Carlos Sanmiguel Vila - 03/05/2020

Capstone Proposal - Machine Learning Engineer Nanodegree

Domain Background

Customer segmentation is one of the hot topics in many relevant industries such as financial service providers. Identifying the appropriate customer categories, the companies can better tailor their marketing efforts to various client subsets in terms of promotional, marketing and product development strategies or even able to identify potential customers. These problems can be tackled using both supervised and unsupervised algorithms using client variables such as their demographic information.

Because of this growing interest, Arvato Financial Solutions, which is one of the market leads in German, proposes a real problem of customer segmentation using their customer data. Moreover, to show the performance of our proposed model, the project has a Kaggle completion associated which makes the project more challenging.

Problem Statement

Arvato Financial Solutions wants to identify potential new customers using their customer data. With the aid of different machine learning techniques, the customer data will be analyzed to find patterns that allow us to identify which customers are more receptive to marketing campaigns from their non-current customer data.

In order to solve this challenge, the data provided by Arvato will be explored with the following objectives in mind:

- Inspect, clean and treat the data.
- Identify which variables characterize Arvato customers.
- Appropriately encode the categorical variables.
- Select a machine learning algorithm which allows us to obtain the best results with a reasonable computational time.
- Choose an appropriate metric to measure our results.

The potential solution that will be sought is a supervised Machine Learning model for classification, where the input features will be the information from both customers and non-customers, and the label will be binary, whether the person is a potential customer or not.

Dataset and Inputs

The data provided by Arvato consist of the following CSV files:

- **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 features (columns)

- **DIAS Attributes - Values 2017.csv**: A detailed mapping of data values for each feature in alphabetical order.
- **DIAS Information Levels - Attributes 2017**: A top-level list of attributes and descriptions, organized by informational category.

Inspecting the data, it is observed that the training data is composed of about 41500 non-customers and 500 customers which means that this is an imbalanced dataset.

Solution Statement

The solution to be considered is to train a supervised Machine Learning model which will be developed in Python. The goal is to distinguish which people are potential customers for Arvato.

With this purpose in mind, the dataset will be cleaned and an appropriate set of variables will be chosen. In order to ensure the replicability of the results, an appropriate preprocessing pipeline will be developed and the random seeds will be fixed. Finally, a gradient boosting algorithm will be tried as a potential solution.

Benchmark Model

In this project, the benchmarking will be a Kaggle competition that will allow us to check the quality of our model. According to the current Kaggle leaderboard, an AUC (Area Under the Curve) value around 0.6-0.8 will be considered as an acceptable result. Since most of the models are using Catboost algorithm, this will be used as a benchmark model and different configurations using this algorithm will be tested.

Evaluation Metric

In this project, the metric that will be chosen is the AUC since this is the metric that is employed in the Kaggle competition that is used as a benchmark. AUC is used since it is a better measure of classifier performance than accuracy because it does not bias on size of test or evaluation data.

Project Design

The workflow of the project will be developed in a Jupiter Notebook that will include the following tasks:

Data Processing

- Inspect the data.
- Compare the customer and non-customer datasets.
- Identify patterns in the data.
- Check the null values and clean them.
- Encode categorical values.
- Investigate the possibility of creating new variables.

Training Process

- Split the data into a stratified train and test samples.
- Deal with the imbalanced data.
- Identify the most appropriate classification algorithm.
- Make a cross-validation process to avoid overfitting issues.

Test Process

- Check the results using our test sample.
- Analyze the impact of using different hyperparameters.
- Interpret the results obtained and check the most relevant features.
- Compare our performance against the Kaggle leaderboard.

For this workflow, the following Python libraries will be used:

- Pandas [1] this library will be used for data processing tasks, such as inspecting the data, analyzing the null percentage and compare the feature distribution between the customers and non-customers data.

- Numpy [2] this library will be used for data processing tasks, such as auxiliary mathematical operations.

- Scikit-Learn [3] this library will be employed to split the data into train and test samples, scale features or encode categorical variables from the data.

- Catboost [4] this library will be employed to train the supervised model. This library is based on gradient boosting on decision trees which is currently one of the best techniques to tackle this type of problems. Catboost has the advantage of supporting categorical features and this is the motivation behind my choice. Our data has a large number of categories and employing Catboost can significantly reduce the preprocessing step.

- Scikit-optimize [5] will be used as a Bayesian optimizer for the hyperparameter search. This choice is motivated by the fact that Bayesian methods can find better model settings than random search in fewer iterations.

References

[1] https://pandas.pydata.org/

[2] https://numpy.org/

[3] https://scikit-learn.org/stable/

[4] https://catboost.ai/

[5] https://scikit-optimize.github.io/stable/