Machine Learning Engineer Nanodegree - Capstone Proposal

Zillow Home Value Prediction

Domain Background

Buying a home is often the largest and the most important investment/purchase people make during their lifetime. People usually spend hundreds of thousands and sometimes millions of dollars to buy a house or an apartment. When it comes to paying for the house, one needs to make sure that a fair sale price is placed on the property, i.e. the price of the house is not artificially inflated or shrank.

Accurate estimation of home prices is a challenging task because the number of factors that play a role in defining the value of a property is endless. These factors include neighborhood quality, size, tax, interest rate, and the overall economy. In the past, people mostly relied on real estate agents to value their property. This could lead to large mismatches between the actual value of a property and the proposed sale prices. Since the majority of the data required for home value estimation is publicly available or can be obtained at low costs, computers and particularly machine learning algorithms can exploit those data to automate the valuation process. Zillow's Zestimate is a well-known framework for home value prediction using machine learning models. Zestimate was established more than a decade ago. Zestimate has since largely influenced the U.S. real estate market by predicting the price of millions of properties. Despite a tremendous success in home value prediction (with a median margin of error of 5% today [1]) , there is still a lot of room for improvements (see ref. 2 to read about a recent lawsuit against Zillow). I personally think availability of a reliable system for home value prediction will greatly help home buyers and particularly first-time buyers get the value they deserve for their properties.

Problem Statement

Zillow's Home Value Prediction competition is a two-round competition in which the ultimate goal is to improve the home-price prediction algorithm of the Zillow Company (aka Zestimate). During the first round of the competition, the objective is to predict the Zestimate's residual error:

$$logerror = log(Zestimate) - log(SalePrice)$$

This means that we need to predict where Zestimate fails and where it succeeds. To train a successful predictive model, our algorithm must be as good as Zillows' algorithm (not better and not worse). In the second stage, however, the objective is to actually improve the home value prediction algorithm. This project will focus on the first stage; my goal is to make a predictive model for Zestimate's residual error that returns an average mean-absolute-error of 0.07 or smaller.

Datasets and Inputs

In the Zillow Home Value competition, we are provided with information on ~3M properties located in three counties in California namely Los Angles, Orange, and Ventura (properties_2016.csv). Among these ~3M properties, we are provided with the residual error information of ~90K properties (train_2016_v2.csv). These files can be obtained from the Kaggle website (https://www.kaggle.com/c/zillow-prize-1/data). The data set "properties_2016" consists of 58 features and 2,985,217 observations. The data set "train_2016_v2.csv" contains 3 variables and 9,0275 observations. The features include information about the size, neighborhood, tax, and location of the properties. The train data has all the transitions before October 15, 2016, as well as some transactions after October 15, 2016. The goal here is to use the features of individual properties to predict the residual error of Zestimate.

To train a model and make predictions, one needs to left-merge the "train_2016_v2.csv" and "properties_2016.csv" data to create a dataset that include both features and the labels. This would create a stand-alone data set that can be used to train and validate different algorithms.

Solution Statement

This project is a supervised regression problem. Our goal here is to predict the residual error of Zestimate given available features and labels for individual data points. To solve this problem, one can use a regression algorithm such as Decision Tree Regressor, Random Forest Regressor, or Gradient Boosting Regressor among others. These algorithms try to best describe the actual value of the residual error given features of a data point. Once these models are trained, they can be used to predict the Zestimate's residual error of any given properties.

Benchmark Model

This project, by definition, is about minimizing the difference between predictions of our new model and the existing Zestimate model. Zestimate itself is our benchmark model here. Accordingly, a small score would imply that our model compares well with our benchmark model and a large score would imply otherwise.

Evaluation Metrics

The evaluation metric that will be used in this project include mean-absolute-error. The mean-absolute-error corresponds to the expected value of the absolute error and is computed using the following formula [3]:

$$MAE(x^{true}, \hat{x}^{pred}) = \frac{1}{N_{samples}} \sum_{i=0}^{N_{samples}-1} \left| x_i^{true} - \hat{x}_i^{pred} \right|$$

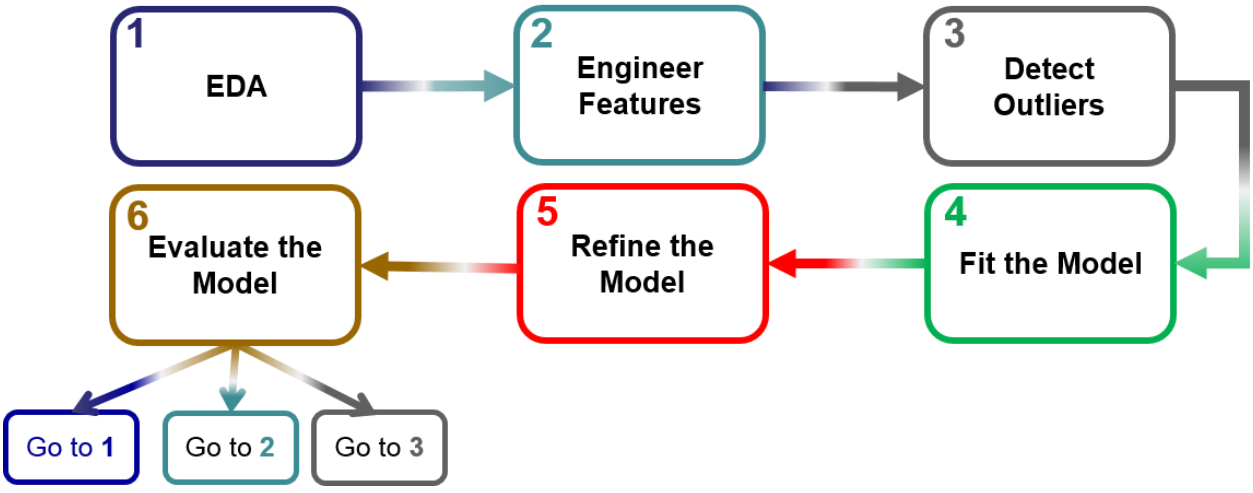
Here, $N_{samples}$ is the number of samples, x_i^{true} is the true value of the i-th data, and $\hat{x}_i^{pred}$ is the predicted value of the i-th data.

Project Design

To approach this problem, the first and foremost task would be conducting an exploratory data analysis (EDA). I would begin with a univariate data analysis and plot some basic histograms to observe distribution of various features and the label. Next, I will perform a bivariate analysis to discover possible correlations between different features (I will generate a correlation plot along with a correlation matrix). Last, I will perform a multivariate analysis, to mainly discover possible correlations between the coordinates (latitude and longitude) of the properties and the logerror. Based on the outcome of my analysis, I may create new feature using available features.

To develop an outstanding predictive model in this project, it is crucial to properly recognize and eliminate outliers. I will initially use Tukey's Method for identfying outliers and apply further refinements of the limits if necessary. I will remove those outliers before fitting any model.

In the next step, I will split my dataset into train/test sets using KFold cross-validator. I will mainly consider ensemble methods such as Random Forest Regressor, XGBoost, and other similar methods. Before, training the model, I will use GridSearch to optimize the hyperparameters of the methods. Once training of a model is complete, I evaluate the model using the test data (and possibly using the Kaggle leaderboard). Depending on the output of the model, I will go back to previous steps such as EDA, feature engineering, and outlier detection to improve the score of the model. This will be an iterative process. I will stop the iterations once the target score of 0.07 or less is obtained. The following graph shows the proposed workflow for this project:



References

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

2. https://www.washingtonpost.com/realestate/zillow-faces-lawsuit-over-zestimate-tool-that-calculates-a-houses-worth/2017/05/09/b22d0318-3410-11e7-b4ee-434b6d506b37_story.html?utm_term=.17a115dcfad1

3. http://scikit-learn.org/stable/modules/model_evaluation.html#mean-absolute-error