Capstone Project: Create a Customer Segmentation Report for Arvato Financial Services

Udacity Data Scientist Nanodegree capstone project to create Customer Segmentation for Arvato Financial Services

1. Project Overview

In this project, we are provided with demographic data of customers of a mail-order company in Germany and demographic data of general population of Germany. Using this data, we are required to identify new customers for the company.

We approach this project in 2 phases:

- Use Unsupervised Learning to perform customer segmentation and identify clusters/segments from general population who best match mail-order company's customer base.
- Use Supervised Learning to identify targets for marketing campaign of the mail-order company who could possibly become their customers.

Goal of this project is to predict individuals who are most likely to become customers for a mail-order sales company in Germany.

Please find detailed overview of the project in blog post: https://medium.com/@shubham.divakar/data-scientist-capstone-create-a-customer-segmentation-report-for-arvato-financial-services-79a4f8f46f8e?postPublishedType=initial

In this project, you will analyze demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population. You'll use unsupervised learning techniques to perform customer segmentation, identifying the parts of the population that best describe the core customer base of the company. Then, you'll apply what you've learned on a third dataset with demographics information for targets of a marketing campaign for the company, and use a model to predict which individuals are most likely to convert into becoming customers for the company. The data that you will use has been provided by our partners at Bertelsmann Arvato Analytics, and represents a real-life data science task.

If you completed the first term of this program, you will be familiar with the first part of this project, from the unsupervised learning project. The versions of those two datasets used in this project will include many more features and has not been pre-cleaned. You are also free to choose whatever approach you'd like to analyzing the data rather than follow pre-determined steps. In your work on this project, make sure that you carefully document your steps and decisions, since your main deliverable for this project will be a blog post reporting your findings.

Part 0: Get to Know the Data

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 files represents a single person, but also includes information outside of individuals, including information about their household, building, and neighborhood. Use the information from the first two files to figure out how customers ("CUSTOMERS") are similar to or differ from the general population at large ("AZDIAS"), then use your analysis to make predictions on the other two files ("MAILOUT"), predicting which recipients are most likely to become a customer for the mail-order company.

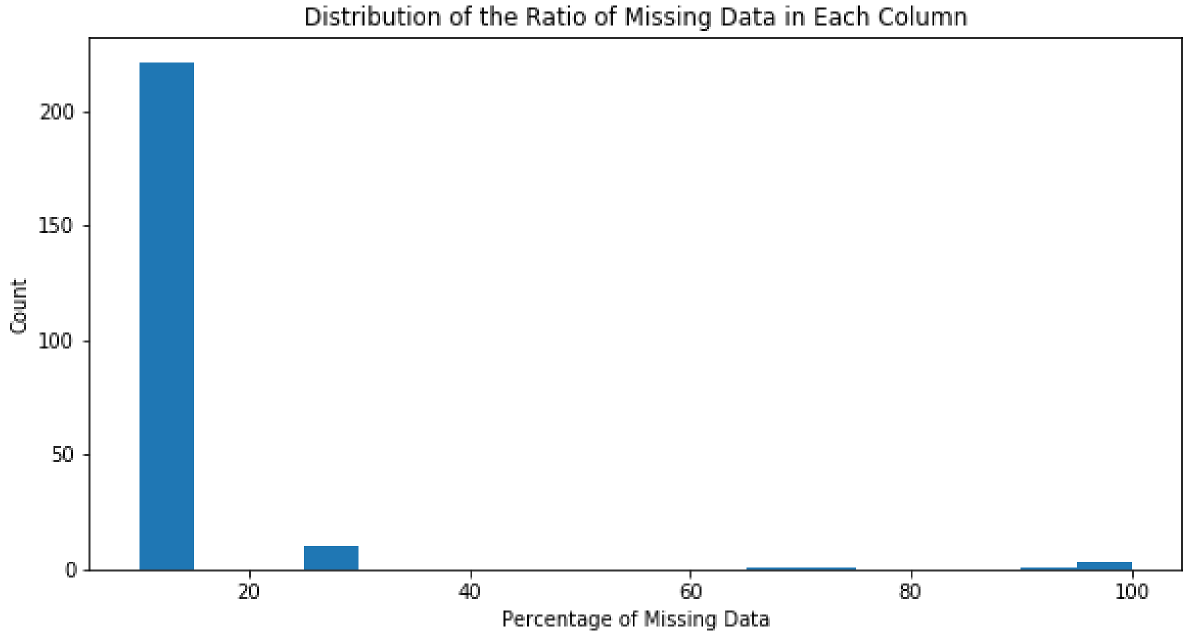
The "CUSTOMERS" file contains three extra columns ('CUSTOMER_GROUP', 'ONLINE_PURCHASE', and 'PRODUCT_GROUP'), which provide broad information about the customers depicted in the file. The original "MAILOUT" file included one additional column, "RESPONSE", which indicated whether or not each recipient 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.

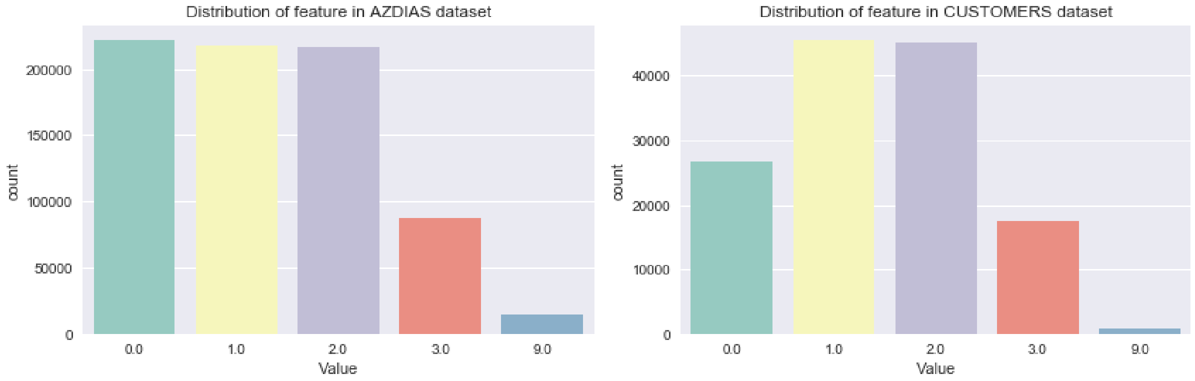
Otherwise, all of the remaining columns are the same between the three data files. For more information about the columns depicted in the files, you can refer to two Excel spreadsheets provided in the workspace. [One of them](./DIAS Information Levels - Attributes 2017.xlsx) is a top-level list of attributes and descriptions, organized by informational category. [The other](./DIAS Attributes - Values 2017.xlsx) is a detailed mapping of data values for each feature in alphabetical order.

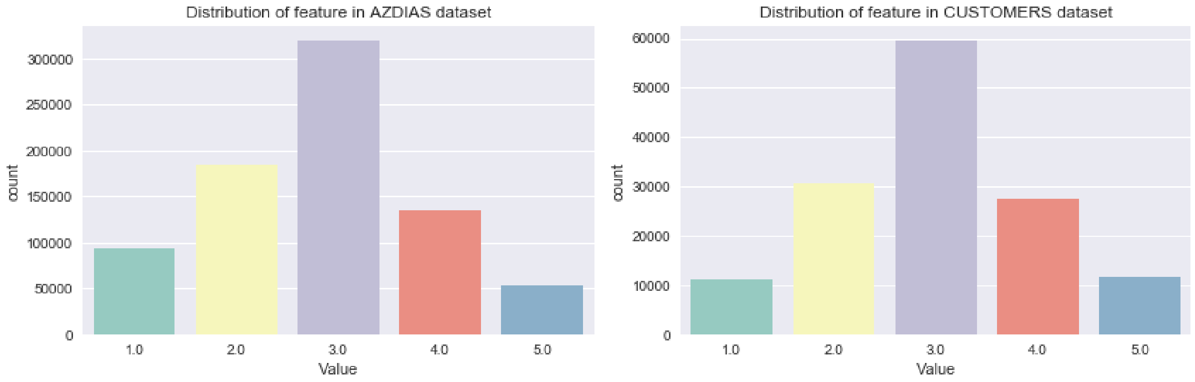
In the below cell, we've provided some initial code to load in the first two datasets. Note for all of the `.csv` data files in this project that they're semicolon (`;`) delimited, so an additional argument in the `read_csv()` call has been included to read in the data properly. Also, considering the size of the datasets, it may take some time for them to load completely.

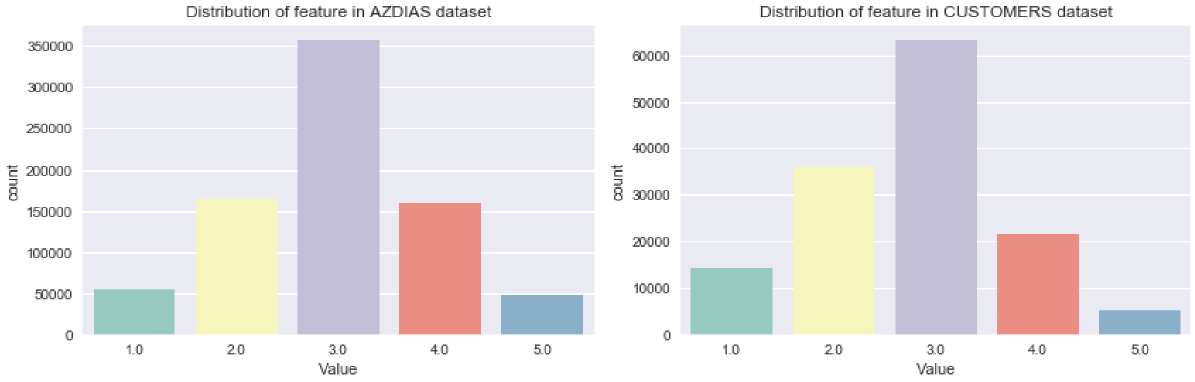
You'll notice when the data is loaded in that a warning message will immediately pop up. Before you really start digging into the modeling and analysis, you're going to need to perform some cleaning. Take some time to browse the structure of the data and look over the informational spreadsheets to understand the data values. Make some decisions on which features to keep, which features to drop, and if any revisions need to be made on data formats. It'll be a good idea to create a function with pre-processing steps, since you'll need to clean all of the datasets before you work with them.

One of the first steps in data exploration is to collect information about missing data in the dataset as this in many cases is one of the key element that some time get overlooked and affects ML model significantly. Identifying and understanding missing data information at earlier stage will help us come up with data processing tasks accordingly.









From analysis we have identified following cases where getting rid of these data helps us reduce noise and focus on our goal

a) Great 65% data is missing in 6 columns

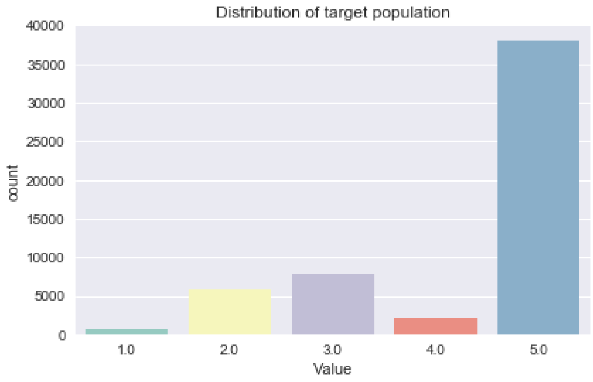
b) Customer dataset has some additional columns which are not part of Gen Pop data

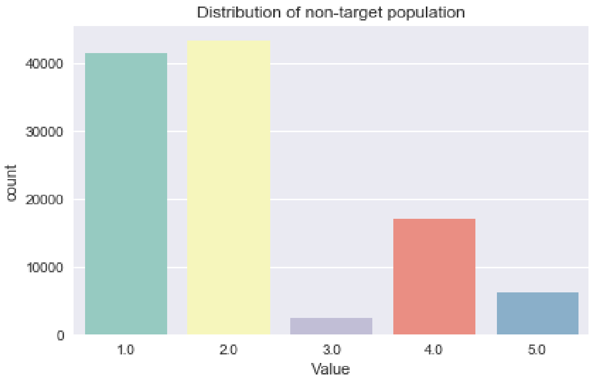
c) Columns that have many different items

d) Highly correlated columns

e) Fill-in missing values based on analysis of rest of data

f) With above operation, we got rid of highly irregular data. Now we focus on dealing with features that have some missing data in them. Replace 'NAN' with most frequent values — this is dealt with scikit's Imputer class.





Part 2: Supervised Learning Model

Now that you've found which parts of the population are more likely to be customers of the mail-order company, it's time to build a prediction model. Each of the rows in the "MAILOUT" data files represents an individual that was targeted for a mailout campaign. Ideally, we should be able to use the demographic information from each individual to decide whether or not it will be worth it to include that person in the campaign.

The "MAILOUT" data has been split into two approximately equal parts, each with almost 43 000 data rows. In this part, you can verify your model with the "TRAIN" partition, which includes a column, "RESPONSE", that states whether or not a person became a customer of the company following the campaign. In the next part, you'll need to create predictions on the "TEST" partition, where the "RESPONSE" column has been withheld.

```
Drop Null Rows and Columns Finished...
Feature Engineering: PRAEGENDE_JUGENDJAHRE Finished...
Feature Engineering: CAMEO_INTL_2015 Finished...
Feature Engineering: Numerical Columns Finished...
CPU times: user 22.1 s, sys: 3.1 s, total: 25.2 s
Wall time: 11.5 s
Unoptimized model
------
ROC score on testing data: 0.6516

Optimized Model
------
ROC score on testing data: 0.7489

------
{'learning_rate': 0.001, 'max_depth': 3, 'n_estimators': 2000}
CPU times: user 53min 26s, sys: 1min 22s, total: 54min 49s
Wall time: 5min 3s
```

After training a base model we can optimize our hyperparameter or the training method starting from there

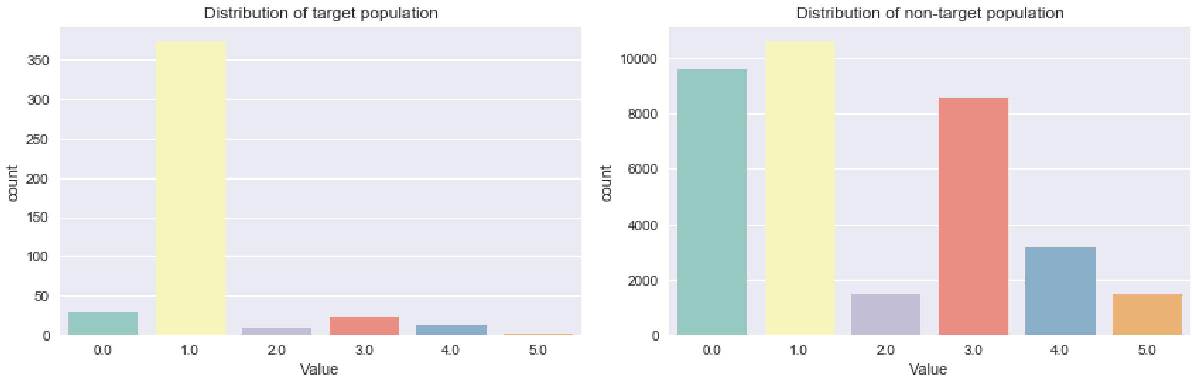
Part 3: Kaggle Competition

Now that you've created a model to predict which individuals are most likely to respond to a mailout campaign, it's time to test that model in competition through Kaggle. If you click on the link here, you'll be taken to the competition page where, if you have a Kaggle account, you can enter. If you're one of the top performers, you may have the chance to be contacted by a hiring manager from Arvato or Bertelsmann for an interview!

Your entry to the competition should be a CSV file with two columns. The first column should be a copy of "LNR", which acts as an ID number for each individual in the "TEST" partition. The second column, "RESPONSE", should be some measure of how likely each individual became a customer – this might not be a straightforward probability. As you should have found in Part 2, there is a large output class imbalance, where most individuals did not respond to the mailout. Thus, predicting individual classes and using accuracy does not seem to be an appropriate performance evaluation method. Instead, the competition will be using AUC to evaluate performance. The exact values of the "RESPONSE" column do not matter as much: only that the higher values try to capture as many of the actual customers as possible, early in the ROC curve sweep.

```
Training until validation scores don't improve for 1000 rounds.
[100]   valid_0's auc: 0.762504
[200]   valid_0's auc: 0.763831
[300]   valid_0's auc: 0.767755
[400]   valid_0's auc: 0.770131
[500]   valid_0's auc: 0.770063
[600]   valid_0's auc: 0.770046
[700]   valid_0's auc: 0.784615…………………………………..

Early stopping, best iteration is:
[1219]  valid_0's auc: 0.748503
The Train ROC Score is : 0.87341
```



There are still some other methods I want to try, for example I can combine the unsupervised learning and supervised learning to see if it can boost the performance. On the other hand, I can build up some other base model such as using neural network, support vector machine or even linear regression, and then using ensemble method to build some higher level stacking model.