Segmentation Report for Arvato Financial Services

Wenwen Huo Follow
Mar 2 · 4 min read

In this project, I analyzed the demographic data for customers of a mail-order sales company in Germany. The unsupervised learning technique was used to reduce dimensionality and identify the important demographic features that may contribute to online purchase. The supervised learning, or Gradient Boosting Classifier, was used next to fit the training data. Parameters were tuned according to accuracy score. The tuned model was then used to predict the test data. Final prediction was submitted to Kaggle competition.

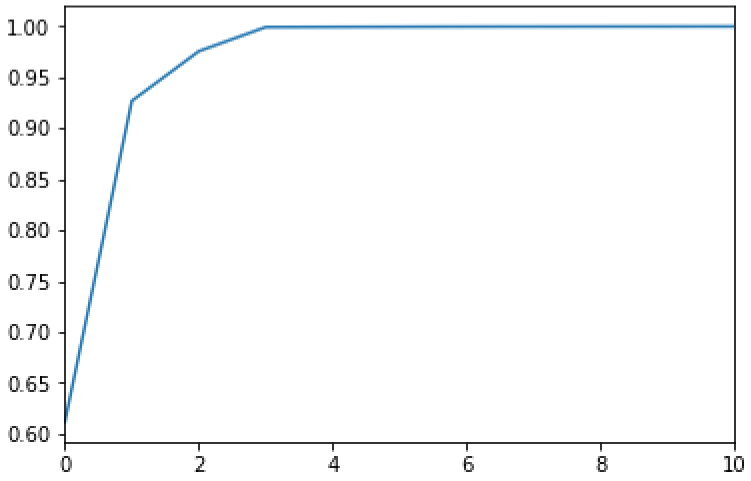
The major workflow in this project include:

1. Preprocessing data includes cleaning data and dimensionality reduction.
2. Unsupervised learning used to identify if customer with purchase can be separated from those without purchase using demographic features.
3. Supervised learning model with parameter tuning to make final predictions.

The input data was from Udacity capstone project: Arvato project. Two input files including azdias (total population) and customers (sub-population with purchase records) data were used for step 1 and step 2. Train and test data were used later for supervised learning tuning and prediction.

Here are some of my findings:

1 After PCA component analysis, the first four components explained more than 99% variance, among which the 1st component explained 60% of the variance in the azdias data.



Cumulative sum of PCA explained variance ratio. x-axis: first 10 components. y-axis: cumulative sum.

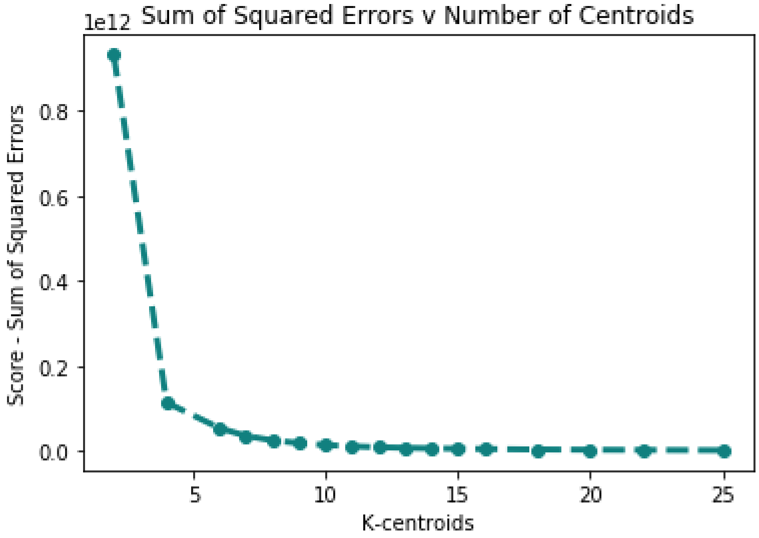
The top features contributing to the first components consist of household information such as transaction activities. Top 5 (by absolute value) features have negative weights:

a. “GEBURTSJAHR” with feature coefficient of -0.660639. Feature contains information of: year of birth. This makes intuitive sense to me since I anticipated a widespread range of year of birth in the population database.

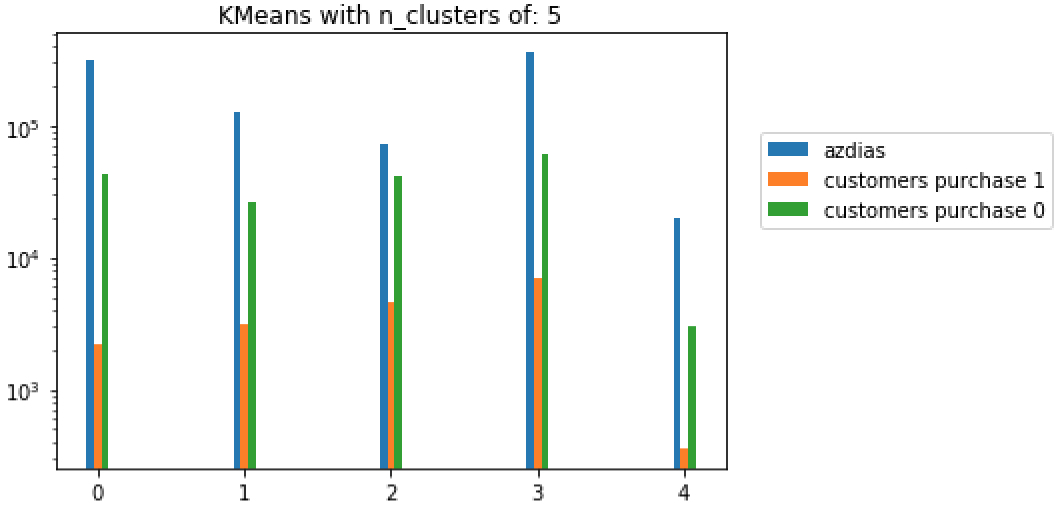
b. “EINGEFUEGT_AM_year” with feature coefficient of -0.440332. Note that this feature was engineered by converting EINGEFUEGT column into timestamps and obtaining the year value. Missing values were filled in with -1 (unknown). I could not find the information on the feature from the attributes excel sheet.

c. “MIN_GEBAEUDEJAHR” with feature coefficient of -0.440197. Feature contains information of: year the building was first mentioned.

2 By using kmean clustering analysis with 5 clusters, the difference between customers with purchase and customers without purchase can be visualized by their distribution. The customers with purchase are much less observed in cluster 4.



Choosing the number of centroids based on the ”elbow” rule.



azdias and customers data were clustered using Kmeans with 5 centroids.

3 Finally, when using supervised model to make predictions, I used pca transformed data with 4 components as input and built a Gradient Boosting Classifier model. I fine tuned

the parameters of n_estimators, learning rate and max depth to obtain the best accuracy of 0.838 training accuracy and 0.875 test accuracy. My Kaggle submission was ranked at 107, which was not too bad for my first try as a beginner.

By doing this project, I had a lot of exposure to data preprocessing. For the missing value, instead of imputing, I filled with unknown value -1. The reason behind this was due to the finding that original data uses -1 or 0 to indicate unknown values in a lot of columns. So filling in with -1 would keep the data consistent with the collected data. For data cleaning step, I had tried different ways to clean the categorical data, including one-hot encoding and multiple label encoding. I liked the multiple label encoding because the original dataset already had a vast amount of features. Adding more one-hot encoding would significantly expand the number of features. One feature that I had trouble cleaning was “EINGEFUEGT_AM”. There was no record of this feature in the attributes excel sheets and they look like time series so I turned them into three columns with year, month and date. Later on, I noticed maybe this feature indicates the age, or correlates with the “year of birth” feature. But I didn’t explore that correlation.

Another part I enjoyed was finding the best model. The model I was focusing on was Gradient Boosting Classifier. In the training dataset, the number of positive data (or RESPONSE=1) is so much less than the negative. It generates a problem where simply predicting everything as negative would give a pretty good score when fitting models. Hence, the models with default tuning didn’t work so well for my purpose. What I had done was to loop through one hyperparamer at a time, calculate the accuracy and pick the best hyperparamers based on the test accuracy, then move on to the next hyperparamers. This way, I was able to visualize overfitting and find the best model (among the ones I have tried).

In general, working on this project was a lot of fun. Even though there’s still a lot to improve, I think I’m pretty satisfied with the Kaggle ranking on the first try.