Capstone Project

Manuel Vazquez

Machine Learning Engineer Nanodegree

April 13th, 2020

Customer Intelligence for Arvato Financial Services

Definition

Project Overview

We will analyze demographics data for customers of Arvato Financial Services, a mail-order sales company in Germany, comparing it against demographics information for the general population. A mail-order company offers the service of buying things without the need of going to the shop in person, that is, tele-buying (tele is a prefix that stands for 'in the distance'). Unsupervised learning techniques such as principal component analysis (data reduction) and k-means (clustering) will be use to perform customer segmentation. Then, we'll apply the previous adquired knowledge to predict which individuals are most likely to become new customers of the company, i.e., propensity modelling. To achieve this final and most important goal we will train a fully connected neural network.

Problem Statement

The problem we want to solve is: find the most prone citizens to become new customers of the mail-order company Arvatos Financial Services. It is important for the company to target the most prone clients so that it can increment the sales in the best optimized way, and it is important for the client to recieve the offers that best fits its needs, that is, customer intelligence.

Evaluation Metrics

The performance of the model will be evaluated with the area under the curve metric (AUC) in the test dataset. This is a good metric because the dataset is highly imbalanced (most of the citizens don't answer marketing campaigns). Other metrics like precision, recall and F1 score are also viable measures. Accuracy wouldn’t be a good option in this environment.

Project Design

First of all, a preprocessing of the data is mandatory, deleting records with not available (NA) information. The columns which are highly linearly correlated will be deleted also. Categorical variables will be transformed into n-1 dummy variables where n is the number of classes in the variable. Before standardizing the data a brief descriptive summary of the variables will be done (mean, mode and quantiles). Some illustrative graphs will be plot before standardizing. After this step, k-means algorithm will be use to achieve unsupervised learning. Finally, a fully connected layer neural network will be set up for modelling.

Benchmark Model

Previous benchmark studies suggest that there really exists a difference in performance between white box and black box models. We will use the logistic regression as the bechmark model for the supervised classification task.

Analysis

Datasets and Inputs

The data that we will use has been provided by Bertelsmann Arvato Analytics, and represents a real-life data science task.

There are four data files associated with this project:

- 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).

Each row of the demographics represents a single person, but also includes information outside of individuals, including information about their household, building and neighborhood.

Data preprocessing

Before starting the technical analysis of the problem let's make a parenthesis. As it can be concluded of the previous section the stated problem is complicated due to the high dimensionality of the datasets provided. The best way of facing this problem would be having a deep understanding of the data and of the problem domain, i.e., customer intelligence. Given that reaching a reasonable knowledge of both the dataset and problem domain exceeds the intented time to complete the project, a shortcut was designed. The first and most important part (having a deep understanding of the data) can be overcome with variable selection. The second one (having a deep understanding of the domain knowledge) is important for building the network and getting the most powerful model for prediction (also for checking results), that is, fine tuning. So this part is not considered essential and we can reach very competitive results without solving it, although they won't be the best ones.

The first thing to be done before anything else is the treatment of the not available cells or missing values. There are lots of methods for doing this, from deleting to imputing. Deleting will be used to reduce the dimensionality of the dataset. Imputation of missing values can be done with the median of that feature or any other descriptive statistic measure, but to be done correctly a knowledge of the data is needed.

The applied criterion in the CUSTOMERS dataset is to delete all the columns containing more tan 20% of missing values and then deleting all the records containing missing values. After this the resulting dataset is ready for any other analysis and the dimension is reduced from (191652, 369) to (185560, 113). This is a reduction of 69.3% in the features space and of 3.2% in the records space.

Before applying any other method to the dataset such as PCA or k-means it is very important to standardize the numerical features and translate the categorical to dummies so that the algorithms work as desired. Outliers detection won't be considered in this project.

Variable Selection

With this step we try to discover which are the features that best explain the dataset, not a transformation of them. This is not exactly a principal component analysis but it can be deducted of it. Of the 113 features that we have we want to find, for example, the 5 that best represent the dataset. Once we have found this features we can perform a exploratory analysis (descriptive statistics and plots).

There are 3 ways to select the variables that I propose:

- Principal component analysis (PCA): remember that the principal components are linear combinations of the datasets columns. The coefficients of this combinations are called the loadings. So, the loading of the principal component that explain the maximun variance of the dataset can be used to choose which columns are most important for this component (the ones with the higher coefficients), that is, the features that most represent our dataset. For this method to work it is very important to standardize the dataset first. Below an example to understand this method:

  $$PC_1 = 0.9 * C_1 + 0.2 * C_2 + 0.1 * C_3$$

  PC_1 is the principal component that explains the maximun variance of the dataset, C_1 is the first column of the dataset, C_2 the second and C_3 the third. The vector (0.9, 0.2, 0.1) are the loadings. It can crearly be seen that the first principal component is more sensitive to changes in the first column that to changes in the other columns (because the coefficient is greater), so the first column of the dataset is more important than the rest of columns. We would conclude that column one best represents the dataset and would be the choosen variable.

- Sparse principal component analysis (SparsePCA): this method is very similar to the one above but it includes a penalty to some coefficients in the loadings, just like ridge and lasso regression methods (also used in feature selection) Below an example to understand this method:

  $$PC_1 = 0.98 * C_1 + 0.002 * C_2 + 0.001 * C_3$$

The conclussions can directly be drawn if above example was understood. This method is more difficult to do in practice because of the hyperparameters that have to be set manually.

- Logistic regression: fitting a model to explain a response variable with some explanatory variables would end in a linear combination as the later ones (assuming the fitting is linear in the estimation parameters). This method could be improved by exhaustive model selection, forward model selection and backward model selection.

The first principal component of the PCA analysis explains 31% of the variance and results in the next 5 features:

1. FINANZ_ANLEGER: financial activity as investor
2. D19_KONSUMTYP_MAX: consumption type
3. VERS_TYP: insurance typology
4. FINANZ_UNAUFFAELLIGER: unremarkable financial typology
5. SEMIO_VERT: affinity indicating in what way the person is dreamily

The Sparse PCA method of the scikit learn library does not have implemented the explained variance of each component. It is very easy to implement the explained variance given the eigenvalues of the covariance matrix of the dataset. Anyway, the explained variance of the first sparse PCA component coincide with the explained variance of the first PCA component. The next 5 features have been found to be the most relevant ones:

1. D19_BANKEN_DAUM: actuality of the last transaction for the segment Banks Total
2. D19_BANKEN_ONLINE_DATUM: actuality of the last transaction for the segment Banks online
3. D19_BANKEN_OFFLINE_DATUM: actuality of the last transaction for the segment Banks offline
4. D19_VERSAND_ONLINE_DATUM: actuality of the last transaction for the segment mail-orden online
5. D19_TELKO_OFFLINE_DATUM: actuality of the last transaction for the segment telecommunication offline

The logistic regression yields the next 5 features (the response variable used is ONLINE_PURCHASE which is an imbalanced variable with only 9% purchasing online):

1. PRAEGENDE_JUGENDJAHRE: dominating movement in the person’s youth (avantgarde or mainstream)
2. SEMIO_ERL: affinity indicating in what way the person is eventful orientated
3. D19_VERSAND_ONLINE_DATUM: actuality of the last transaction for the segment mail-orden online
4. CJT_TYP_4: customer journey typology
5. FINANZ_VORSORGER: be prepared financial typology

Now that we know the 14 (two of them are repeated what is a good indicator) features that best explain our dataset, it is convenient to find linear correlations between them so that we can remove those highly correlated (removing redundancy in our analysis). In the table below we can see the correlation between the three first features detected by logistic regression and rest of selected feratures. The other eleven columns have been omitted in the table, but have been taken into acount.

Table 1. Pearson coefficient between selected features

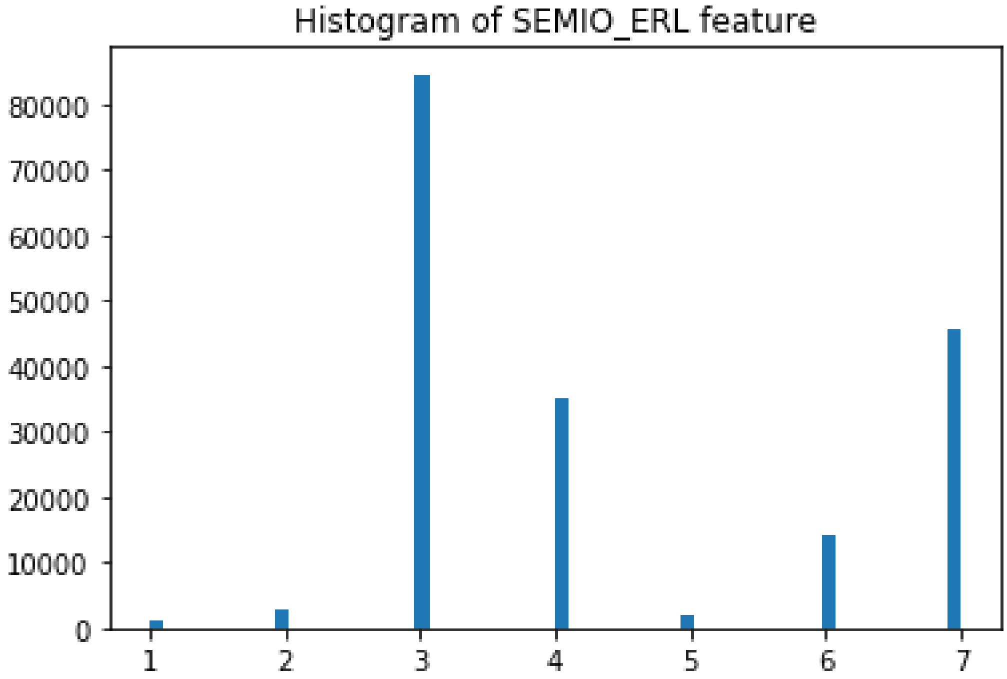
	PRAEGENDE_JUGENDJAHRE	SEMIO_ERL	CJT_TYP_4
PRAEGENDE_JUGENDJAHRE	1.000.000	0.213945	-0.703966
SEMIO_ERL	0.213945	1.000.000	-0.051468
CJT_TYP_4	-0.703966	-0.051468	1.000.000
FINANZ_VORSORGER	0.016930	0.391486	0.232804
D19_BANKEN_DATUM	-0.247157	-0.060567	0.165046
D19_BANKEN_ONLINE_DATUM	-0.209278	-0.046092	0.151132
D19_BANKEN_OFFLINE_DATUM	-0.082056	-0.028381	0.038170
D19_VERSAND_ONLINE_DATUM	-0.490006	-0.145910	0.332795
D19_TELKO_OFFLINE_DATUM	-0.136899	-0.040997	0.079605
FINANZ_ANLEGER	-0.331741	-0.368199	0.009809
D19_KONSUMTYP_MAX	-0.553435	-0.386178	0.237331
VERS_TYP	0.608190	0.479940	-0.276497
FINANZ_UNAUFFAELLIGER	-0.334220	-0.449360	0.035962
SEMIO_VERT	0.464251	0.066292	-0.163572

As we expected the selected features are not highly correlated (Pearson coefficient near 1) because PCA analysis is finding new variables that are orthogonal between them, that is, uncorrelated. The most correlated features are those related to finance, but any of them reaches 0.85 of correlation. The most correlated features with the others are VERS_TYPE and D19_KONSUMTYP_MAX with a 49% and a 48% respectively. For now all of this features will be kept for later analysis.

Exploratory analysis

The 70% of records with a higher affinity to events purchase online. There can be several explanations to these relatoinship. For example, a higher social activity is related with a higher desire of purchasing goods or with a need of. The x-axis of the histogram below goes from more affinity (1) to less (7).

Table 2 Histogram of SEMIO_ERL feature



A few descriptive statistics such as mean and standard deviation are listed below for each selected feature, mainly for those records that purchase online (which are the cluster under study). So, we first select all the recors that purchase online and then show the statistics for each choosen variable. Below we can see the first 5 columns:

Tabla 3 Descriptive statistics

	PRAEGENDE_JUGENDJAHRE	SEMIO_ERL	CJT_TYP_4	FINANZ_VORSORGER	D19_BANKEN_DATUM
count	16662.0	16662.0	16662.0	16662.0	16662.0
mean	6.0	4.0	4.0	4.0	9.0
std	5.0	2.0	1.0	1.0	2.0
min	0.0	1.0	1.0	1.0	1.0
25%	0.0	3.0	3.0	3.0	9.0
50%	6.0	4.0	4.0	4.0	10.0
75%	10.0	4.0	5.0	5.0	10.0
max	15.0	7.0	5.0	5.0	10.0

The rest of the table can be found in the jupyter notebook. We can use the table above to detect which are the characteristics of the records (in mean) that pruchase online. For example, those records interested in online-shopper advertising are online shoppers, which can intiutively be checked with marketing domain knowledge. To better understand the table above it is mandatory to look for the variables in the Excels spreedsheets.

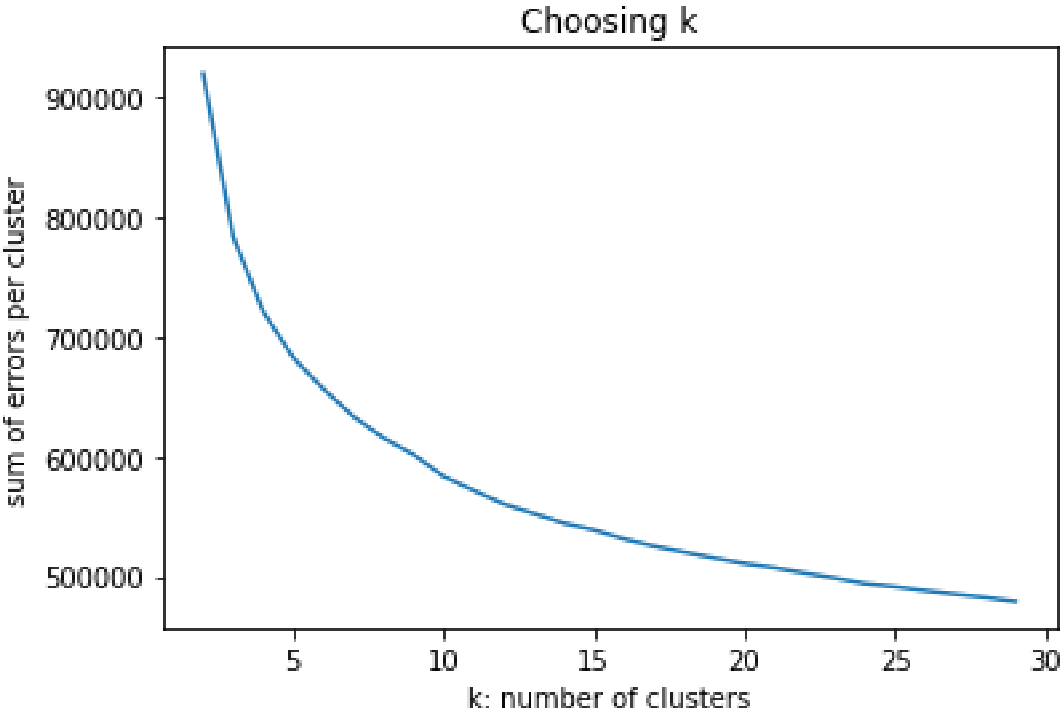
Methodology

Clustering

In this section we use k-means clustering algorithm to find groups (clusters) of similarity. This clusters will be later used to see how the extra columns CUSTOMER_GROUP, ONLINE_PURCHASE and PRODUCT_GROUP are distributed along the dataset. This three columns provide broad information of the customers in the CUSTOMERS dataset. After this, we will use this infromation to find relationships between citizens across germany, which can be useful to find targets for a campaign. This targets would be those citizens which are nearest to the cluster centers found before.

For the k-means algorithm to work faster and properly it is convenient to find the principal components first. With the three principal components that explain the most of the variance only the 40% of the total variance is explained, that is not enough to resume the whole dataset in this three components. With 50 components we can exlpain 91% of the variance and with 25 the 76%. Given that computing complexity is a problem that must be overcome, 25 components were choosed.

Tabla 4 Choosing the number of clusters, k



The above plot is very useful to choose the number of clusters, k. The optimus k is when above curve stops decreasing, that is, when the error can not be reduced by augmenting k. In this case, thirty number of clusters seems to be the optimus, but it would not be computationally possible. So, k = 5 was choosen not only for less computational complexity but also to gain in interpretability.

Once the clusters have been found, at least to things can be achieved. The first one is to find differences in the three columns listed above between clusters. In the next table you can found how the clusters are distributed along each of the classes of the columns CUSTOMER_GROUP, ONLINE_PURCHASE and PRODUCT_GROUP. Each of the rows represents a cluster (from 0 to 4) and each of the columns is a class: online pruchase have the classes 0 or 1 (op_0 and op_1), customer group have the classes multibuyer and singlebuyer (cg_multibuyer and cg_singlebuyer) and product group have the classes cosmetic, cosmetica_food and food (pg_cosmetic, pg_cosmetic_food and pg_food).

Tabla 5 Distribition of clusters

k	op_1	op_0	pg_cosmetic	pg_cosmetic_food	pg_food	cg_multibuyer	cg_singlebuyer
0	1.93	98.07	4.85	11.30	83.85	14.74	85.26
1	1.40	98.60	3.57	8.43	88.00	11.02	88.98
2	2.33	97.67	5.79	13.29	80.92	17.50	82.50
3	2.39	97.61	5.96	13.87	80.17	18.21	81.79
4	0.97	99.03	2.48	5.77	91.75	7.59	92.41

It can be seen that all the clusters don’t purchase online (only around 1.5% pruchases online) and that most are single buyers that prurchases food. If we sort the rows by cg_singlebuyer we find that also pg_food and op_0 are sorted, as we can see in the next table. That is, the most singlebuyer the less online pruchase (op_1) and the most pg_food.

Tabla 6 Last table ordered by singlebuyer

k	op_1	op_0	pg_cosmetic	pg_cosmetic_food	pg_food	cg_multibuyer	cb_singlebuyer
3	2.39	97.61	5.96	13.87	80.17	18.21	81.79
2	2.33	97.67	5.79	13.29	80.92	17.50	82.50
0	1.93	98.07	4.85	11.30	83.85	14.74	85.26
1	1.40	98.60	3.57	8.43	88.00	11.02	88.98
4	0.97	99.03	2.48	5.77	91.75	7.59	92.41

The second one is to find the citizens of germany (stored in the dataset AZDIAS) that are closest to the center of each cluster. For example, the 5000 citizens of germany that are closest to the cluster center number 3 would be good targets of a merketing campaign, since this cluster is the one with highest purchase online rate. The implemented code to do this can be seen in the jupyter notebook.

Binary Classification

A neural network will be trained in each of the clusters found. That is, the records of the train dataset will be classified into the closest cluster. After this, a model will be trained in each of the clusters, methodology that is found to achieve better results. The same methodology will be done with the logistic regression, so that the results of both models are comparable. A model willl be trained in the whole dataset so that it can be submitted to Kaggle. Stratified training, validation and testing (different from the one provided) subsets have been created.

Given that the train dataset is imbalanced in the RESPONSE variable, a little exploratory analysis is done in each of the clusters. The next table shows the percentage of 1's per cluster (C1), the distribution of 1's per cluster (C2) and the percentage of records per cluster.

Tabla 7 RESPONSE variable distribution per cluster

k	C1	C2	C3
0	1.25	16.18	16.14
1	1.31	34.87	33.04
2	1.18	16.18	17.10
3	1.43	24.28	21.08
4	0.84	8.48	12.64

The logistic regression have been found to perform really bad in the selected features. Another set of features was selected but this model was still unable to find relationships. Neither in the train clusters or in the whole dataset this model was able to overcome 0.51 of AUC (out of 1). Instead of using the AUC metric another method for treating imbalanced datasets was used. Records of the train dataset were removed until the number of 1's passed from 1% to 15%. The logistic regression was trained in the balanced dataset but the model was still as good as tossing a coin (0.5 probability of a new record to become a new customer or not).

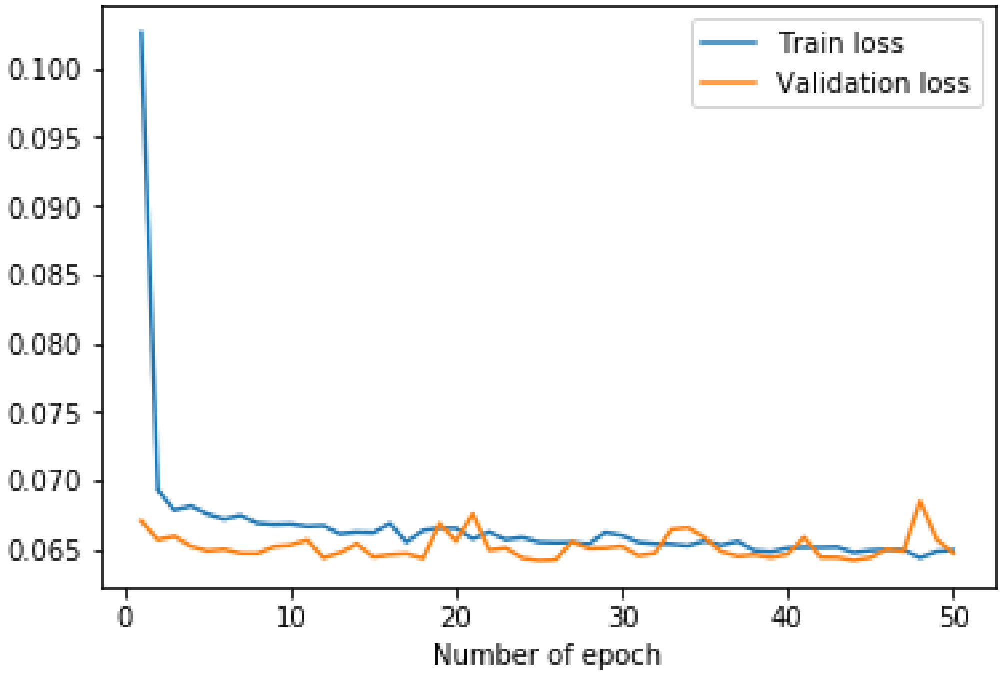
In the other hand, the neural networks trained perform much more better than the previous model. The next table is a review of the results for the different training datasets and network configurations. The different configurations means the number of hidden dimensions and the number of neurons per dimension because the input and output are the same, 14 and 1 respectively. Relu activation is used across the forward step (as well as dropout) to avoid non scaled numbers and overfitting respectively. Finally, the output of the neuron is the sigmoid of the last hidden layer and the cross entropy loss with Adam optimizer are used.

Tabla 8 Neutal network review

	Number Hidden dimension	Number of neurons per dimension	Number of epochs	Learning rate	AUC in the test dataset
Cluster_0	3	(100,100,100)	50	0.001	0.50
Cluster_1	3	(300,50,300)	50	0.0001	0.57
Cluster_2	3	(300,50,300)	50	0.0001	0.73
Cluster_3	3	(300,50,300)	50	0.0001	0.61
Cluster_4	3	(300,50,300)	50	0.0001	0.34
Whole Dataset	3	(300,50,300)	50	0.0001	0.68

As it can be seen in the previous table, the best results we get are for the whole dataset. In the plot below we can see how the training los decreases asthe network is trained (along the number of epoch). This plot is useful for detecting overfitting, which in this case did not occur.

Tabla 9 Training and validation loss



Conclussions and Improvements

It have been shown that a black box model perfoms better than white box model, because a fully connected neural network can achieved 0.68 AUC in this problem while logistic regression (benchmark model) in the best case obtains 0.51 AUC. These are the results for the selected variables, that could have been imporved. For example, two of the selected features are full of the unknown class (usually -1, 0 or 9), that should have been considered as missing values and thus not have been taken into account in futher analysis (deletion of this features). Another set of features could also have been considered.

A lot of different net configurations have been tested, from different number of hidden dimensions to different number of neurons. Increasing the number of dimensions was increasing the performance of the net, so three dimension net was finally choosen. However, an external pretrained neural network with more complex structure could be used for checking purposes, that is, tranfer learning.