Exploratory Data Analysis ”The Zillow Prize Challenge”

Dibyajyoti Pati

Stony Brook University

Abstract

In this investigation, a detailed exploratory analysis was made in order to understand the behavior and response of target variable, log error with respect to the provided input parameters.The machine learning pipeline comprised of steps involving analysis of data types of the input data, its distribution over the training sample and entirety ,understanding the Pearson correlation of these parameters, cleaning of data: data type conversion, data imputation, & introduction of new features, Creation of a baseline model using the sklearn linear regression model using the top six parameters obtained by performing Recursive Feature Elimination method over the available numerical parameters only ,followed by creation of model using Random Forest Regressor using the same set of numerical parameters . This alone reported over 480% improvement over the baseline linear regression when evaluated on the validation set. The Random Forest model was augmented with few additionally categorical variables (categorical variables were converted to Binary using One Hot Encoding) which further reported over 100% improvement on the previous random forest model

Keywords: Linear Regression, data imputation, Recursive feature elimination, Pearson correlation, Random Forest regressor, One Hot Encoding

Feature Selection & Cleaning:

The data provided in "properties_2016.csv" was loaded into a pandas data frame, followed by segregation of feature types i.e. Continuous numeric, Categorical & String.The dataset consisted of 58 features comprising of Floats, integers, strings & categorical variables.A subset of features was created comprising of only numerical features from the understanding of the data provided in the data dictionary. This subset of data was subsequently used to understand the correlation amongst the features. It was observed that there was a strong correlation among features relating to finished square footage of the area, various parameters in taxation i.e. tax assessed value, computed property tax. So, in order to preserve the variance and avoid usage of multiple correlated features, three features were created from the above set which was later used for training purposes. The missing data in the numerical features were imputed with the mean value of the same feature

Random Forest Regressor:

A Baseline model was created with six top-ranked features from the list of numerical features as reported from the recursive feature elimination. This model was used as the baseline for comparing all other subsequent models. There are two metrics used in the analysis 1.The R squared(r2) value and the score reported upon submission by the kaggle scoring algorithm. The baseline r2 value was determined as 0.001141 & Zillow score of 0.0650457. This score is the best of 30 epochs trained on linear models. The low value of the r2 value from the linear model led to an exploration of models suited for understanding non-linear relations.

Random Forest Regressor is based on the decision trees but uses a bootstrapping method for both for selection of Sample and Features in each iteration. The idea behind using bootstrapping(selecting random samples with replacement) samples on different trees in the forest are to prevent them from being correlated to each other.The second step involves bagging

of features to be used for training by the trees. The outcome in case of a regressor is the average of all trees. This method of bootstrapping essentially reduces the variance of the models (overfitting) and at the same time preventing the bias from shooting up(underfitting). In the model, a set of control parameters were carefully chosen (by the method of brute force rather than cross validation) to prevent the forest from creating a highly overfitted model. Mostly the number of estimators, minimum number of leaves and number of features to be retained were used for tuning the model.

Model Evaluation:

The initial model built over the same features as of the baseline linear regression model performed considerably well over the validation set (which is 30% split from the training sample) reporting a whopping 4 times better over the baseline r2 from the linear model. The first model had a max depth of 10 for the trees with a number of trees set to 20 and all the parameters being set to default. The r2 value from the model was 0.004998 but this model did not perform well when used to predict the errors for the rest of the records. This was evident from the fact that the kaggle score dropped from the baseline value of 0.06504 to 0.0769434. This led to the exploration of inclusion of categorical variables to improve the score of the models and four new features were introduced on top of the best 5 numerical features. The second model comprised of building quality, property type id in the feature list. Further, this model was tuned by setting the max depth as 20, number of trees as 30 whereas dropping the total features to be retained to 60% of the original input space. This method led to r2 score improvement of 153 % from the previous Random forest and around 10 times better than the baseline model. This also showed improvement in the kaggle scoring algorithm with score of 0.0661506

Interesting experiences:

It can be inferred from the above data that a general linear regression model built over fewer but important features would outperform a more sophisticated learning algorithm. Addition more features led to a poorer model performance for both linear and random forest. And even though the r2 error showed improvement from the baseline, the random forest models could not beat the performance of linear models when asked to predict on the unknown data set. It is also to be noted that the loss function used for optimizing the cost functions in both linear regression and random forest were based on mean squared errors, whereas the kaggle scoring algorithm would be using a completely different metric (and maybe loss function) for evaluating the performance of the submitted scores. The model performances can be improved on working on the following items:

1. Tuning of hyperparameters
2. The sampling of data for training
3. Selecting an appropriate metric for comparing models instead of r2 (or mean squared error)