Machine Learning Engineer Nanodegree Capstone Proposal

Sergio Calderón Perez-Lozao - April 15th, 2020

The main objective of this project is to be able to segment Arvato Financial Solutions clients in order to extrapolate to prospects.

Contents

Note

This proposal is based on one of the projects that Udacity proposes at the nanodegree.

Domain Background

Customer segmentation is a very common issue in machine learning projects. It is possible to work both on the supervised part of the problem and in an unsupervised way, using clustering algorithms.

It is usually segmented by demographic or behavioural variables, in this case, it is a question of demographic variables.

For this problem it is useful to know that Arvato Financial Solutions is one of the market leads in German and whole world by doing risk management, payment solutions, factoring and collection.

Furthermore, the provided data in this exercise is provided by public data, as says one of its senior key account manager, Timo Reis, in this video. They struggle with any kind of data source and the implications that it has.

In this website we can see the summarized history of the company.

Problem Statement

Arvato Financial Solutions has a lot of customers and demographic data about them.

With this machine learning project we are going to explore this demographic data for finding patterns within them and trying to find the difference between these and non-current customer data.

Probably there are a lot of leads that have similar features that some Arvato´s customers, so they will be more likely to become an Arvato client than others. This is our hypothesis.

There are several problems with the problem data, as they have a very high nullity percentage. But with some different approaches in terms of selecting data population, algorithms or feature engineering we can address it.

This is a real life problem so it´s interesting to realize that many times we are going to encounter similar problems. It is not the typical Kaggle without missing values and with all the variables prepared where we only focus on the algorithm and few more details.

Here, decisions such as with which data set to train the models, how to avoid biases in the predictions and how to solve the problem that most variables are not numerical is much more important than tuning the models even though it is the most funny part.

In addition, actionability is quite important in a problem like this, where there is a vital need to be able to interpret the model results in order to, perhaps, make a good post-processing of the data and orquest the mailout campaigns well.

Solution Statement

We have two data sets, customer data and the one in which the characteristics are those of the general population of Germany.

One of the first things we have to look at is whether the datasets have the same features since above all, our customer set may have columns that the German population dataset don't and with that all our segmentation would be wrong.

We will also have to check the variables description in the documents provided to us to see if there are not some kind of data leakage as unique identifiers that could also breaks the clustering.

Once the two data sets are aligned we should see the nullity of each one of them. It is to be expected that there are more missing values in the general population part, but this can precisely alert us that some variables are reported much more when a lead becomes a client, so adding it to our machine learning algorithms can be counterproductive.

When we are sure which columns can be used to segment, it is very useful to create a target for this data and train a model to see if it is capable of distinguishing the population of Germany from our clients.

To deal with the problem of unbalance classes we will implement some techniques of downsampling of the majority class or oversampling of the minority class, in our case be client.

If this model doesn't performs well we have a problem, since the datasets are not easy separable.

This would be the end of the first part, exploratory data analysis and some useful insights from input data. Then we will implement some machine learning models with different data samples, features and hyperparameters, iterating in order to achieve a precise, interpretable, deployable model.

Datasets and Inputs

We are provided with four data sets and two spreadsheets with variable information, the data sets look like this:

- **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)

As we can see, the files have a lot of information, plus they will take a lot of work to clean up and understand. Understanding how more than 350 variables are impacting in implementing a model with them is not so easy, so it is important to rely on the spreadsheets you have given us and make a good descriptive analysis of the data to see how useful each of the attributes is.

Target variable is only presented in **Udacity_MAILOUT_052018_TRAIN.csv** and its ratio is 1 every 100 registers, an unbalanced problem that we have to trait on.

Benchmark Model

We will use three approaches and the choosing a winner:

1. Training with customers vs German population data
2. Training with train specific data
3. Training with both approaches described above

For first and third approach we have a large enough data population to be able to use "state of the art" algorithms such as catboost or lightgbm that know how to deal with categories automatically, either xgboost or gbm (H2O), for which it will be necessary to encode the categorical variables prior to use.

For the second approach we can apply this type of model (gbm models), but it is more likely to have problems of overfitting. Therefore, a logistic regression could work here, with a good encoding of the categorical ones and a standardization step of the variables in the model pipeline.

Evaluation Metrics

We will use experiment tracking from MLFlow to log all of the models, hyperparameters, metrics and the data samples we use to train them. This is a must because if we don´t use this type of organization it´s very difficult to have a well documented and easy reproducible machine learning project.

All the code using for these project will be hosted with love in Github and it will be reproducible and highly modular.

Because this is a binary classification problem (customer or non-customer), we will mainly use two metrics, auc and precision recall auc.

The use of auc is because it allows us to have an idea of how well we rank the leads/clients given the probability inferred by the model.

Since it is a rather unbalanced problem, only 1% of the training data records are clients, the use of pr auc is recommended over auc, for example in this article.

Therefore with these two metrics we can compare our models and see which ones perform better, always following good practices and separating the data well in order to validate correctly and also logging them.

Project Design

The whole project will be developed in python, with a module that facilitates the most repetitive tasks and a notebook to follow in a more visual way all the steps we are taking from the beginning of the project until its completion.

The main libraries that will be used for this project are pandas, sklearn, mlflow and shap. With this tools we have all of the capstone project lifecycle: data processing, modeling, interpretability, tracking and deploy. These libraries are not the only ones, since h2o, catboost or imbalanced-learn among others will also be used.

We will analyze the customer data to find useful patterns and then apply them using a machine learning model to output the propensity of a record to become a customer.

As we mention before, we will separate the problem into three data populations and we will run different experiment for each part, using state of the art gradient boosting machine models and logging them with MLflow.

The population in which the models are trained is as important as the models or their hyperparameters. In addition, the treatment we give to non-numerical variables will also play a very important role.

Interpreting our results is very important so we will use shap for this purpose.

The choosen model will be packaged in an MLproject so that it can be easily used by anyone without even using the source code or notebooks.

Finally, we score the test data set and send it to Kaggle to compare our method with that of other trainees. In real life, instead of this kaggle submission, we will trigger the model by sending the most likely leads to these mailout campaigns.

References

- Arvato Financial Solutions website
- Handling imbalanced datasets in machine learning
- A Unified Approach to Interpreting Model Predictions
- How to setup MLflow in production