Machine Learning Engineer Nanodegree

Capstone Proposal

Pingping Chen
April 15th, 2018

Domain Background

This is one of completed competitions on Kaggle hosted by Zillow. Zillow has a system called Zillow's Zestimate home valuation. It would estimate home values based on million statistical and machine learning models. But Zillow wanted to improve the accuracy of the Zestimate. Since the real estate industry is growing very fast in Bay Area. And the machine learning model may differ than years ago. My job is also about using machine learning algorithms to predict continuous variable, so I think this competition is going to help me to practice my machine learning skills and improve my working skills.

Problem Statement

The project is using transaction data which provided by Zillow to predict the variable called log error which is defined as $logerror = log(Zestimate) - log(SalePrice)$. The target variable is continuous, I could use the Multiple Regression, LGBM, or Neural Network to estimate based on all other independent features. The metrics which can measure the different models' performance are Mean Absolute Error between the predicted log error and the actual log error.

Datasets and Inputs

There are two datasets called 'properties_2016.csv' and 'properties_2017.csv', respectively and each row is one transaction record. Each row contains 58 features which describe the houses. Two datasets called 'train_2016_v2.csv' and 'train_2017.csv', respectively which contain the house's ID and true logerror. Since there are no testing dataset, randomly sample data points from the training dataset is necessary to evaluate the future models.

Solution Statement

I have done some research online to know that there is a algorithm called LGBM[1] which is a gradient boosting framework that uses tree based learning algorithm and it has high speed with handling the large size of data. And LGBM is going to be very popular because its high speed and focuses on accuracy of results. And LGBM could be tuned on loss function of mean absolute error.

Benchmark Model

I will apply Multiple Regression as the benchmark model because it is the simplest model and could use all independent variables in the model. In the Multiple Regression, I would like to use the function LinearRegression in the sklearn.linear_model. Before applying Multiple Regression, I will do three things: first, fill in all missing values, second encoding categorical variables, third, implement Feature Selection since there are several variables are highly correlated. After applying Multiple Regression, I will calculate the evaluation metrics in order to compare it with my solution model.

Evaluation Metrics

In this project, one of evaluation metrics is Mean Absolute Error between the predicted log error and the actual log error. As I mentioned how to calculate log error in problem statement part, the Mean Absolute Error is defined as

$$|log(actual) - log(predicted)|$$

And it is obvious to see that smaller Mean Absolute Error means better model performance.

Another evaluation metric is Mean Squared Error between the predicted log error and the actual log error, and it can be defined as

$$\frac{\sum_{i=1}^{n}(logerror(actual) - logerror(predicted))^2}{n}$$

The better model will with smaller number of Mean Squared Error.

Project Design

1. Get the dataset from Kaggle.
2. Explore the dataset to know how many and what variables in there. Identify the target variable (logerror) and independent variables.
3. Exploratory data analysis to know the distributions of target variable and explanatory variables. Calculate variance-covariance matrix of variables to know which variables are highly correlated. Dealing with missing values such as ‘fireplacecnt’, I will fill in 0 to NaN because NaN means there is no fireplace in the houses. Another type variable with

missing values is ‘propertylandusetypeid’, there is no any clue to know how to fill with missing values. For this kind of variable, I would conduct a knn algorithm to predict missing values based on all other variables.

4. Encode categorical variables and split the dataset into train/test sets to make the data ready for models.
5. Build the benchmark model, Multiple Regression, to calculate the evaluation metrics (Mean Absolute Error and MSE)
6. Build and tune the LGBM model to the best model performance (the smallest Mean Absolute Error or MSE).
7. Compare the LGBM to the benchmark model and makSe conclusion.

Reference

[1] What is LightGBM, How to implement it? How to fine tune the parameters?