Udacity Machine Learning Nanodegree Capstone Project Proposal

Canburak TÜMER

Domain Background

Online marketplaces, classified advertisement websites have the same problem, which is the low quality advertisement that cause a low profile demand. Since the aim of these websites is to create sales, they desire more quality advertisements which drive demand and bring traffic to their websites.

Problem Statement

Avito, a Russia based online classifed advertisement company, wants to solve this demand problem. So they would like to predict demand from the features of advertisements and warn the seller about how to optimize the advertisements' features and get a better demand or better income.

Aimed output of the problem is a probability value between 0 and 1, inclusive and continous. This probaility is interpreted as demand. That's why problem seem to be a regression problem, but I believe it can be also solved by Bayesian approach and using Naive Bayes. The input which will be used is the data of former ads, which are explained in more details in Datasets and Inputs topic of this document.

Datasets and Inputs

Dataset consists of two sets of four files. One of the sets is the training set, and the other one is the test set. All the test set records are stripped from the target variable, which is **deal_probability** for this problem.

Let's deep dive into files in the training set:

train.csv 307.61MB :

Train data with the following features

- item_id
- user_id
- region
- city
- parent_category_name
- category_name
- param_1
- param_2
- param_3
- title
- description
- price
- item_seq_number
- activation_date
- user_type

- image
- image_top_1
- **deal_probability**

train_active.csv 2.52GB :

Supplemental data from ads that were displayed during the same preion as test.csv. Same schema as the train data, minus **deal_probability.**

train_jpg.zip 49.39GB :

Images from the ads in train.csv

periods_train.csv 170.26MB :

Supplemental data showing the dates when the ads from train_data.csv were activated and when they were displayed. With following features :

- item_id
- activation_date
- date_from
- date_to

All the data can be found on Kaggle by following this link : https://www.kaggle.com/c/avito-demand-prediction/data

Solution Statement

Since the train data has its target variable, and the aim of the problem is to predict this variable. It is a nice example for the supervised learning. I will be trying different supervised models for the problem. Including Naive Bayes to Logistic Regression. Even a normal regression model may be fit to data to get an continous number output.

Benchmark Model

Benchmark model is Naive Random Forest model and it scored a 0.243 with given evaluation metric. Since the metric is RMSE, the model I am trying to develop needs to be have a lower score than 0.243 to be accepted as a successful model. Because this problem is originally a Kaggle Competition, they've provided this benchmark model and its score for the competitors to evaluate their models success.

Evaluation Metrics

Project output will be scored by using Root Mean Squared Error formula. Formula can be seen below, where y-hat is the calculated probability and y is the real probability for the records. Since the output is a probability value. It would be easier and meaningful to calculate the distance between predicted and real values for evaluation. And to avoid negative and positive values to even-out themselves, we are using the square of errors. And since this task is an Kaggle competition originally, they've selected RMSE as the evaluation metric, that's why I will use it for evaluation.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

1 - Root Mean Squared Error

Project Design

Project will flow step by step, following the outline below

1. Exploratory Data Analysis : In this phase I will look into data to learn more about its structure, missing values, cleaning abilities, highly correlating features, derivable new features form the existing ones. I will focus on train.csv and periods_train.csv files because of my hardware limitations. I am aware that there may be more useful features to extract in other files also but my capacity is limited.
2. Generating Model Inputs : After EDA phase, I will clean, merge and transform the data to prepare it to models to read.
3. Training Models : I will train at least two different algorithms with a number of different hyper parameters. Also because I am not able to score my outputs using test data, I will divide my train data or use cross validation techniques to calculate success metrics of models.
4. Selecting Models : Depending on the success metrics I will select an model and a hyper parameter set for the selected model and use it against the test set.
5. Generating Results : After selected model ran agains the test set. I will submit the outputs to the Kaggle to see my final result. Depending on the success, I will loop between step number 1 and 5.