Project 5

OTTO GROUP PRODUCT CLASSIFICATION CHALLENGE

Shobhit Lamba | Applied Artificial Intelligence | 12/6/2017

Problem Statement

The Otto Group is one of the world's biggest e-commerce companies, with subsidiaries in more than 20 countries, including Crate & Barrel (USA), Otto.de (Germany) and 3 Suisses (France). We are selling millions of products worldwide every day, with several thousand products being added to our product line.

A consistent analysis of the performance of our products is crucial. However, due to our diverse global infrastructure, many identical products get classified differently. Therefore, the quality of our product analysis depends heavily on the ability to accurately cluster similar products. The better the classification, the more insights we can generate about our product range.

For this competition, we have provided a dataset with 93 features for more than 200,000 products. The objective is to build a predictive model which is able to distinguish between our main product categories. The winning models will be open sourced.

DATA DESCRIPTION

Each row corresponds to a single product. There are a total of 93 numerical features, which represent counts of different events. All features have been obfuscated and will not be defined any further.

There are nine categories for all products. Each target category represents one of our most important product categories (like fashion, electronics, etc.). The products for the training and testing sets are selected randomly.

FILE DESCRIPTIONS

- trainData.csv - the training set
- testData.csv - the test set
- sampleSubmission.csv - a sample submission file in the correct format

DATA FIELDS

- id - an anonymous id unique to a product
- feat_1, feat_2, ..., feat_93 - the various features of a product
- target - the class of a product

METHODOLOGY USED

The data is imported and preprocessed by dropping all ids and labels before training. Then, XGBoost (Extreme Gradient Boost) method is used for training.

STEPS TO RUN THE PROGRAM:

Pre-Requisites:

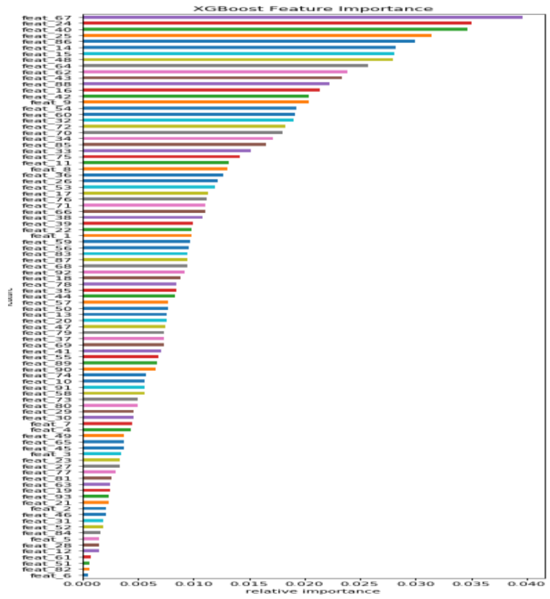
1. Python 3.x
2. SKLearn, Pandas, Matplotlib, XGBoost and Operator modules.
3. Note: To install XGBoost in Anaconda, run “conda install -c rdonnelly py-xgboost”.

Instructions:

1. Add deadlock.py, sampleSubmission, train and test data files in the same directory.
2. Execute deadlock.py.
3. It will produce an fmap, feature importance map and prediction file.
4. The prediction file can be submitted to Kaggle for evaluation and check the score.

RESULTS:

The feature importance graph looks something like this, with feat_67 on the top and feat_6 at the bottom. The best public score my code got was **0.45715.** The best score overall was **0.38055.**


XGBoost Feature Importance
feat_67
feat_24
feat_40
feat_25
feat_86
feat_14
feat_15
feat_48
feat_64
feat_62
feat_43
feat_88
feat_16
feat_42
feat_9
feat_54
feat_60
feat_32
feat_72
feat_70
feat_34
feat_85
feat_33
feat_75
feat_11
feat_8
feat_36
feat_26
feat_53
feat_17
feat_76
feat_71
feat_66
feat_38
feat_39
feat_22
feat_1
feat_59
feat_56
feat_83
feat_87
feat_68
feat_92
feat_18
feat_78
feat_35
feat_44
feat_57
feat_50
feat_13
feat_20
feat_47
feat_79
feat_37
feat_69
feat_41
feat_55
feat_89
feat_90
feat_74
feat_10
feat_91
feat_58
feat_73
feat_80
feat_29
feat_30
feat_7
feat_4
feat_49
feat_65
feat_45
feat_3
feat_23
feat_27
feat_77
feat_81
feat_63
feat_19
feat_93
feat_21
feat_2
feat_46
feat_31
feat_52
feat_84
feat_5
feat_28
feat_12
feat_61
feat_51
feat_82
feat_6
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040
relative importance