Machine Learning Engineer Nanodegree

Capstone Proposal

Customer Segmentation and Prediction – Arvato Financial Solutions

Binbin Zhou
May 21st, 2020

Proposal

Domain Background

Arvato, the second-largest division of Bertelsmann, is a global services company headquartered in Germany, whose service include customer support, information technology, logistics and finance.

In this capstone project, Arvato is trying to help a mail-order sales company in Germany to boost their client population. To achieve this goal, we are going to identify segments of the general population, understand the patterns of current customers, and target the potential customers to send out mail advertisement/invitation.

Problem Statement

The problem statement of this project is: How can this mail-order sales company send out mail advertisement/invitation to those people who are more likely to be converted to their customer?

I am going to solve this problem by the following steps:

1. Exploratory data analysis: data cleaning, categorical value encoding.
2. Use unsupervised learning techniques to cluster the general population and current customers, and look into the difference.
3. Based on the training sample provided, use supervised learning technique to build a model to predict how likely a person can be converted to their customer.

4. As a competition is hosted on the Kaggle, the prediction data will be uploaded to Kaggle so we can see how our model perform on out of time validation sample

Datasets and Inputs

There are 4 datasets provided by Arvato, all of which have demographic features:

1. ***Udacity_AZDIAS_052018.csv***: Demographics data for the general population of Germany; 891 211 persons (rows) x 366 features (columns)

2. ***Udacity_CUSTOMERS_052018.csv***: Demographics data for customers of a mail-order company; 191 652 persons (rows) x 369 features (columns)

3. ***Udacity_MAILOUT_052018_TRAIN.csv***: Demographics data for individuals who were targets of a marketing campaign, with a column "RESPONSE" to show if they turn into customers; 42 982 persons (rows) x 367 (columns).

4. ***Udacity_MAILOUT_052018_TEST.csv***: Demographics data for individuals who were targets of a marketing campaign; 42 833 persons (rows) x 366 (columns).

Besides, there are another 2 files containing the meta-information for features in these tables:

5. ***DIAS Attributes - Values 2017.xlsx:*** A detailed description of each data values for each variable

6. ***DIAS Information Levels - Attributes 2017.xlsx:*** A top-level description for each attributes, grouped by informational category

Data 1 and data 2 can be used for exploratory, setting up data cleaning procedures, and unsupervised customer segmentation; Data 3 will be used for model training, which will be applied to data 4 to generate the prediction. Data 5 and data 6 can be used to map to all demographic tables, which will be helpful to get a better understanding of the segmentation and model. I found data 5 is especially helpful to map missing/unknown data values.

Solution Statement

Mainly there are two parts for my solution:

1. Customer segmentation:
   During this part, I used unsupervised learning methods to cluster the population into different groups. First of all, non-numerical features are encoded, and feature scaling method is applied to prepare data for principle component analysis. A popular dimension reduction technique PCA is used, to reduce the dimension around half number of total features. After that, KMeans analysis is applied on the PCA processed dataset, to separate the whole population into subgroups.

   Segmentation can be applied both to the general population and the customer dataset, which can be used to compare with each other and look into the difference to find any potential pattern for customers.

2. Customer targeting model:
   Supervised learning technique is applied to build a model and predict who is more likely to be converted to a customer. For this particular project, a tree based model would be appropriate concerning that there are around 500 features in total. I choose light GBM as the target model for this project, which is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm.
   lightGBM is picked due to these reasons:
   - Faster training speed and higher efficiency
   - Lower memory usage
   - Better accuracy
   - Compatibility with large datasets

   Also, **GridSearchCV** is used for parameter tuning, for better performance of the model and avoiding overfitting.

Benchmark Model

A simple gradient boosting model without parameter tuning can be used as a benchmark model. The ROC-AUC (explained in the following part 'Evaluation Metrics') for such simple model is around 0.72 based on my test.

Evaluation Metrics

For principle component analysis, there are three types of model attributes:

- Mean: the mean that was subtracted from a component in order to center it.
- Components_: the makeup of the principle components;
- S: The singular values of the components for the PCA transformation.

From s, we can get an approximation of the data variance that is covered in the first n principle components. The approximate explained variance is given by the formula: the sum of squared s values for all top n components over the sum over squared s values for all components:

$$\frac{\sum_n S_n^2}{\sum S^2}$$

For the second part for the predictive model, as the target rate in the dataset is highly imbalanced (1.2%), it will not be appropriate to just use accuracy to evaluate the performance of the model. Instead, I proposed to use ROC-AUC, which is a performance measurement for classification problem at various threshold settings. ROC (**Receiver Operating Characteristics**) is a probability curve and AUC is the **area under this curve**. ROC curve is plotted with TPR (**true positive rate**) on x-axis against the FPR (**false positive rate**) on y-axis.

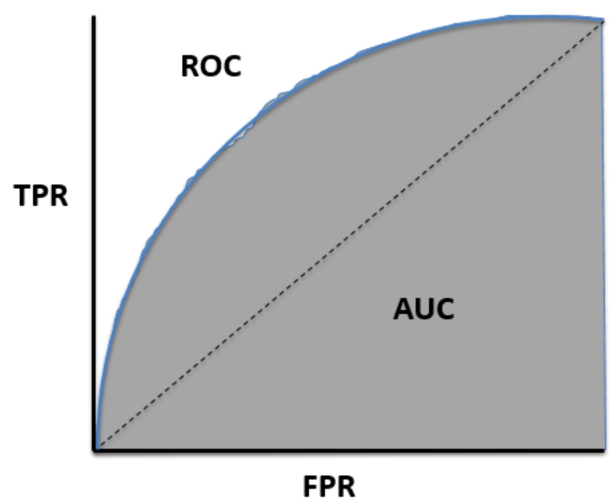

Image1: AUC-ROC curve (my own drawing)

As the image1 shows, higher area under the curve, model can have higher TPR with controlling FPR under a certain threshold. In our project here, higher ROC-AUC means that the model can identify higher percent of customers that can be converted to real customer, while in this targeted group, the false positive rate can be controlled in a relative small number.

Project Design

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

1. Check missing rate for each variable
   First of all, I checked the missing rate for each variable in the whole dataset, can try to remove those with high missing rate. When I check the meta-information for these features, I found that the data not only contains NaN, for lots of features, 0 or 9 could also represent missing/unknown, so I go through each variables and check the missing rate for NaN, 0, and 9 (depends on the metadata information for value explanation). Finally, I removed the features that have missing rate higher than 50%.

2. Check categorical features and re-encode
   Feature type is checked and categorical features will be picked in this step. Then I checked the number of unique levels for each categorical feature, if there are only 2 unique values, replace them with 0 and 1. Otherwise, the feature will be converted to dummy variables.

3. Combine former steps to a general preprocessing step
   A general preprocess function was generated to combine the steps for removing high missing rate features, and categorical data encoding. This function can be applied to the following steps to make sure all dataset have the same preprocess step.

4. Principle component analysis
   PCA is applied to reduce the dimension from original number of features (around 500 after encoding) to 300, which in total can explain around 95% of variance, and the top 150 components can explain 75.7% of variance.

Top 3 components were checked in detail and for each of these components, the top 3 and bottom 3 features contributed to this components were picked and their definition were checked to get more understanding.

5. KMeans analysis
   Based on the 300 components identified by the PCA in the upper steps, MiniBatchKMeans were applied to analyze the data further. Based on the elbow method, it seems that separate the population into 16 clusters gives the smallest SSE (sum of squared error). The algorithm was applied to both the general population and the customer data, whose clusters are compared to each other to identify the overrepresented cluster and underrepresented cluster compared to the general population. For these two clusters, I checked the mean value for the top features picked from the PCA analysis (top 3 and bottom 3 features for top 2 components, 12 features in total). Combining with the metadata information, I summarized some characteristics for the over represented group.

6. Supervised model building
   Based on former experience, a decision tree based model would be appropriate for this project. And as I mentioned above, light GBM was picked for model building, as compared to the popular algorithm XGBoost, lightGBM is faster and require smaller computer resource.

7. Parameter tuning
   GridSearchCV is applied to model training and parameter tuning. These parameters are tuned to gain the best ROC-AUC score for the model:
   - n_estimators
   - colsample_bytree
   - max_depth
   - num_leaves
   - min_data_in_leaf
   - reg_alpha
   - reg_lambda
   - min_split_gain
   - subsample

8. Make prediction and submit data for Kaggle ranking
   With the best model picked after parameter tuning, predictions for the test data (after preprocessing same to the training dataset), the submission on Kaggle returned the score 0.80437, which ranks 30th/187.