

Carlos Sanmiguel Vila - 07/05/2020

Capstone Project - Machine Learning Engineer Nanodegree

Definition

Project Overview

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. Because of this growing interest, Arvato Financial Solutions, which is one of the market leads in Germany, proposes a real problem of customer segmentation using their customer data. The challenge proposed here is to identify potential new customers using their current customer data. With this purpose in mind, the following datasets are provided:

- **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.

The process described in this project is implemented in the Jupiter notebook, **Udacity_FinalProject.ipynb**.

Problem Statement

The final goal of the project is to create a model which can predict the people who are more likely to respond to a marketing campaign. This model must use the demographics data provided in the CSV file **Udacity_MAILOUT_052018_TRAIN** and its performance will be validated using the CSV file **Udacity_MAILOUT_052018_TEST** in a Kaggle competition.

Taking this into account, the project solution is divided into two phases. In the first phase, the data from **Udacity_AZDIAS_052018** and **Udacity_CUSTOMERS_052018** is used to analyze the different demographic features and obtain some first insights about the data quality or the features that may allow characterizing an Arvato customer. For this reason, a supervised model is implemented to classify people using a particular set of features and predict whether they are customers or not. In a second phase, the knowledge obtained by the previous analysis, i.e. null treatment, relevant features, model design, is employed to train a model using the data from

Udacity_MAILOUT_052018_TRAIN. Finally, this last model is upload to Kaggle to check its performance against the data from **Udacity_MAILOUT_052018_TEST**.

Metrics

In this project, the chosen metric 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. Apart from that, AUC also allows us to deal with datasets which have a skewed sample distribution such as this particular dataset.

Analysis

Data Exploration

To analyze the different features of the dataset, a split of 25% for the CSV files **Udacity_AZDIAS_052018** and **Udacity_CUSTOMERS_052018** is considered as representative. Using this split ratio the following shapes are obtained:

- Customer dataset (df_customer): (47849, 370)
- German Population dataset (df_azdias): (223060, 367)

At first sight, the proportion of null values is quite high some variables as it is observed in the screenshots reported in Figure 1. For this analysis, it has been considered as NaN value the values of ['D19_UNBEKANNT', 'X', 'XX', -1].

```
In [7]: df_azdias.head(10)
Out[7]:
```

Unnamed: 0	LNR	AGER_TYP	AKT_DAT_KL	ALTER_HH	ALTER_KIND1	ALTER_KIND2	ALTER_KIND3	ALTER_KIND4	ALTERSKATEGORIE_FEIN	ANZ_HAU
0	7	910261	NaN	1.0	14.0	NaN	NaN	NaN	NaN	14.0
1	10	645165	0.0	1.0	10.0	NaN	NaN	NaN	NaN	10.0
2	22	612606	NaN	8.0	0.0	NaN	NaN	NaN	NaN	NaN
3	23	612609	NaN	6.0	16.0	NaN	NaN	NaN	NaN	NaN
4	32	796646	NaN	4.0	19.0	NaN	NaN	NaN	NaN	18.0
5	37	796666	2.0	1.0	20.0	NaN	NaN	NaN	NaN	13.0
6	42	796676	NaN	3.0	19.0	NaN	NaN	NaN	NaN	19.0
7	45	507843	1.0	1.0	21.0	18.0	NaN	NaN	NaN	11.0
8	46	507844	NaN	9.0	0.0	NaN	NaN	NaN	NaN	NaN
9	53	507892	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
In [13]: df_customer.head(10)
Out[13]:
```

Unnamed: 0	LNR	AGER_TYP	AKT_DAT_KL	ALTER_HH	ALTER_KIND1	ALTER_KIND2	ALTER_KIND3	ALTER_KIND4	ALTERSKATEGORIE_FEIN	ANZ_HAU
0	3	143873	1.0	1.0	8.0	NaN	NaN	NaN	NaN	8.0
1	14	102239	2.0	1.0	6.0	NaN	NaN	NaN	NaN	6.0
2	15	110278	2.0	5.0	17.0	NaN	NaN	NaN	NaN	12.0
3	23	110344	NaN	7.0	14.0	NaN	NaN	NaN	NaN	14.0
4	27	5545	NaN	2.0	19.0	NaN	NaN	NaN	NaN	NaN
5	40	125539	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	41	125543	2.0	1.0	15.0	NaN	NaN	NaN	NaN	9.0
7	42	125547	2.0	1.0	9.0	NaN	NaN	NaN	NaN	8.0
8	43	125549	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	51	125584	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 1 Overview of a subset of rows and columns.

Comparing both datasets, it is observed that there are extra columns in the customer dataset ['CUSTOMER_GROUP', 'PRODUCT_GROUP', 'ONLINE_PURCHASE']. Since these features are not present in the German Population dataset, they are dropped as a preprocessing step. Additionally, 'LNR' appears to be a data index and 'EINGEFUEGT_AM' an insertion date. For this reason, both features will be also dropped from both datasets.

Exploratory Visualization

Analyzing the data type of the different features, the following distribution is obtained:

- **float** 64 273 features
- **int64** 90 features
- **object** 4 features

It has to be remarked that most of the variables are categorical. For this reason, the calculation of statistics, such as mean or variance, is not very helpful. Moreover, due to a large number of features, analyzing the distribution of every categorical feature is a long hard work. That is why a first model will be implemented and based on the results obtained, the following steps will be taken.

As observed in the previous section, the null values are quite relevant in some variables. Therefore, an analysis of their distribution in both datasets is carried out. In Figure 2, different ranges of null contain are considered. For clarity, the null levels are considered in the ranges: (100%-40%), (40%-20%).

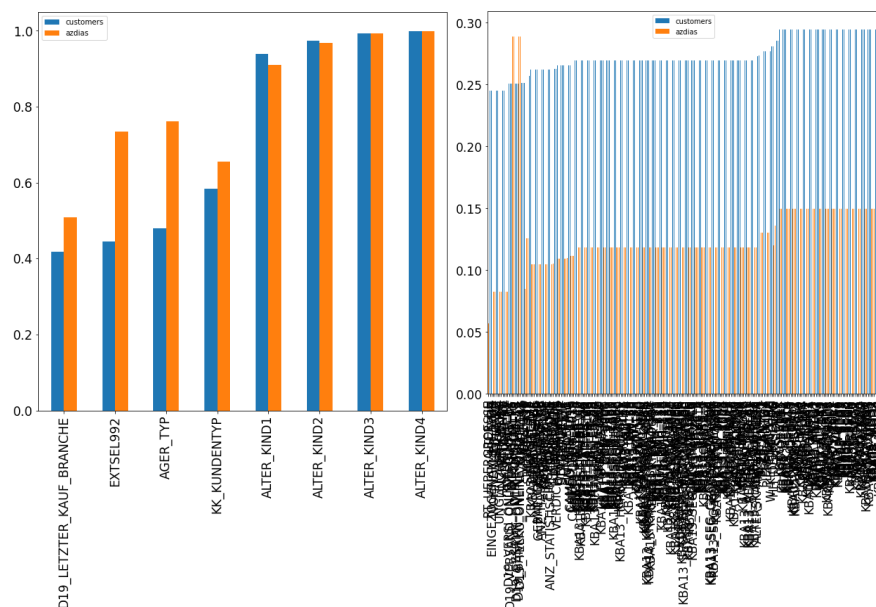


Figure 2 Percentage of NaN values for different features. Left, features with a NaN contain between 100% and 40% in the customer dataset. Right, features with a NaN contain between 40% and 20% in the customer dataset.

Comparing both null distributions in Figure 2, it is observed that they have similar behavior. In Figure 3, the difference between both null distributions is shown. These results will later be used to drop some features to try to improve the results of the model.

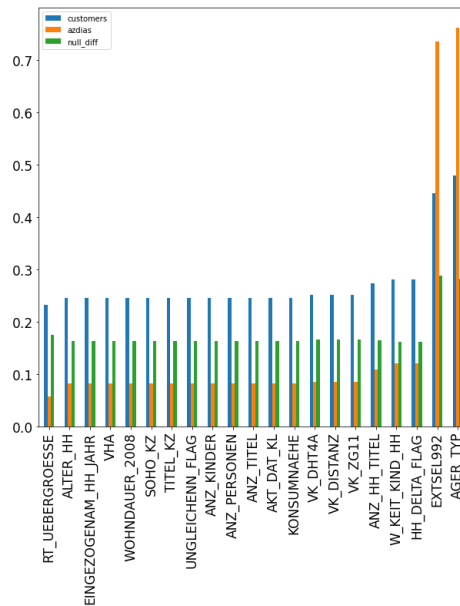


Figure 3 Features with larger differences in null distributions.

Algorithms and Techniques

In order to choose the most appropriate algorithm, some key points from the previous sections must be highlighted:

- Our dataset has a high number of features.
- Most of the features are categorical.
- As observed in **DIAS Information Levels - Attributes 2017**, most of the categories have a large number of levels.

Taking into account these conclusions, Catboost is chosen as the most appropriate algorithm for a first trial. The reasons for choosing this algorithm are the following:

- Catboost algorithm is a supervised algorithm which is based on gradient boosting on decision trees [2] which is currently one of the best techniques to tackle this type of problems. It is used by many Kaggle competition winners.
- Catboost has the advantage of supporting categorical features and null values without previous preprocessing. Since the data has a large number of categories and null values, employing Catboost can significantly reduce the preprocessing phase.
- Catboost allows analyzing which features are more relevant for the model performance.

Benchmark

In this project proposed by Udacity, there is a Kaggle competition [3] which is used as a benchmark. Figure 4 shows the first 10 and the last 10 competitors as a reference. It has to be remarked that values higher than 0.5 must be obtained to beat the randomness.

Using this as a benchmark score, the goal of this project is to get an AUC value close to 0.8.





















#	Team Name	Notebook	Team Members	Score 📊	Entries	Last
1	Ambresh Patil			0.81063	58	3mo
2	Julio Guijarro Hernandez			0.80954	16	1h
3	[Deleted]			0.80954	12	2h
4	Telmo			0.80936	57	3mo
5	TensorFrozen(Shihao)			0.80819	29	1y
6	Gaurav Ansal			0.80816	27	9mo
7	weft169Aston			0.80811	62	3mo
8	Michel N			0.80777	24	1y
9	Tianxiang Ma			0.80762	1	1y
10	Ahmed			0.80687	63	2mo
164	Mitch Rea			0.50797	1	5mo
165	Pavan			0.49964	1	5mo
166	Fabian Seitz			0.49367	2	1mo
167	RebeccaKelly			0.49005	1	5mo
168	Tobias Hoke			0.48275	1	1mo
169	Nokaido			0.47020	1	6mo
170	Bill Tong			0.44490	1	7mo
171	nikator_19			0.41285	2	3mo
172	Guilherme			0.39195	1	21d
173	Benjamin Ehrensberger			0.30535	4	5mo

Figure 4 Kaggle leaderboard on 7th May.

Methodology

Data Preprocessing

For the preprocessing of the data, the following steps:

- The following values have characterized to NaN ['D19_UNBEKANNT', 'X', 'XX', -1]. The German word UNBEKANNT means unknown and therefore, it has been included as NaN.
- NaN values have been replaced to 999 values to configure CatBoost to get minimal values as null ones.
- Features which are not present in both dataset (German Population and Customers) are dropped ('CUSTOMER_GROUP', 'PRODUCT_GROUP', 'ONLINE_PURCHASE').
- Features which do not include any relevant information such as index data or insertion date are also dropped ('LNR', 'EINGEFUEGT_AM').

- Features which have a high number of NaN values are characterized.

Implementation

Once that the preprocessing stage is finished, the dataframe obtained from **Udacity_AZDIAS_052018** and **Udacity_CUSTOMERS_052018** is combined in a unique dataframe that will be used to train our first classification model. The feature 'arvato' is created to classify which rows are from Customers (1) or German Population (0) dataframe. After that, the data is split into the following parts 80% train set and 20% test set. Since the dataframe is highly imbalanced, the split has been done in a stratified manner.

After that, the following Catboost model is created:

```
1. model_train = CatBoostClassifier(n_estimators=45,  
2.                               max_depth=3,  
3.                               loss_function='Logloss',  
4.                               nan_mode='Min',  
5.                               custom_metric=['AUC'],  
6.                               random_state=seed)
```

With this model an AUC value of 0.932 is obtained. Since this result is quite good, the next step is to move forward towards the Kaggle competition.

With this purpose in mind, the same preprocessing steps described in previous sections are applied to the dataset **Udacity_MAILOUT_052018_TRAIN**. After that, using the feature 'RESPONSE' as the target variable, the previous model is used. The score obtained is an AUC value of 0.753 which is significantly lower than the AUC obtained for the previous dataset. One of the possible reasons for this result may be that the data in both CSV files do not have the same distribution, so the model would be affected. For this reason, a second Catboost model is trained using the data from **Udacity_MAILOUT_052018_TRAIN**. In this case, the AUC obtained is 0.8447 which is an improvement to the previous prediction.

Using this model, a first Kaggle submission is performed and the score obtained is around 0.79 with a 78 position in the leaderboard. This is not a bad result since we are in the top 50% and our result is close to the leader 0.81, but it is worth to try to improve our result.

The part of finding a model with a score higher than 0.79 was the most challenging part of the project. Different hyperparameter combinations were tested, and despite the improvement in the training score, the test score was not enhanced due to overfitting issues.

Refinement

In order to improve our results, the parameters of max_depth and the n_estimators were modified with different values but there was not any model improvement. After different tests, the most common concern identified was overfitting of the model in the training data. This is a typical issue when working with gradient boosting algorithms. To address this issue, the learning rate was reduced from 0.5 to 0.25 and as a consequence, the test score was improved to 0.80485 with a train score of 0.822.

```

1. model_train_kag = CatBoostClassifier(n_estimators=45,
2.                                     max_depth=3,
3.                                     loss_function='Logloss',
4.                                     nan_mode='Min',
5.                                     custom_metric=['AUC'],
6.                                     random_state=seed,
7.                                     eta = 0.25)

```

In other attempts, the features with a higher number of nulls were dropped, but the model performance was also not improved.

Results

Model Evaluation and Validation

To evaluate the model robustness a cross-validation study using 5 folds is performed. The resulting AUC has a value of 0.76964 which is lower than the 0.822 obtained, but it is a satisfactory result.

```

1. auc_scores = []
2. n_splits = 5
3.
4. skf = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=seed)
5.
6. for tr_ind, val_ind in skf.split(X_kag, y_kag):
7.     X_train = X_kag.iloc[tr_ind]
8.     y_train = y_kag.iloc[tr_ind]
9.
10.    X_valid = X_kag.iloc[val_ind]
11.    y_valid = y_kag.iloc[val_ind]
12.
13.    model = CatBoostClassifier(n_estimators=45,
14.                              max_depth=3,
15.                              loss_function='Logloss',
16.                              nan_mode='Min',
17.                              custom_metric=['AUC'],
18.                              random_state=seed,
19.                              eta = 0.25,
20.                              logging_level='Silent'
21.    )
22.
23.    model.fit(X_train,
24.             y_train,
25.             cat_features=cat_features_kag,
26.             eval_set=(X_valid, y_valid)
27.    )
28.
29.    y_pred = model.predict_proba(X_valid)[: , 1]
30.    auc = roc_auc_score(y_valid, y_pred)
31.    auc_scores.append(auc)
32.
33. sum(auc_scores)/n_splits

```

The features more relevant for the model are shown in Figures 5 where the SHAP values are used to the feature relevance. The more relevant values are D19_SOZIALES, D19_KONSUMTYP_MAX and EINGEZOGENAM_HH_JAHR. None of these features is described in the documentation, the only variable that it is reported in the documentation is D19_KONSUMTYP as “consumption type”, and therefore we can assume that D19_KONSUMTYP_MAX is connected with this one. We cannot draw any firm conclusion regarding the interpretation of our model without this information.

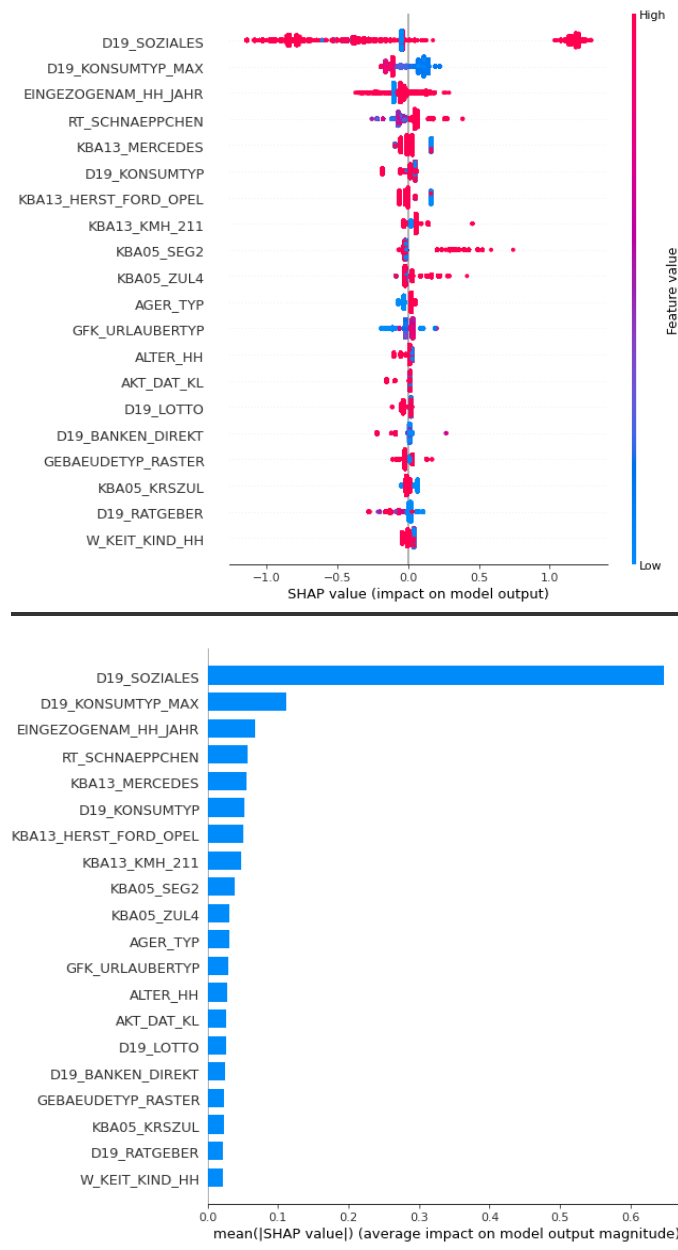


Figure 5 SHAP values for the best model.

Justification

Finally, the Figure 6 shows the Kaggle leaderboard where our best model is ranked in a 23 position which corresponds to a top 13%. This result is quite good not only for the position obtained but also because its score is only 0.05 below the top 1. These results allow accepting the model as a good business solution. If we could get additional information regarding the most relevant features, maybe the model would get better performance.

	Overview	Data	Notebooks	Discussion	Leaderboard	Rules	Team		My Submissions	Submit Predictions
13	Aaron Davis								0.80660	35 9mo
14	ming								0.80657	1 10mo
15	Lisa Vladulescu								0.80640	21 1y
16	Kirill Shipitsyn								0.80570	7 4mo
17	Dlee152								0.80570	10 3mo
18	Labienus								0.80568	3 9mo
19	Rahul Dixit								0.80561	7 1y
20	jxtrbtk								0.80557	163 3mo
21	anacolada								0.80555	15 4mo
22	FC Su								0.80526	57 9mo
23	Carlos Sanmiguel								0.80485	12 1h

Figure 6 Kaggle leaderboard on 7th May.

References

- [1] Prokhorenkova, Liudmila, et al. "CatBoost: unbiased boosting with categorical features." *Advances in neural information processing systems*. 2018.
- [2] https://en.wikipedia.org/wiki/Gradient_boosting
- [3] <https://www.kaggle.com/c/udacity-arvato-identify-customers/>