I wanted to compare it with a simple linear regression model. I reverted back to my original set of features, fitting a linear regression model with the same train/test data split. To my surprise, the MSE for this model was slightly lower than that of my best random forest model. Despite having assumed that this data would require more complexity, linear regression was actually very suitable.

Fourth Model - Test MSE with Linear Regression of All Features: **0.025751**

Fourth Revision: After realizing that linear regression was sufficient, I decided to follow the direction of simplicity to see if I could further improve my model and lower the MSE. I collected the top 20 most important features in my regression model, and noticed that bedroom and bathroom count were not significant, so I fit a new model without them. The MSE barely changed (improved by 0.000001), which proved that these variables were indeed not significant with this choice of model.

Fifth Model - Test MSE without Bedroom or Bathroom counts: **0.02575**



Above is a bar plot of the top 10 most influential features in the final linear regression model. One surprising observation is that tax value seems to have a negative relationship with error. This could be interpreted as properties with higher tax values having lower potential error in Zestimate evaluations. This might imply that properties assessed to have high tax values are easier to predict based on their government tax evaluations, which is a relationship I did not anticipate in the data.

Results

Model Performance:

The fifth and final model produced the lowest MSE of 0.02575. While it was surprising that the linear regression model performed better than the random forest model, it also suggests a few things about the data and the error trends of the Zestimate algorithm. First of all, the log error of the Zestimate algorithm has a relatively strong positive correlation with variables like square footage, which indicates that it needs improvement in terms of accuracy for larger homes especially. Linear regression might have also been ideal because it is less prone to overfitting, and with categorical variables like zip code, random forest might have overfit the training data. Furthermore, not all zip codes were equally frequent in the data, and data sparsity can lead to unreliable splits when using a random forest model. These are some of the reasons that random forest underperformed, and why linear regression was ultimately a better fit.

While I was able to achieve lower MSEs with slight model adjustments, a final MSE of 0.02575 still leaves room for improvement. If I were to continue with this project, I would consider incorporating non-linear transformations, or even integrating additional economic data from sources like Yahoo finance. This could open up room for time series analysis (like ARIMA modeling) to see how the real estate market changes from 2016-2017 (the years in the data) impacted the levels in error of the Zestimate. Moreover, I did not experiment with fitting an XGBoost model, which could be combined with linear models to yield even better performance. A deeper analysis of prediction errors by ZIP code could also uncover more systematic biases in the model, but this would require more intricate feature tuning.

Conclusion:

My analysis overall produced a lot of valuable insight, some of which reaffirmed my hypotheses while other takeaways were unexpected and surprising. For example, I expected that square footage would play a role in Zestimate error because anomalies in size are harder to predict. However, for the random forest model's feature importance, I did not expect lot size to be a bigger factor in the error than the calculated finished square footage (area of living space). In hindsight, this might make sense for California properties because land varies so drastically in value for different areas within the state, while the house infrastructure itself carries similar value in different places. I was also surprised that bedroom and bathroom count were insignificant in the linear regression modeling for error. I thought properties with an unusual number of rooms would lead to more error in prediction, but perhaps room count is more consistently formulaic because it is also part of the house infrastructure.

In conclusion, the main implication of my results is that property attributes with greater variance across different regions propose greater challenges in prediction, and consequently contribute most to Zestimate errors. This might suggest that the Zestimate algorithm needs to be more finely tuned for each specific region, and consider different variables with different weights according to where the property resides. For buyers, my results indicate that Zestimate evaluations are more reliable in areas with more listed properties (more data points). Perhaps buyers looking at homes in unpopulated areas, or with unique features like large lot size, should consider cross-validating the Zestimate using other sources or professional appraisers who specialize in that type of property.

Sources:

https://www.kaggle.com/competitions/zillow-prize-1

https://www.geeksforgeeks.org/feature-importance-with-random-forests/

https://medium.com/@amit25173/linear-regression-vs-random-forest-7288522be3aa

https://www.zillow.com/z/zestimate/

https://www.businessinsider.com/is-my-zestimate-accurate-home-prices-obsession-zillow-algorithm-homeowner-2024-12

https://codesignal.com/learn/courses/regression-models-for-prediction/lessons/evaluating-model-accuracy-mse-rmse-and-mae-in-regression-analysis

https://lao.ca.gov/reports/2012/tax/property-tax-primer-112912.aspx

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

https://www.geeksforgeeks.org/python-linear-regression-using-sklearn/