CRISP-DM

1. Business Understanding

1.1. Business Objectives

- Customer Segmentation

  Identify the parts of the population that best describe the core customer base of the company

- Better results from marketing campaign

  Using the different features that define customer groups (from Customer Segmentation report), get to know individuals that are most likely to becoming into customers for the company in order to target them in the marketing campaign.

- Prepare a Supervised ML model to predict if an individual is likely to become a customer.

1.2. Derivables

- Prepare a Customer Segmentation report

  Unsupervised machine learning techniques to perform the customer segmentation

  Describe the relationship between the demographics of the company's existing customers and the general population of Germany.

  Describe parts of the general population that are more likely to be part of the mail-order company's main customer base, and which parts of the general population are less so.

  Main Derivable

  Arvato_Customer_Segmentation.ipynb - Customer Segmentation Report

  Additional files

1. *Arvato_EDA.ipynb* - Jupyter Notebook with EDA of AZDIAS and CUSTOMERS dataset where data preprocessing steps for Customer Segmentation are identified.
2. *data_preprocessing.py* - Python script that performs data cleaning and preprocessing for Cluster analysis on AZDIAS and CUSTOMERS.

- Supervised Learning Model - *Arvato_Supervised_Model.ipynb*

  Build a prediction model that uses demographic information from each individual to decide whether or not it will be worth it to include that person in a marketing campaign.

2. Data Understanding

2.1. Data Sources

Bertelsmann Arvato has provided the following 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).

Note there are terms and conditions for the use of this data. Terms and conditions are attached below.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/6759a017-034c-4091-9bd3-8093958aac47/terms.pdf

2.2. Data Description

- **Each row** of the demographics files represents a **single person**, but also includes information outside of individuals, including information about their **household, building, and neighborhood**.
- `Udacity_AZDIAS_052018.csv` and `Udacity_CUSTOMERS_052018.csv` will be used to figure out how customers ("CUSTOMERS") are similar to or differ from the general population at large ("AZDIAS") - **Customer Segmentation**
- `Udacity_MAILOUT_052018_TRAIN.csv` and `Udacity_MAILOUT_052018_TEST.csv` will be used to train and test the supervised learning model. (I will split `Udacity_MAILOUT_052018_TRAIN.csv` into train and validation sets.)

Additional Notes

- The "CUSTOMERS" file contains three extra columns ('CUSTOMER_GROUP', 'ONLINE_PURCHASE', and 'PRODUCT_GROUP'), which provide broad information about the customers depicted in the file.
- The original "MAILOUT" file included one additional column, "RESPONSE", which indicated whether or not each recipient became a customer of the company. For the "TRAIN" subset, this column has been retained, but in the "TEST" subset it has been removed; it is against that withheld column that your final predictions will be assessed in the Kaggle competition.
- Otherwise, all of the remaining columns are the same between the three data files. For more information about the columns depicted in the files, you can refer to the following spreadsheets:

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f2370eb3-f5ea-4aa7-9211-ec4c4a1bb9e1/DIAS_Information_Levels_-_Attributes_2017.xlsx
```

Top-level list of attributes and descriptions, organized by informational category.

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/23737ecb-8e25-4364-9544-c32d33a2a722/DIAS_Attributes_-_Values_2017.xlsx
```

Detailed mapping of data values for each feature in alphabetical order.

2.3. First EDA on AZDIAS data and CUSTOMERS data

I am gonna take the first glance at the data. I will probably need to do some cleanliness.

First important aspect noticed - I have +350 features → "High-dimensional" data, which is tricky for clustering analysis.

1. Create `attributes.csv` with all features. There are features in the .csv files that are not in the attributes list. It is important to identify unknown values that should be mapped to missing values. List of variables not included in attributes lists:
    - AKT_DAT_KL - Assumed ordinal (already encoded)
    - ALTERSKATEGORIE_FEIN - Let´s assume it is ordinal (already encoded) and that 0's are unknown values.
    - ALTER_KIND1, ALTER_KIND2, ALTER_KIND3, ALTER_KIND4 - Assume categorical but already encoded. hard to say.
    - ANZ_KINDER - Assume categorical but already encoded.
    - ANZ_STATISTISCHE_HAUSHALTE - Assumed numeric.

- ARBEIT - assume numeric. There is 1No. outlier.
- BIG_FLAP is in excel but not in customers.csv
- CJT_KATALOGNUTZER - Assume categorical already encoded.
- CJT_TYP_1 - CJT_TYP_6 - These variable seem to be related to CJT_KATALOGNUTZER. Maybe worth checking collinearity.
- CUSTOMER_GROUP - categorical. Needs to be encoded. Variable in CUSTOMERS only.
- D19_KONSUMTYP_MAX - Assumed similar to D19_KONSUMTYP.
- D19_KK_KUNDENTYP is in excel but not in customers.csv
- D19_LETZTER_KAUF_BRANCHE has variable names only. Can bem
- dropped.
- D19_SOZIALES - Assumed ordinal. 0´s unknown
- D19_TELKO_ONLINE_QUOTE_12 - Assumed ordinal. There seem to be lots of unknown so it can be dropped.
- D19_VERSI_DATUM - Assumed as other _DATUM variables.
- D19_VERSI_OFFLINE_DATUM - Assumed as other _DATUM variables.
- D19_VERSI_ONLINE_DATUM - Assumed as other _DATUM variables.
- D19_VERSI_ONLINE_QUOTE_12 - Assumed ordinal. There seem to be lots of unknown so it can be dropped.
- DSL_FLAG - No clue. Can be dropped.
- EINGEFUEGT_AM - This is datetime. Probably not useful.
- EINGEZOGENAM_HH_JAHR - Año retraido?? Son años para cada cliente pero no se muy bien de que.
- EXTSEL992 - Es una variable numérica pero no se que significa.
- FIRMENDICHTE - Assumed as ordinal and -1 unknowns.
- GEMEINDETYP - Assumed categorical.
- GEOSCORE_KLS7 is in excel but not in customers.csv

- HAUSHALTSSTRUKTUR is in excel but not in customers.csv
- HH_DELTA_FLAG - Categorical but no clue about meaning.
- KBA13_ANTG1 - KBA13_ANTG4 - Assumed ordinal. 0's unknowns.
- KBA13_BAUMAX - Assumed ordinal.
- KBA13_GBZ - Assumed ordinal.
- KBA13_HHZ - Assumed ordinal
- KBA13_KMH_210 - Assumed ordinal
- KK_KUNDENTYP - Assumed ordinal
- KOMBIALTER - Assume ordinal and 9's unknowns.
- KONSUMZELLE - Assumed categorical
- LNR - It seems to be a unnique ID. DOUBLE CHECK
- MOBI_RASTER - Assumed ordinal
- ONLINE_PURCHASE - Assumed categorical
- STRUKTURTYP - Assumed categorical
- PRODUCT_GROUP
- RT_KEIN_ANREIZ - Assumed ordinal

```
RT_SCHNAEPPCHEN;ordinal;[]
RT_UEBERGROESSE;ordinal;[0]
```

- UMFELD_ALT - Assumed ordinal
- UMFELD_JUNG - Assumed ordinal
- UNGLEICHENN_FLAG - Assumed cat
- VERDICHTUNGSRAUM (territorio) - Assumed cat and o's unknowns.

```
VHA;ordinal;[0]
VHN;ordinal;[0]
VK_DHT4A;categorical;[]
```

```
VK_DISTANZ;categorical;[
VK_ZG11;categorical;[]
```

- WACHSTUMSGEBIET_NB is in excel but not in .csv

Imp NOTES on variables:

- ANZ_HAUSHALTE_AKTIV - There are some weird values. Check value_counts()
- ANZ_PERSONEN - There are outliers. (Note tipically from 1-3)
- CAMEO_DEU_2015 - Maybe One-hot y luego PCa en one-hot.
- D19_ is transactional activity
- GEBURTSJAHR - year of birth. There are many 0 that are unknowns. I need to preprocessed these.
- KBA05_ is info about cars
- KBA05_ALTER1 - KBA05_ALTER4 - Share or car owners across ages. This is important when imputing missing values.
- KBA05_ANTG1 - KBA05_ANTG4 - number of family houses in the cell in categories. This is important when imputing missing values.
- KBA05_CCM1 - KBA05_CCM4mm - share of cars…categories. Seem to be exclusive information but not sure.
- KBA05_FRAU - This is just about female.
- KBA05_HERST1 - KBA05_HERST5 - share of different brand cars…categories.
- KBA05_KRSHERST1 - KBA05_KRSHERST3 are similar to KBA05_CCM*. Check collinearity.
- KBA05_KW1 - KBA05_KW3 - categories again.
- NOTE - Many of KBA05_variables might have collinearity.
- PLZ8_ANTG1 - PLZ8_ANTG4 might be correlated

2.4. EDA for Supervised Learning Model

`Udacity_MAILOUT_052018_TRAIN.csv` and `Udacity_MAILOUT_052018_TEST.csv` datasets.

3. Data Preparation

3.1. Data Preparation for Customer Segmentation

Data preprocessing steps were identified in `Arvato_EDA.ipynb` and then a python script `data_preprocessing.py` that contains cleaning and preprocessing functions was created to perform the data preprocessing of AZDIAS and CUSTOMERS datasets.

To summarise:

- Map unknown values to NaN's: An .csv file called `attributes.csv` was manually created because some of the unknown values in the variables given in the attributes information spreadsheets had not been converted to NaN's. By using these file, the map_unkwons( ) function convert these unknown values into NaN's.
- Drop features that are not useful due to its natures or the large proportion of missing values.
- Drop outliers
- Impute missing values using the 'Mode' or 'Most Frequent' value.
- Label encode categorical variables that had not been already encoded.
- Save preprocessed dataset into a pickle file that can be loaded in later for Cluster analysis.

3.2. Data Preparation for Supervised Learning Model for prediction

For the data preparation I reutilised the python script that I prepared for the customer segmentation part as the datasets are very similar. I needed to make some modification though.

File - supervised_model_data_preprocessing.py

Then, in the ML pipeline I scaled the data and, as I have class imbalance, I oversample the minority class using SMOTE and undersample the majority class.

4. Modeling

4.1. Approach for Customer Segmentation

Data preprocessing → PCA trained on AZDIAS population → Select components that explain 80% of the variability in the data→ Transform AZDIAS data and CUSTOMERS data → Cluster analysis using K-Means trained on PCA transformed AZDIAS data (fit_transform AZDIAS and transform CUSTOMERS) → Find under-represented and over-represented clusters.

4.2. Supervised Learning Model

I spot-checked the following models:

- RandomForestClassifier
- BalancedRandomForestClassifier
- XGBClassifier
- ADABoostClassifier with BalancedRandomForestClassifier as base estimator

For this, I used RepeatedStratifiedKFold and cross validation. The most promising model turned out to be XGBClassifier.

I ran GridSearchCV to fine-tune the parameters of the transformers in the pipeline and the hyper-parameters of XGBClassifier.

5. Evaluation (Supervised Learning Model Only)

5.1. Evaluation Metrics

Imbalance dataset

Area Under the ROC Curve (AUC) - Kaggle Competition

Source - https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

An **ROC curve** (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True positive rate (TPR): a synonym for recall.

$$TPR = \frac{TP}{TP + FN}$$

- False positive rate (FPR)

$$FPR = \frac{FP}{FP + TN}$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

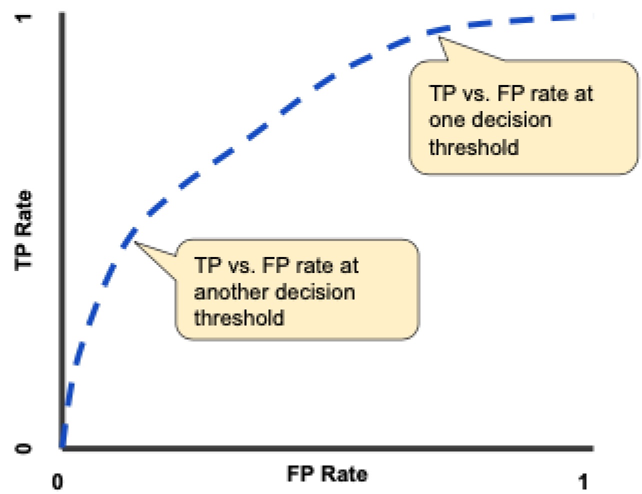

Figure 4. TP vs. FP rate at different classification thresholds.

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).

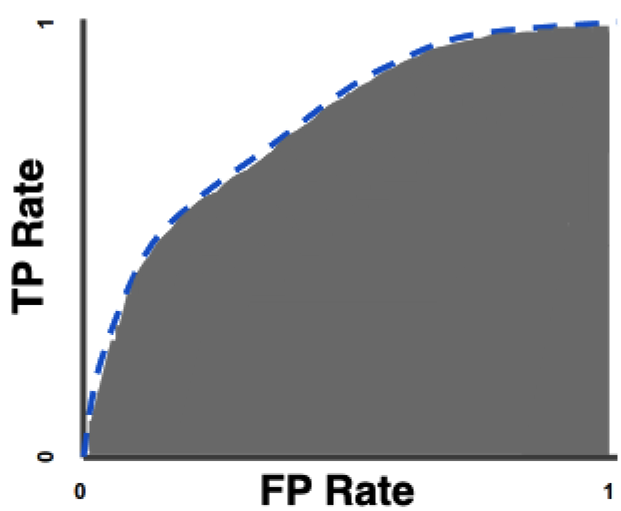

Figure 5. AUC (Area under the ROC Curve).

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as **the probability that**

the model ranks a random positive example more highly than a random negative example. For example, given the following examples, which are arranged from left to right in ascending order of logistic regression predictions:

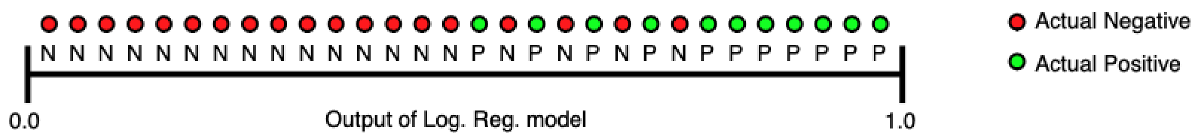

Figure 6. Predictions ranked in ascending order of logistic regression score.

AUC represents the probability that a random positive (green) example is positioned to the right of a random negative (red) example.

AUC ranges in value from 0 to 1. **A model whose predictions are 100% wrong has an AUC of 0.0**; one whose predictions are 100% correct has an AUC of 1.0.

AUC is desirable for the following two reasons:

- AUC is **scale-invariant**. It measures how well predictions are ranked, rather than their absolute values.
- AUC is **classification-threshold-invariant**. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

However, both these reasons come with caveats, which may limit the usefulness of AUC in certain use cases:

- **Scale invariance is not always desirable.** For example, sometimes we really do need well calibrated probability outputs, and AUC won’t tell us about that.
- **Classification-threshold invariance is not always desirable.** In cases where there are wide disparities in the cost of false negatives vs. false positives, it may be critical to minimize one type of classification error. For example, when doing email spam detection, you likely want to prioritize minimizing false positives (even if that results in a significant increase of false negatives). AUC isn't a useful metric for this type of optimization.

6. Deployment

Not applicable for this project.