Capstone Project

Zillow's Home Value Prediction Model

Project OverView:

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.

In this project we started with data provided by Zillow.com on housing market and have used various regression models to pedict the log error between the Zillow proce prediction(zestimate) and the actual selling price of the house.

We then cleaned the data by removing houses for which we did not have actual sales value (meaning no log-error, we also removed redundant or duplicate columns and replaced NAN values with another more logical one.

We concluded the project by applying various regression techniques and trying to improve the results via Hyper-Parameter tuning and using techniques of ensemble learning. The best mean absolute error we achieved was 0.6766.

Problem statement:

This project is targeted towards helping Zillow predict the logerror 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.

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.

Evaluation Metrics

The evaluation metrics will be Mean absolute error between calculated and actual log errors. Mean absolute error measures the average magnitude of errors in a set of predictions. It does not consider the direction of the errors, just the magnitude. The formula is

$$MAE=1/n \sum |y - y^|$$

This method ensures that all errors weather large or small are given equal weight. As against Root Mean Squared Error. As the aim of this project is to minimize overall error and not just large errors, Mean Absolute Error is the metric chosen.

Data Exploration:

The Zillow data contains data of 90,274 houses that have made a sale in 2016. The id used for this data is 'parcelid' and we have the y value, i.e. log error for all these houses

We have properties data x value of 2,985,217 houses. These cover all the houses that were sold (90,274) and more which have been listed pending sale. The x value or house properties table consists of 58 columns with details like

S. No.	Property Name	Description
1	airconditioningtypeid	Not Sufficiently described
2	architecturalstyletypeid	Not Sufficiently described
3	basementsqft	We are using finished square feet and Has basement as an indicator
4	bathroomcnt	Number of Bathrooms. NAN replaced with Median.
5	bedroomcnt	Number of Bedrooms
6	buildingclasstypeid	Not Sufficiently described. NAN replaced with Median
7	buildingqualitytypeid	We have entered 0 for NAN and retained other numbers
8	calculatedbathnbr	There is seprate details on Number of bbedrooms and bathrooms
9	decktypeid	Not Sufficiently described
10	finishedfloor1squarefeet	The finished square feet columns are merged into one and the final column is used for our calculations

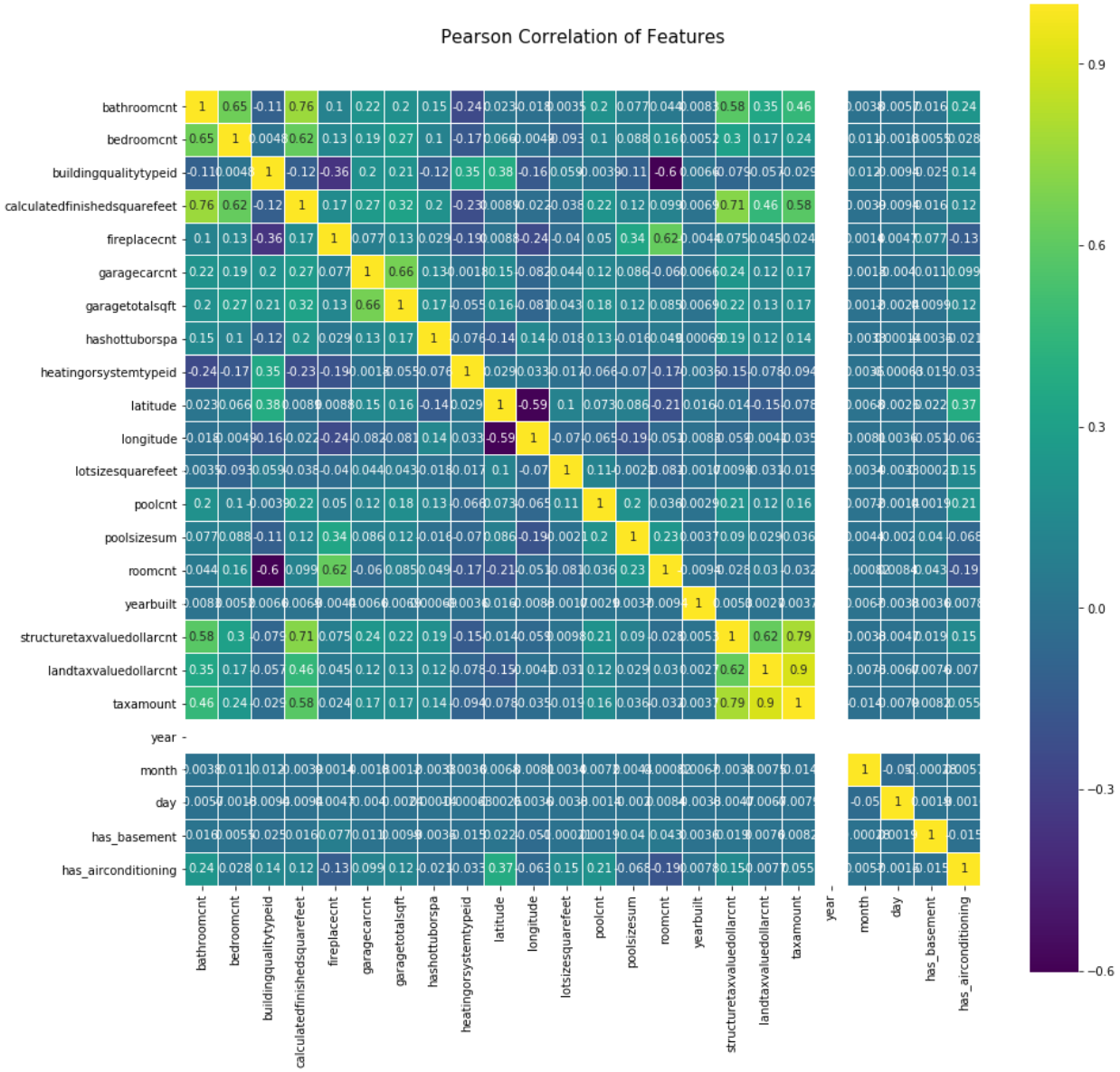
11	calculatedfinishedsquarefeet	Total square footage of finished housing
12	finishedsquarefeet12	The finished square feet columns are merged into one and the final column is used for our calculations
13	finishedsquarefeet13	The finished square feet columns are merged into one and the final column is used for our calculations
14	finishedsquarefeet15	The finished square feet columns are merged into one and the final column is used for our calculations
15	finishedsquarefeet50	The finished square feet columns are merged into one and the final column is used for our calculations
16	finishedsquarefeet6	The finished square feet columns are merged into one and the final column is used for our calculations
17	fips	Not Sufficiently described
18	fireplacecnt	Number of fireplaces. Replaced 0 in place of NAN
19	fullbathcnt	Covered in the number of Bathrooms column
20	garagecarcnt	Count of parking space for cars
21	garagetotalsqft	Area of garage
22	hashottuborspa	whether It has hot tub or spa
23	heatingorsystemtypeid	ype of heating or system. Replaced 0 for NAN.
24	latitude	Latutude
25	longitude	Longitude
26	lotsizesquarefeet	Total area of the lot
27	poolcnt	Number of pools
28	poolsizesum	Size of pool
29	pooltypeid10	Not Sufficiently described
30	pooltypeid2	Not Sufficiently described
31	pooltypeid7	Not Sufficiently described
32	propertycountylandusecode	Using Latitude and Longitude information
33	propertylandusetypeid	Using Latitude and Longitude information
34	propertyzoningdesc	Using Latitude and Longitude information
35	rawcensustractandblock	Using Latitude and Longitude information
36	regionidcity	Using Latitude and Longitude information
37	regionidcounty	Using Latitude and Longitude information
38	regionidneighborhood	Using Latitude and Longitude information
39	regionidzip	Using Latitude and Longitude information
40	roomcnt	Number of rooms
41	storytypeid	Not Sufficiently described
42	threequarterbathnbr	Covered in Number of bathrooms
43	typeconstructiontypeid	Not Sufficiently described
44	unitcnt	Not Sufficiently described
45	yardbuildingsqft17	Not Sufficiently described
46	yardbuildingsqft26	Not Sufficiently described

47	yearbuilt	Built in Year
48	numberofstories	Not Sufficiently described
49	fireplaceflag	Covered in Has Fireplace
50	structuretaxvaluedollarcnt	Tax value of the building
51	taxvaluedollarcnt	Covered in Land Tax value and structural tax value
52	assessmentyear	This is the year of assessing the properties
53	landtaxvaluedollarcnt	Tax Value of the land
54	taxamount	Totak Tax Amount
55	taxdelinquencyflag	Not Sufficiently described
56	taxdelinquencyyear	Not Sufficiently described
57	censustractandblock	Not Sufficiently described
58	transactiondate	This is divided into year, day and month and then used.

There were various tasks required for cleaning the data. We performed them all, as mentioned below.

1. Large number of values are missing from many properties, like Has Air-conditioning, Has Fireplace etc. We replace dthe NAN values of relevant featured with withwe 0 or Median.
2. A large number of properties are not properly defined like Building Type ID, Tax Delinquency Flag. We removed these from our relevant Properties.
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. We removed these from our relevant Properties.

Data Visualization



As we can see in this visualization, most features are correlated to some extent. Some correlations that jump out to me are:

1. Structure Tax Amount, Land Tax Amount and Tax Amount seem to be highly correlated
2. Calculated finished square feet is also directly correlated with bedroom count and bathroom count
3. Bedroom count and bathroom count are also highly correlated.
4. Has air conditioning is directly correlated with latitude as expected

This means that though the features add extra information, the dimensionality of the data can be reduced using a technique like PCA without losing the variance in the data.

Algorithms and Techniques

For this project I intend to use various regression models.

I will first try the regression model and then based on the result try ensemble learning methods for the same.

In the end I will evaluate feature importance's to try and understand which properties of a house are most relevant for its log-error in Zillow.

The models I am planning to use are: Decision Tree Regressor, Lasso, Elastic Net, Random Forresr, Gradient Boosting, ADA Boosting. I will use validation scores for Mean Squared error to understand which model works best.

Decision Tree Regressor: Decision tree regressor tries to split the data based on a particular feature so that the splitting maximizes information gain. It is like 20 questions, the first question tries to divide the sample set into two equal parts so as to narrow down the possible answers as much as possible. This splitting of data continues till we have a distinct answer or till we reach a point that the tree is too long. A decision tree regressor is a good choice because most of the features of a house are discrete and splitting data based on discrete features can yield fast yet reliable results.

Lasso: **(**least absolute shrinkage and selection operator) Lasso is a method of regularizing the weights of independent variables while performing regression. This method is good for the Zillow price prediction because the independent variables available to us are actually correlated and therefore restricting the variable coefficients would help us avoid undue weight being assigned to the wrong parameters.

Elastic Net: This regularization combines the benefits of both lasso and ridge regression in penalizing large coefficients to independent variables. I used this model here to see if we can achieve a better result than lasso by using another method to restrict larger coefficients.

Random Forrest: This is an ensemble method of decision trees. In this training data is divided into parts and each part is used to create a different decision tree. Now while predicting, each tree is used to make predictions and the final answer is achieved by counting the votes of all the trees

Gradient Boosting: This is a ensemble learning method built on top of decision trees. Gradient boosting creates a weak predictor, understands its shortcomings, boosts the next predictor so that it tries to overcome these shortcomings.

ADA Boosting: This is another boosting method. Here many weak classifiers are created and their prediction is weighted and added to come up with the final answer. I am using ADA Boosting on top of decision tree regressor in this projet.

Benchmark Model

I have benchmarked my model against Kaggle leaderboard. 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.

Data Preprocessing

Step 1. Merge the data between the House properties and log error. Only retain the houses that have a log error value.

Step 2. Remove Duplicate Columns

Step 3. Remove columns which are not explained

Step 4. Calculate relevant values from multiple columns, for example calculate the final finished square feet from all the different columns by using the max function

Step 5. Remove NAN values by replacing them with either 0 or relevant median

Step 6. View the data to see if any more editing needs to be done

Step 7. Divide the merged table into properties and labels

Step 8. Scale the properties table with standard scalar

Step 9. Split the data into train and test set

Implementation

For implementation, I imported relevant libraries from sklearn I then fit the model on the train data, use it to predict test and evaluated my results based on Mean Absolute Error.

The main challenge faced while working on this data was that ensemble learning methods when combined with cross validation required a very large run time. For this I reduced the dimensions by using PCA, this provided good results but also increased our mean absolute error slightly.

The models used are:

1. Decision Tree Regressor
2. Lasso Regression
3. Elastic Net Regression
4. Extra Trees Regressor
5. Random Forrest

6. ADA Boosting on Decision Trees
7. Gradient Boosting on Decision Trees
8. Dimensionality Reduction using PCA
9. K-Fold Cross Validation
10. Feature_importances parameter of the best working algorithm

Refinement

For refinement I started with the basic models like Random Forrests, Lasso and elastic net. I would then change hyper parameters and try, followed by ensemble learning like Boosting methods and Random forests. The process was based on trial and error. The details of the models with their parameters tried and results obtained are given in the results section.

Results

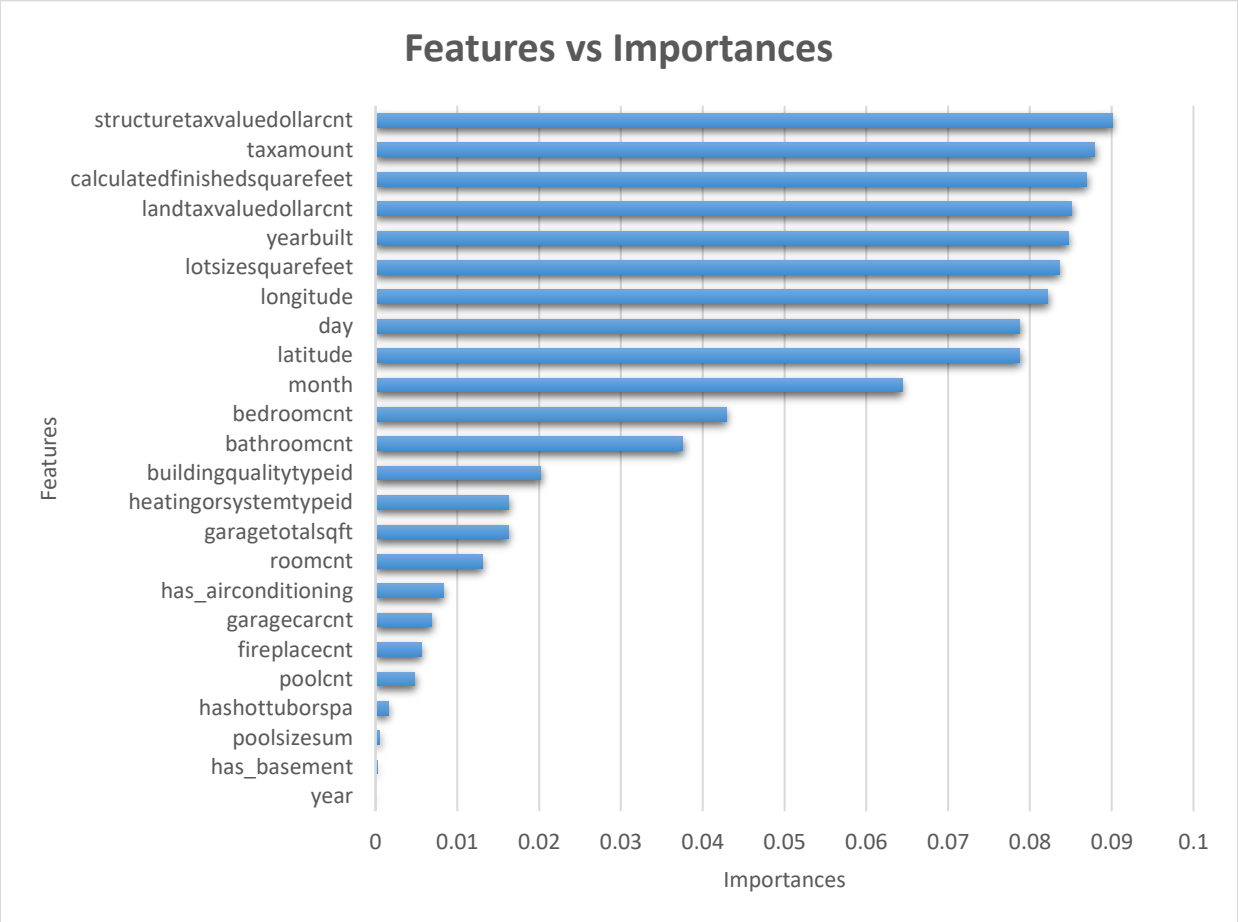
The best results I have achieved is a mean absolute error of **0.067882**. This was achieved with the ADA boosting method on Decision Tree Regressor. Below are the results obtained through our model exploration:

Model	Mean Absolute Error
Random Forrest Regressor (max_depth =10)	0.069351
Lasso (alpha=1e-6)	0.068490
Lasso (alpha=1e-7)	0.068535
Elastic Net (alpha=1e-6)	0.068467
Elastic Net (alpha=1e-7)	0.068525
ADA Boost on Decision Tree Regressor(Max_Depth=50, n_estimators=100)	0.068192
ADA Boost on Decision Tree Regressor(Max_Depth=50, n_estimators=200)	0.068022
ADA Boost on Decision Tree Regressor(Max_Depth=100, n_estimators=200)	0.067940
ADA Boost on Decision Tree Regressor(Max_Depth=100, n_estimators=300)	**0.067882**
Gradient Boosting Regressor (n_estimates=25, max_depth=25)	0.070391
Gradient Boosting Regressor (n_estimates=50, max_depth=25)	0.072321
Extra Trees Regressor (n_estimators=50)	0.080256
Elastic Net (alpha=1e-6) (after PCA)	0.068710

As shown above ADA Boost method is tried using a number of hyper parameters. The results obtained were consistently good. We have verified the robustness of the model by using K-Fold Cross Validation with 5 folds. This was done on reduced data by passing it through PCA (because of time constraints). The results obtained are ([-0.06847512, -0.06698877, -0.0691936 , -0.07008986, -0.06760575]). These results show that the model is able to predict the log error accurately and is in line with our other observations.

Conclusions

We were able to predict the log error of houses with a mean absolute error of 0.06766. This is lower than our realistic benchmark of 0.070. The most important features for predicting log error in order are shown in the graph below.



Future Improvements:

There are a number of ways we can improve on the model given above. Some of my ideas are as explained below:

1. Use Grid Search CV for achieving even better tuning. I was unable to use this technique on such a large dataset as it required a lot of time
2. Try instance based learning methods like K nearest Neighbors. These will however require a lot of time to predict and therefore not suitable is required for a real time implementation
3. Using advanced boosting methods like xgboost
4. Understand the missing features more and use them properly to improve predictions

Reflection

This project had several small pieces which fit together to give us the result. There was a lot of data cleaning and pre-processing required. The first piece was finding the houses for whom we had the log error. This reduced our dataset by more than half and increased our run time for all future operations. I was particularly interesting was that there were 57 features in the begining. Many of the features were duplicates and many others were not explained properly. I had to look at the features carefully to identify their purpose, choose features to keep and those to delete. Found ways to fill the NAN value for features I did not understand completely but believed would be essential to the final result (One such feature was 'buildingtypeid').

Another interesting part of this project was, exploring how much ensemble learning can improve a model. I tried various ensemble learning methods on Decision Tree regressor and understood the difference between Gradient Boosting and ADA Boosting.

Finally I was happy to find that using the feature importance's showed us that the structure tax value contributes the maximum to log error. This was in line with my expectation as Buying a house is an aspirational decision and aspirational decisions are not completely dependent on the hard facts like square footage and location, but on the quality of the build of house.