ZILLOW HOME VALUE PREDICTION

**Zillow:**

Zillow is an online real estate database company founded in 2006 - Wikipedia

**Zestimate:**

Zillow uses Zestimate system which uses 7.5 million statistical and machine learning models that analyze hundreds of data points on each property to come up with the price value for any given house

**Problem Statement**:

In this problem, Zillow is asking us to build a ML model on the residuals ofZestimate system described above. The target variable given in the dataset is log transformed as given below

**Log error= log (Predicted price) – log (Actual Price)**

**Metric:**

Models are evaluated on **Mean Absolute Error**

**Input data:**

* Transaction data –2016
  + The transaction data file has all the houses that were sold in 2016 across the state of California and log error made by Zillow system
* Property data (27 million)
  + The property data file has details about the properties such as Mortgage Records Attached, Tax Assessments, Physical features of the Property (Bed Count, Living Area, Age of the building), Location Details etc

**Sample Input Attributes (X):**

|  |  |  |  |
| --- | --- | --- | --- |
| **House\_Area** | **House\_Details** | **Region** | **Tax** |
| area\_basement | num\_unit | region\_county | tax\_total |
| area\_patio | num\_story | region\_city | tax\_building |
| area\_shed | num\_room | region\_zip | tax\_land |
| area\_pool | num\_bathroom |  | tax\_property |

**Project Milestones**

**Chosen approach:**

* Exploratory data analysis
* Impute missing values
* Feature Engineering
* Model Building & Fine Tuning

**Exploratory data analysis**

As part of exploratory data analysis, we tried to understand the data using various visualization techniques which is available in **the EDA jupyter notebook file attached**

**Impute Missing values**

One of the challenges in dealing with this data is, it had 35% of the attributes missing more than 90% of values. We handled imputation using 2 strategies given below

Approach 1:

* + Imputation after joining the 2 datasets (on 90k records) -Using KNN

Approach 2

* + Imputation before join operation (on 27million records) - Using KNN

The purpose of running the imputation model using second approach is to take advantage 27million records to find better neighbors to impute

**Feature Engineering**

As part of feature engineering we created new variables using variable interaction and also did feature selection using different approaches.

* **New features using variable interaction**

* + Total rooms = bath\_count + bedroom\_count
  + Average room size = total\_finished\_living\_area\_sqft/roomcnt
  + Ratio of structure tax to land tax = structure\_tax/land\_tax
  + ExtraSpace = lot\_area\_sqft - total\_finished\_living\_area\_sqft
* **Feature Selection & Dimensionality Reduction**

The below mentioned techniques are used to make feature selection and dimensionality reduction

* + Recursive feature elimination (RFE)
  + XgBoost feature importance (Information Gain & Gini Index)
  + TreeRegressor feature importance (Information Gain & Gini Index)
  + PCA (Principal Component Analysis)
* **Model Building & Fine Tuning**
  + All the ML algorithms used in this project are tuned with gridsearchcv using scikit-learn algorithms
  + The whole data is divided into train 90% and test 10%, models are built on train data using cross-validation techniques

**Results:**

**Kaggle Benchmark : 0.064**

|  |  |  |
| --- | --- | --- |
| **Imputation on transaction data** | **Imputation on Properties data** | **PCA** |
| **Linear Regression = 0.0672** | **Linear Regression=0.0672** | **Linear Regression = 0.0673** |
| **Ridge Regression = 0.0672** | **Ridge Regression=0.0672** | **Ridge Regression=0.0672** |
| **Random Forest =0.673** | **Random Forest =0.673** | **Random Forest=0.673** |
| **Gradient Boosting=0.0671** | **Gradient Boosting=0.0671** | **Gradient Boosting=0.0671** |
| **XgBoost =0.0670** | **XgBoost=0.0670** | **XgBoost=0.0672** |

**Inferences and future improvements:**

The main reason why even the widely used boosting algorithms couldn’t increase the error metric is:

* The whole model is built over the residuals from a powerful machine learning model
* The second reason could be because the amount of missing information in each attribute
* The other reason could the limitation we had with the hardware, where we couldn’t build an imputation model using all 27 million records
* Some of the imputation techniques that we used using mean and median, could have potentially changed the distribution of data
* One other interesting observation in linear regression model is that, the R squared value is less than 1%,

Which shows that the model couldn’t really express/explain much about the underlying data

* The future improvements could be potentially trying different imputation techniques, using xgboost’s inbuilt ability to handle missing values

**Team Members**

* Abhishek Dhameja (4868345770)
* Deepti Rajashekharaiah Siddagangappa (1716400396)
* Eswar Lakshminarayanan (7534803085)

**Division of Tasks**

* Data Exploration
* ML Modeling Equally divided among Team members
* Visualization