Bertelsmann/Arvato Project Proposal

by/ Noureldin Yosri

Domain background

we need to classify people into 2 classes (e.g. whether we should target a person with ads or not), the output will be a score between 0 and 1 where the higher the score the more likely this person to become a customer, in the end we can convert these score into 0,1 decisions based on whether the score >= a specific threshold, however for the model to learn we will use binary cross entropy

Problem statement

Essentially we want to answer the question: How can the mail order company acquire new clients more efficiently?

using the provided demographics dataset and the dataset of existing customers we want to know which individuals are more likely to be customers of the mail company if targets with ads

examples of this and similar problems include google or facebook[1] ads which decide on whether you as a customer should see a specific ad or not, a similar but more general problem is Amazon/Netflix recommender systems.

Datasets and inputs

the same dataset provided by Bertelsmann which as explained in the provided notebook are:

- `Udacity_AZDIAS_052018.csv`: Demographics data for the general population of Germany; 891 211 persons (rows) x 366 features (columns).
- `Udacity_CUSTOMERS_052018.csv`: Demographics data for customers of a mail-order company; 191 652 persons (rows) x 369 features (columns).
- `Udacity_MAILOUT_052018_TRAIN.csv`: Demographics data for individuals who were targets of a marketing campaign; 42 982 persons (rows) x 367 (columns).
- `Udacity_MAILOUT_052018_TEST.csv`: Demographics data for individuals who were targets of a marketing campaign; 42 833 persons (rows) x 366 (columns).

[1] Recommending items to more than a billion people

Solution statement

- using EDA, SVD and clustering algorithms identify and engineer important features
- transform each import using the transformation obtained from the EDA
- train a deep neural network to compute the scores and minimize the loss function

Benchmark model

an XGBoost model trained on Udacity_MAILOUT_052018_TRAIN.csv data

Evaluation metrics

the most suitable metric is the binary cross entropy since the output score/person is actually a probability that this person is a potential customer

Project design

1. run XGBoost to obtain the baseline model
2. Exploratory data analysis
    - examin NA values and decide which to try to extrapolate and which to drop
    - using correlation matrix and SVD reduce the dimensionality
3. Feature Engineering
    - use the features obtained from SVD
    - run clustering algorithms like DBScan, hierarchical and self organizing maps to cluster the data where the id of the cluster becomes a new feature in the resulting data set
4. split the training data into train, validation data
5. train a deep NN and use grid search and other methods to fine tune hyperparameters with respect to its performance on the validation data
6. obtain the final results and submit on kaggle
7. if results still not satisfactory refine and repeat :D