Capstone Proposal MLND

Diego Gallegos September 25, 2017

Domain Background

This is project is base on Kaggle competition called Zillow House Prediction. Zillow is a company that has a platform that does online real estate accompanied with very accurate home valuation services for buyers and sellers. The competition consists on improving the accuracy of the prediction algorithm indirectly by predicting the difference between the predicted log error and the actual log error. The log error is defined as Zillow's Prediction Estimate (Zestimate) and the real sales price.

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

Problem Statement

The dataset has training examples and labels for them. This implies a supervised learning problem. Ensemble methods will be used to solve the problem. Ensemble methods combine different models to obtain a more accurate and comprehensive model. First, Linear regression was considered but being a high dimensional problem then it would be complicated to obtain a model that does not overfit over the data. Meanwhile, random forests have had many successes on this kind of models in the past. One notable case is the Netflix Machine Learning Competition to build a collaborative filter. Additional to this, in case of not improving the model with the method described, stacking is considered with other algorithms to improve the leaderboard score.

Datasets and Inputs

The datasets are provided by the Kaggle Competition It consists in:
- **properties_2016.csv** - all the properties with their home features for 2016.
- **train_2016.csv** - the training set with transactions from 1/1/2016 to 12/31/2016
- **sample_submission.csv** - a sample submission file in the correct format.

Properties dataset contains all the training features of the parcels. The shape of the data is 2,985,217 x 58. There 58 columns where 57 represent the training features and one is the parcel id. Some of the most important features are:

- transaction_month: month when the transaction took place.
- latitude: Latitude of the middle of the parcel multiplied by 10e6
- longitude: Longitude of the middle of the parcel multiplied by 10e6
- calculatedfinishedsquarefeet: Calculated total finished living area of the home
- structuretaxvaluedollarcnt: The assessed value of the built structure on the parcel
- landtaxvaluedollarcnt: The assessed value of the land area of the parcel
- yearbuilt: The Year the principal residence was built
- lotsizesquarefeet: Area of the lot in square feet
- taxvaluedollarcnt : The total tax assessed value of the parcel
- regionidzip: Zip code in which the property is located
- finishedsquaredfeet12: Finished living area
- bedroomcnt: Number of bedrooms in home
- bathroomcnt: Number of bathrooms in home including fractional bathrooms
- rawcensustractandblock: Census tract and block ID combined - also contains blockgroup assignment by extension

The rest of the explanation features can be obtained on the **zillow_data_dictionary.csv** file in the datasets on the kaggle data section.

Training dataset contains 90,275 training examples:

- Parcelid: the id of the property to be referenced on the properties dataset.

- Logerror: the target variable which is the logerror between the real price and Zestimate (Zillow's prediction)
- Transactiondate: the date when the sale was made.

Solution Statement

The first step towards solving the problem is doing exploratory data analysis. I can describe
the process as follows:

1. Feature explorations: Explicitly knowing the structure of house data and sales data. Both of them are provided and should be merged. Correlation analysis is going to be done to determined what variables provide a better explanation of the prediction. Feature importance must be validated with decision tree analysis to discover which provides better explanatory power by comparing its splitting power based on entropy.
2. Data cleansing: Exploring variable distribution sparsity and analyzing different imputation techniques to see which best fits the data.
3. Feature importance: analyzing the features that most affect the variance of the
4. prediction.
5. Feature Engineering: creating new features to reduce the dimensionality of data.
6. Cross validation: to assess the predictive power of the model by creating different folds of data and rotating them between test and training data.
7. Prediction: I will be analyzing to ensemble algorithms one is xgboost (Extreme Gradient Boosting) and lightgbm (Light Gradient Boosting). This algorithms have shown great speed and accuracy for many hard and sparse regression problems. As is well known, boosting main objective is to convert weak learners into strong learns by reducing the variance and bias.
8. Assess Predictive power on training set as we have a way to compare the accuracy

Benchmark Model

The data should be tested and benchmarked again three algorithms:

- Linear Regression
- Extreme Gradient Boosting
- Light Gradient Boosting

Linear Regression is a simple method that fits a line to data. Then its prediction accuracy is assessed by taking the L2 error of the prediction and correct labels. This yields to the update of the weights. Linear Regression avoids overfitting by using regularization terms added to it. If the regularization is a L1 measure is called Lasso Regression and if its L2 measure then is called Ridge Regression.

Gradient Boosting is also used extensively to solve regression problems. It uses ensembles of weak decision tree models to produce a strong learner. It does this by optimizing a differentiable loss function. Naïve Decision trees are known to overfit when the number of levels of the tree increases. Gradient Boosting does the following:

1. Fits a model to the data
2. Then it obtains the residual, which is the subtraction of the labels and the prediction.
3. Then it fits a model to the residual.
4. Then it combines both models obtaining a better accuracy.
5. Finally, optimization is attempted using the sum of the learners by using gradient
descent.

Hyper-parameter tuning is made by cross validation. The libraries that are mentioned on the benchmark used random sampling of the columns and rows of the data to improve the amount of information learned for the regression tree. These types of trees are called Random Forests. One of the main differences between Xgboost and Lightgbm is the training times obtain by the latter. This are two very strong boosting methods that are gaining fast adoption by practioners on solving sparse and highly dimensional regression problems.

The comparison will be between different boosting methods such as lightgbm and xgboost . Also other methods will be compared, linear regression and neural networks will be ran on the model to test the accuracy. As a last benchmarking model will be a stack of all these methods using optimization to obtain the best weights for each prediction of each model.

Running some prediction on naïve algorithms with no hyperparameter tunning, the following results are shown:

1. Decision Tree Regressor
    a. Accuracy (R2) on training set: 0.9992
    b. Accuracy (R2) on test set: -1.284
    c. Mean squared error (MSE)
2. AdaBoost
    a. Accuracy on training set: -2.436
    b. Accuracy on test set: -3.655
    c. Mean squared error (MSE): 0.104744
3. Gradient Boosting Regressor
    a. Accuracy on training set: 0.5084
    b. Accuracy on test set: -0.07517
    c. MSE: 0.0242

All of these experiments are based on splitting the training dataset on a 90% - 10% ratio. Also these algorithms were chosen to reproduce the underlying ideas between xgboost and lightgbm as they use decision trees and boosting.

Evaluation Metrics

The competition uses Mean Absolute Error to verify the improvement of the submitter's solution against the best model. The mean absolute error subtract the prediction with the labels and its divided by the number of training examples.

Project Design

On the problem statement section, a little bit about this was explained. Expanding more
about this there are several tests that are planned to be the ran:

1. Removing extreme sparse rows and benckmark their importance on the final
prediction. (Kaushik)
2. Use several imputation methods and evaluate its effect on the prediction:
a. Mean Imputation
b. Median Imputation
3. Test Gradient Boosting Methods separately.
4. Explore on FWLS (Feature-Weighted Linear Stacking) to combine Xgboost and LIghtgbm results. Evaluate the effect on combining them. Other additional methods that can be stacked is Linear Regression and Neural Networks.

Bibliograph

Chen, E. (s.f.). *http://blog.echen.me/*. Obtenido de Edwin Chen Blog: http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/

CMU, R. T. (2014). High-dimensional regression - Advanced Method for Data Analysis. Pittsburgh, PA, USA.

Kaggle. (september de 2017). Obtenido de https://www.kaggle.com/c/zillow-prize-1 Kaushik, N. (s.f.). *Linkeding*. Obtenido de https://www.linkedin.com/pulse/exploring- lightgbm-naveen-kaushik/

Tianqi Chen, C. G. *Xgboost: A Scalable Tree Boosting System*. Washington, USA: Arvix.