Udacity Machine Learning NanoDegree

AVITO DEMAND PREDICTION CHALLENGE

Capstone Proposal

Sachin Lamba

Feb 2019

- ***Domain Background***

Today, any website we visit always tried to recommend something depending on our taste (or we can say history). Machine Learning is a field to check from data what *history* can say about our future tastes.

In India, Some companies like OLX are helping their users to sell their products online. But What Sellers (Customer for company) don't know about it is that what details they are filling, can make their items price go up or down. Machine Learning can help in predicting the likelihood of the price depending on the info provided by the seller.

Similar to this situation, Avito (Russian based company) created a challenge on kaggle (https://www.kaggle.com/c/avito-demand-prediction) to predict demand for an online advertisement based on its full description, its context and historical demand for similar ads in similar contexts.

I want to check how provided details of a product can affect its demand in live market. It is like a recommended system, except that the user will get a prediction of how well their product is and how well he defined it for a profit.

- ***Problem Statement***

When selling used goods online, a combination of tiny, nuanced details in a product description can make a big difference in drumming up interest. Details like:

Well-Taken, Authentic Photos

Too Glossy

Authentic

Poor Quality

Believable and Informative Description Copy

Description:
AMAZING WATCH FOR SALE!!!!

DON'T MISS THIS DEAL. IT'S THE DEAL OF THE CENTURY!!

Unlikely

Description:
I have an adjustable Chaleur D'Animale Watch for sale.

It's never been worn and still in the original box. Battery included.

Informative

Description:
fancy watch for sale

no low ball offers, cash and carry

Poor Quality

And, even with an optimized product listing, demand for a product may simply not exist–frustrating sellers who may have over-invested in marketing.

Avito, Russia's largest classified advertisements website, is deeply familiar with this problem. Sellers on their platform sometimes feel frustrated with both too little demand (indicating something is wrong with the product or the product listing) or too much demand (indicating a hot item with a good description was underpriced).

In this Kaggle competition, Avito is challenging to predict demand for an online advertisement based on its full description (title, description, images, etc.), its context (geographically where it was posted, similar ads already posted) and historical demand for similar ads in similar contexts. With this information, Avito can inform sellers on how to best optimize their listing and provide some indication of how much interest they should realistically expect to receive.

*I want to take a **subset** of their data to test and verify how prediction by details help a user to get best price of their product.*

I will be treating this problem as Regression one. As the value can be in [0,1] both inclusive, as we can never be sure 100% of ad demand.

- ***Datasets & Inputs***

Dataset is available under challenge: https://www.kaggle.com/c/avito-demand-prediction/data

- train.csv - Train data
  - item_id - Ad id
  - user_id - User id
  - region - Ad region
  - city - Ad city
  - parent_category_name - Top level ad category as classified by Avito's ad model
  - category_name - Fine grain ad category as classified by Avito's ad model
  - param_1 - Optional parameter from Avito's ad model
  - param_2 - Optional parameter from Avito's ad model
  - param_3 - Optional parameter from Avito's ad model
  - title - Ad title
  - description - Ad description
  - price - Ad price
  - item_seq_number - Ad sequential number for user
  - activation_date- Date ad was placed
  - user_type - User type
  - image - Id code of image. Ties to a jpg file in train_jpg. Not every ad has an image
  - image_top_1 - Avito's classification code for the image
  - deal_probability - The target variable. This is the likelihood that an ad actually sold something. It's not possible to verify every transaction with certainty, so this column's value can be any float from zero to one
- test.csv - Test data. Same schema as the train data, minus deal_probability

Missing Attributes in Columns I will fill in data preprocessing step:

title, category_name, parent_category_name, description, param_1, param_2, param_3

Subset Columns I will not use for my prediction:

Image, image_top_1

Abstract Data:

- Training Data have about *1503424* rows.
- Testing data have 508438.
- I will do split of data in training set and validation set in ratio of 0.75:0.25.
- Deal_probability will be the target column. As I checked with a threshold of 0.5 for this, this class is not balanced, I will be doing resampling technique.

- ***Solution Statement***

First we need to Preprocess data like filling missing column values, one-hot encoding for text data and split of train.csv data into training and validation dataset.

As this is a supervised learning problem with a regression solution with range [0,1]; I want to use different ensemble methods with tuning of hyper-parameters for the best model later for improvements. I will be trying following algorithms:

- XGBRegressor
- GradientBoostingRegressor
- AdaBoostRegressor
- CatBoostRegressor

- ***Benchmark Model***

As this is like a recommended system, I want to use Decision tree as with *GridSearchCV* as hyperparameter tuning to save its best model.

- ***Evaluation Metrics***

Metrics are used to evaluate how our algorithms are working. As per the Kaggle challenge, I will be using the following metric:

Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \widehat{y_i})^2}$$

$\widehat{y_i}$ is the predicted value and y is the original value.

- ***Project Design***

  Order for project solution

  - Exploring the data
    - Loading libraries & data
    - Visualization of data
  - Data preprocessing/cleaning
    - Text data will be one-hot encoded for the required text columns (except for item_id, user_id, title, description).
    - As I will be working with csv data, Image related columns will be dropped.
    - For Outliers detection, I will be using Interquartile range process.
    - I will fill the missing values with some constant.
    - Split data into training & validation set
  - Evaluate Algorithm
    - Build Models (try in between AdaBoost, GradientBoost, Lightgbm, Xgboost, Catboost)
    - Select best model
    - Make predictions on validation set
  - Model tuning to improve results
    - Select best model with tuning hyper-parameters. First I will choose a range of parameters in some step increments & also some randomized values for covering large parameter space also.
  - Final Conclusion
    - If the trained model is giving me accuracy upto what I want or similar to kaggle competition, I will be satisfied. I will create custom ensemble model, if required for fallback if nothing work to my expectations.

- ***References***

    - Cite as: **arXiv:1603.02754 [cs.LG]**

      (https://arxiv.org/abs/1603.02754)

    - https://medium.com/@gabrieltseng/gradient-boosting-and-xgboost-c306c1bcfaf5

(I will be adding other references I will need in future work.)