Capstone Proposal

Zillow's Home Value Prediction Model

Domain Background:

Houses are valued based on a number of criteria. The most common of these are Location, area, year built in, features etc. These are also the more factual predictors of the value of a house. These criteria however are not sufficient to predict the exact price. Most families buy a house to live in. They therefore do not necessarily make a decision based on some mathematical calculations. Some nuances about the house also cannot be captured in a factual manner. Things like how well the house was maintained by the previous owner, how well is the house presented during sale, cannot be completely captured in a table format.

Problem statement:

In this project I would be required to predict the log error between zestimate (Zillow's predicted home value) and the actual sales price for fall of 2017.

The log error is defined as:

logerror=log(Zestimate)−log(SalePrice)

We need to predict this log error accurately. Accuracy will be measured by Mean Absolute Error. We are provided with a training set of over 90,000 houses, with their properties and logerror mentioned.

The data has 56 properties. Properties are like Longitude, Latitude, Finished Square Feet, Number of bedrooms, Number of Bathrooms, Has Airconditioning, Tax evaluation, Built in year etc.

In order to estimate the log error, we would need to try and tune various regression models to gain least possible Mean Absolute Error.

Reason for choosing tis problem:

I have two reasons for choosing this problem:

1. It will help me gain more insight into various regression techniques. As this is a problem that many have worked on already, I would be faced with the challenge of finding a better and better accuracy by using different regression techniques.
2. It will help me understand how the value of a house is calculated factually, not the perceived value. As at this stage in my life, I am looking at prospective houses, this project would definitely provide some interesting insights.

Data Source

The data for this project has been provided by Zillow itself on Kaggle.com. It comprises of the log errors and house properties of over 90,000 houses. The data can be found at https://www.kaggle.com/c/zillow-prize-1/data.

Issues with Data

1. Large number of values are missing from many properties, like Has Air-conditioning, Has Fireplace etc.
2. A large number of properties are not properly defined like Building Type ID, Tax Delinquency Flag
3. Many properties are duplicates. For example there are three tax columns, land_Tax_Value, Building_Tax_Value and Total_Tax_Value. This way one column is just the sum of the other two columns and can be ignored. Also once we have latitude and longitude, properties like city id, neighborhood id do not need to be specified.

Solution statement

For preprocessing the data, I will start with identifying duplicate columns and unexplained columns and removing NANs. Then I would like to scale them using the standard scaler so all properties can have equal value at the beginning of the model.

For this project I will try and use various regression models. The ones I am planning to use ADA Boost, Random Forrest, Extra Trees, Gradient Boosting , Lasso, Elastic Net etc. I will try each of these models combined with K-Fold Cross Validation. I will use the validation scores to make a decision as to which model works best.

Benchmark Model

The aspirational target will be to achieve a Mean Absolute Error of 0.0642. This is the best achieved so far by any participant. A more realistic goal will be to achieve a Mean absolute error less than of 0.07.

Evaluation Metrics

The evaluation metrics will be Mean absolute error between calculated and actual log errors.

Project Design

The project will be divided into 8 steps:

1. Download data and load into Pandas Database. Explore Data. Clean the data and handle errors like NaN or blanks, Use encodings where required.
2. Define correlation between various house properties. Reduce redundancy if required by a dimensionality reduction method like PCA
3. Divide data into Train Test Split.
4. Try different regression models and compare their results. Compare test and train accuracies.
5. Optimize each regression model, by using different hyper parameters
6. Finalize the model, or an ensemble of models
7. Use validation methods like K-Fold Cross Validation
8. Present the result