Two Sigma Connect: Rental Listing Inquiries

Two Sigma Connect: Rental Listing Inquiries

Objective

The objective of this competition is to predict how popular a new listing is based on the listing content like text description, photos, number of bedrooms, price, etc.

Data

The data comes from renthop.com, an apartment listing website. These apartments are located in New York City.

The target variable, interest_level, is defined by the number of inquiries a listing has in the duration that the listing was live on the site.

In the training dataset, there are 13 raw features, which are bathrooms, bedrooms, building_id, listing_id, manager_id, created_date, display_address, street_address, description, a list of features about this apartment, latitude, longitude and price.

There are also listing images organized by listing_id.

Challenge

I need to do preprocess on text (description) and date (created_date).

I need to choose different encoding techniques, such as label encoding, one-hot-encoding, frequency encoding and leave-one-out encoding for different categorical variables.

I need to understand the business lll and create meaningful features from simple features such as bathrooms, bedrooms, price, latitude, longitude, manager id and etc.

Feature Engineering

1. I use TfidfVectorizer() and CountVectorizer() in Python to extract words from description, features and street address. CountVectorizer() has better performance.

2. I create year, month, day, day of lek, day of year, lek of year, hour, if it is holiday from created_date.

3. I create a lot of new features such as price per bathroom, price per bedroom, price over average price in certain boroughs, length of description, words of description, if there is phone number in description, if there is email in description, number of exclamations in description, number of words in uppercase in description, average interest level per manager, distance between street address and display address and etc.

4. I use label encoding and one-hot-encoding for normal categorical variables. For high-cardinality categorical variables (building id, manager id, display address), I use leave-one-out encoding.

   A Preprocessing Scheme for High-Cardinality Categorical Attributes in Classification and Prediction Problems

Model

I use the xgboost and lightgbm library in Python to building gradient boosted tree models and apply blending technique on the models.

I use cross validation to train models and use BayesianOptimization model in Python to automatically tune parameters.

The final model has 10 lightgbm models and 10 xgboost models in the first layer and 10 xgboost models with blended features in the second layer.

Improvement

1. Include models with less correlation with gradient boosting models, such as random forest (RF) models, ET models, Keras models, StackNet and multi-layer perceptron model, into the final blending procedure.

   2nd Place Solution

2. Include features extracted from images.

3. Be more creative and try to create more new features from simple features.