USER MANUAL

Application Name: Zillow Dataset on Kaggle

Abstract:

The task is to predict logerror for 201610, 201611, 201612, 201710, 201711, 201712 for the Zillow dataset.

How to set the environment and run the project:

- Install Anaconda from https://conda.io/docs/user-guide/install/windows.html
- When the anaconda prompt opens, type jupyter notebook.
- In the Home tab, click New -> Notebook.
- Paste the code from my .ipynb file in that note book.
- Press Shift+Enter to execute line by line.
- After the entire file executes, the file zillowaai5.csv will be generated in C:/Users/[‘respective-name’]
- Paste the path of properties17, train16 and train17 in the respective paths in the code.
- I have not used properties2016 file as it does not serve a useful purpose after the release of 2017 properties file.

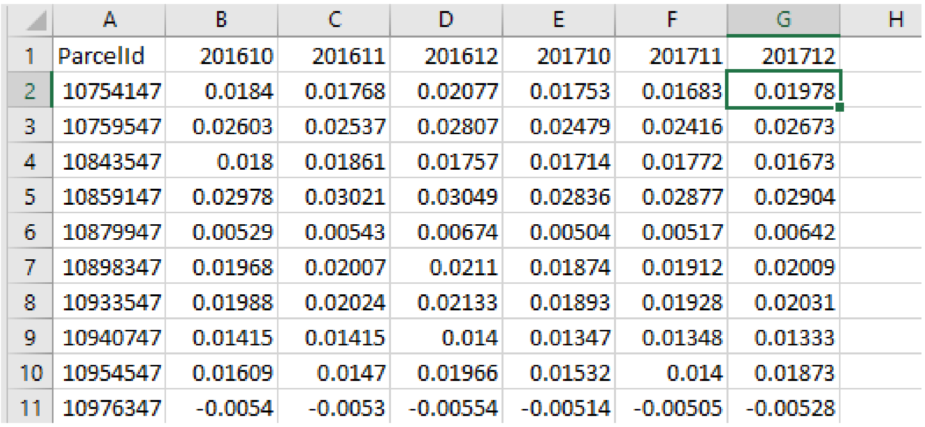
Techniques Used:

- Used pandas and sklearn to read the csv files and create the dataframes**.**
- Used CatBoostRegressor to predict the logerrors**.**
- For pre processing, I used ggplot to visually analyze the missing values in the data and to see how the features affect the target variable, logerror.
- For installing ggplot run pip install ggplot in the terminal.
- To install xgboost, cd to the path where respective whl file is stored.
- Run pip install xgboost

Result:

- The eval metric I used is mae-mean absolute error.
- mae for train data is 0.059 and mae for test data is 0.061
- On Kaggle my public score is 0.0643 and private score is 0.0752

Output sample screenshot of csv file generated:



	A	B	C	D	E	F	G	H
1	ParcelId	201610	201611	201612	201710	201711	201712	
2	10754147	0.0184	0.01768	0.02077	0.01753	0.01683	0.01978	
3	10759547	0.02603	0.02537	0.02807	0.02479	0.02416	0.02673	
4	10843547	0.018	0.01861	0.01757	0.01714	0.01772	0.01673	
5	10859147	0.02978	0.03021	0.03049	0.02836	0.02877	0.02904	
6	10879947	0.00529	0.00543	0.00674	0.00504	0.00517	0.00642	
7	10898347	0.01968	0.02007	0.0211	0.01874	0.01912	0.02009	
8	10933547	0.01988	0.02024	0.02133	0.01893	0.01928	0.02031	
9	10940747	0.01415	0.01415	0.014	0.01347	0.01348	0.01333	
10	10954547	0.01609	0.0147	0.01966	0.01532	0.014	0.01873	
11	10976347	-0.0054	-0.0053	-0.00554	-0.00514	-0.00505	-0.00528	