Machine Learning Engineer Nanodegree

Capstone Project

Mridul Gangwar
March 09, 2020

I. Definition

Project Overview

Finding new customers for a product is always a challenge. In an ideal world where you have a lot of funds available to acquire new customers, you can run promotions, discounts, online and offline campaigns to get new customers. However, when you have limited funds and staff hours available, you should narrow down your efforts to a target segment, which is more probable of buying your products.

"If I'm trying to expand sales, I have to find out who my existing customers are. What are their demographics? What do they look like?" Jerry Osteryoung, director of outreach for the Jim Moran Institute for Global Entrepreneurship at Florida State University. In this project, I will be creating a machine learning model to assist marketing team to target the right segment.

Problem Statement

In this project, I will be working on data provided by Arvato Bertelsmann. I will analyze demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population.

1) I will be using a clustering algorithm to create a segment of customers and identifying a similar group of people in the global population dataset. Since the characteristics of these two groups are the same, we can expect them to more probable to become customers.
2) The learnings from the previous comparison will be used to create a model to predict which individuals are most likely to convert in the marketing campaign.

Metrics

It’s a classification problem, whether the person will respond to the mail or not. I am expecting the data to be unbalanced as very few people react to the mail campaigns; the best metrics to be used are AUC/ROC and f1 score metrics. I will be using AUC to evaluate the models.

II. Analysis

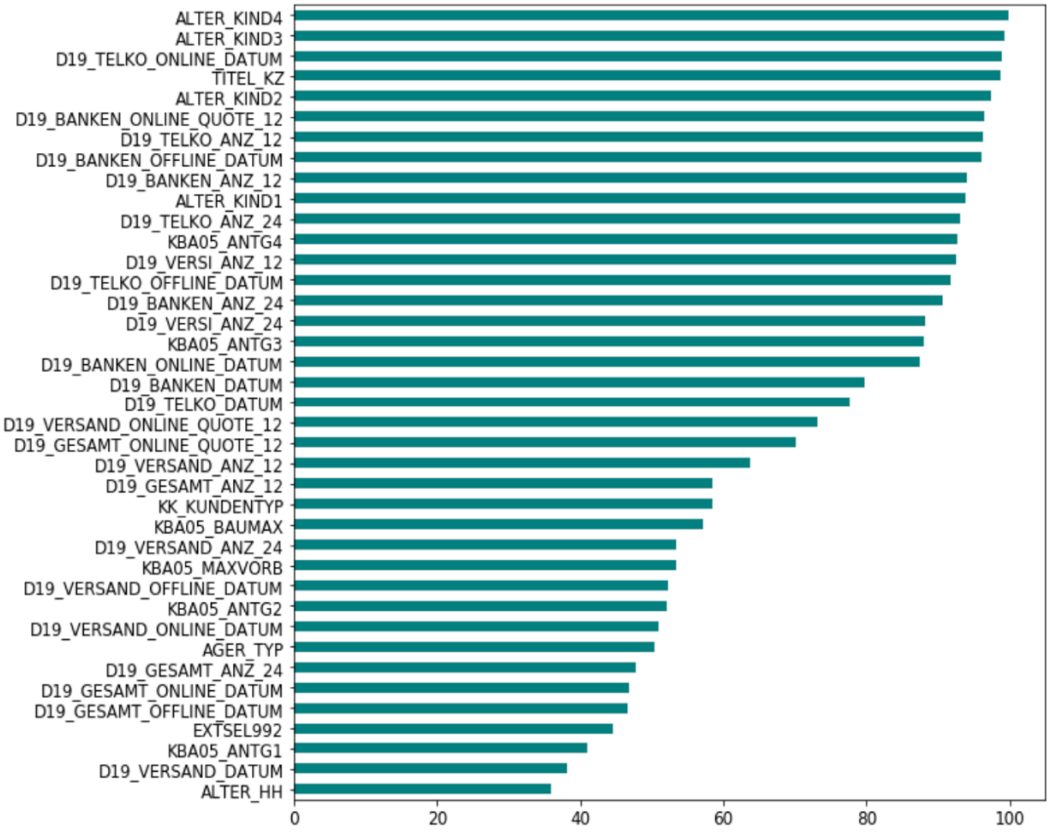
Data Exploration:

Six datasets included in this project:

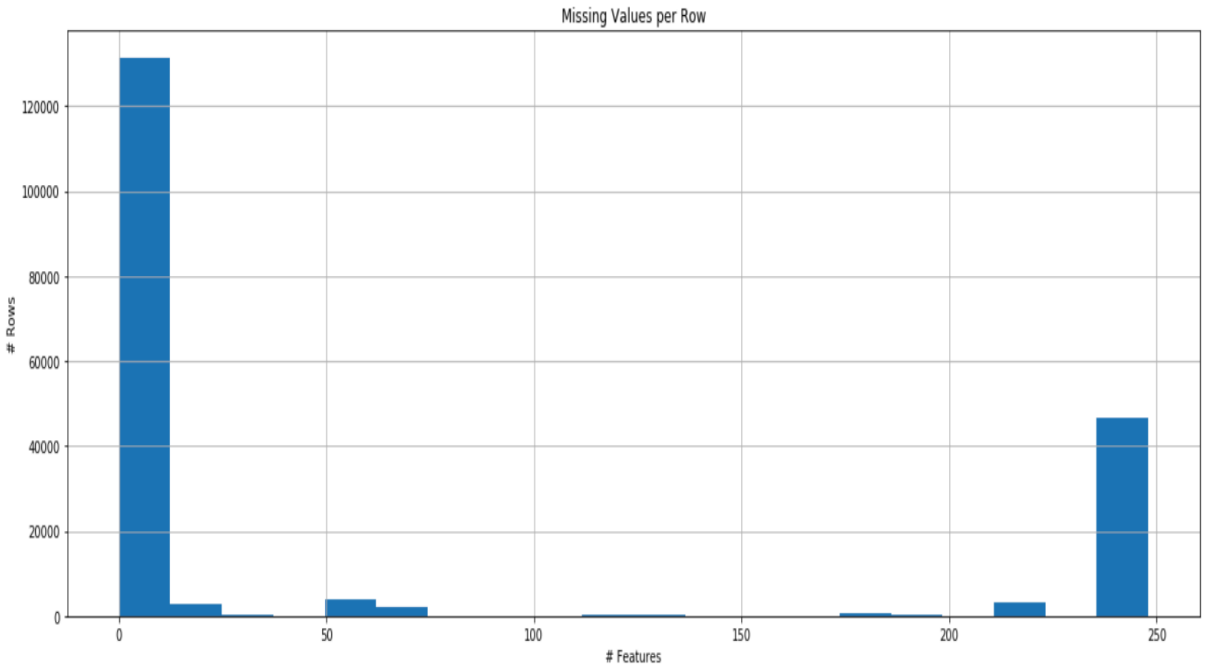
1) AZDIAS: demographics data for the general population of Germany
2) CUSTOMERS: demographics data for customers of a mail-order company
3) DIAS Attributes: detailed mapping of data values for each feature in alphabetical order
4) DIAS Information Level: top-level list of attributes and descriptions, organized by informational category
5) Udacity_MAILOUT_052018_TRAIN: This dataset contains all the demographic information of the targeted population and the response whether they responded to the mail or not. This dataset is used to train the machine learning model
6) Udacity_MAILOUT_052018_TEST.csv: This is the dataset used to generate prediction, which is submitted to the Kaggle competition

Steps performed in the data exploration:

1) I calculated the shape of the dataset. While working with azdias, I was getting timeout error or idle workspace error. That is why I decided to work with a sample of the azdias dataset. This sample is created by randomly selected 1/5 of the rows of the total azdias dataset
2) I calculated mean, minimum, different percentiles of all the numerical columns of both the azdias and customer datasets. Most of the columns have similar values and not deviated a lot.
3) After reading data, I realized the datatype in column 18 and 19 (i.e. 'CAMEO_DEUG_2015', 'CAMEO_INTL_2015') are of mixed type. From the DIAS Attributes dataset, I realized there are some garbage values in these two columns. Hence, I created a function to replace these garbage values with Nan.
4) Using DIAS Attributes dataset, I realized for most of the columns, some value represents null or unknown values. I identified those values for each column and then replaced those values with nan.
5) I visualized which columns have more than 40% of the rows as missing. I identified those columns and later deleted from further analysis. Below is the snippet for the customer's dataset:



6) Next, I visualized the distribution of the rows with the number of null values in them. Lat er, I deleted rows with more than 250 null values. Below is the snippet for the customer' s dataset:



7) Also, by manual inspecting DIAS Attributes dataset, identified a few features which could be used to represent data in a better way. For example, reducing the number of categories in order to better represent the data.

Benchmark

I chose a threshold of 40% to remove columns with null values more than the threshold. I prefe r this threshold as a small value will result in a loss of data, and a large value could result in som e garbage columns.

III. Methodology

Data Preprocessing

Based on the learnings from the Data exploration:

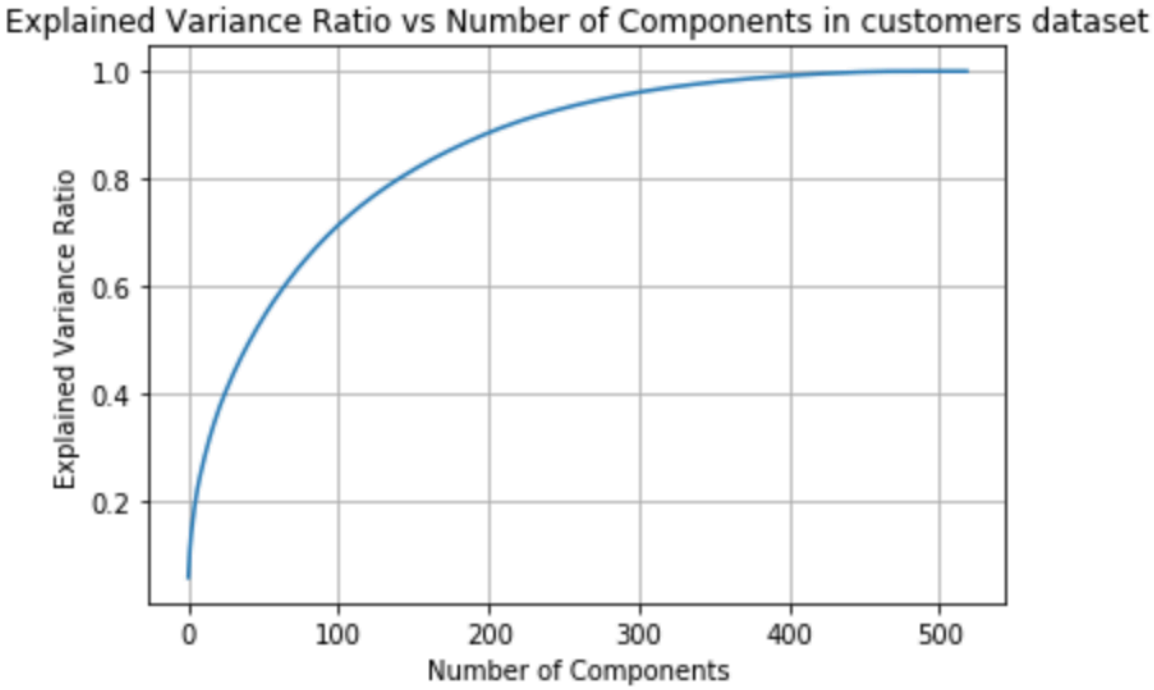
1) Created a function to handle the mix datatype columns.
2) Deleted columns with more than 40% of the values missing.
3) Deleted rows with more than 230 null values.
4) Created some columns based on the learning from manually inspecting DIAS Attributes dataset.
5) All the columns present in the dataset are categorical. However, if I perform One Hot encoding (OHE) on every single column, then my dimensionality of the dataset will explode. Also, I noticed most of the variables have some ranking in their meanings (like – Very high, High, Low, Very low, etc.). To prevent the dimensionality curse, I performed OHE only on those columns where categories don't have any ranking in them. The rest of the columns are treated as numerical columns.
6) After OHE, I performed imputation replacing null values with the most common value in the column. For imputation, I tried Simple Imputer, but I received an error "could not convert a string into float". So I created a function to replace null value with the most commonly occurring value in the column.

Implemented algorithms:

1) After imputation, the next step was to reduce dimensionality. By reducing the dimension of the data, we have a fewer relationship between variables to consider and are less likely to overfit the model and hence, accurate results.

   There are two ways to reduce dimensionality – features elimination and feature extraction. In feature elimination, we drop some variables and gain no information from those variables. However, in feature extraction techniques like Principal Component Analysis (PCA), we combine our input variables in a specific way, then we can drop the "least important" variables while still retaining the most valuable parts of all of the variables[1].

Below is the snippet of percentage of variance explained by different principal components in customers dataset:



I decided to keep 150 principal components, as they are explaining more than 80% of the data (which is big enough).

2) Identified columns which are contributing most to the First principal component (first principal component explained the highest variance in the dataset).

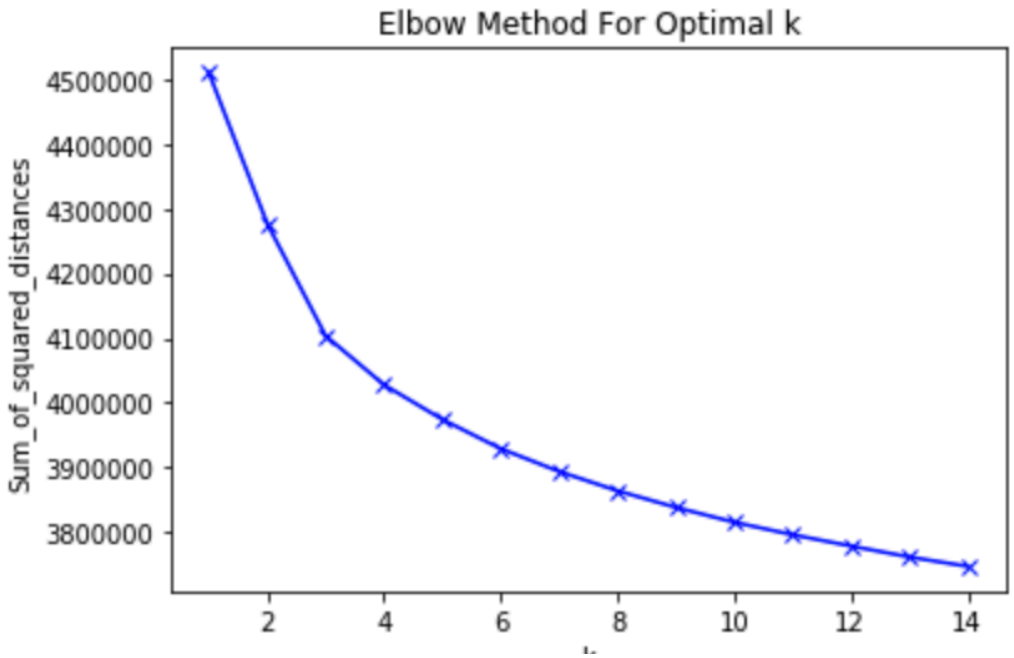
3) After applying PCA with 150 pc, the next step in the pipeline is clustering. We can use clustering to create segments of customers and then use that clustering algorithm to identify the similar segments in general population dataset. I used K-means algorithm to cluster the data. K-means algorithm is one of the simplest and popular unsupervised machine learning algorithms.

   K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. The process stops when there is no change in the value of centroids or number of iterations has been achieved[2]. One of the inputs to the k-mean clustering is how many

clusters you want to create. There are two ways (that I know of) to identify number of clusters – use subject matter expertise or use elbow method to identify the appropriate number of clusters.

The elbow method runs k-means clustering on the dataset for a range of values of k (say from 1-15) and then for each value of k computes an average score for all clusters. We can visualize this score by connecting score for each K. If the line chart resembles an arm, then the "elbow" is a good indication that the underlying model fits best at that point[3].

Below id the elbow graph for a range of clusters (K=1 to 15) for customers dataset:



Although there is no clear elbow, I am taking 10 clusters to be on the safe side.

4) Last part of the project is to create a supervise learning model on the mailout dataset. This dataset has output response from the targeted customer. I created a function which performs all the data cleaning steps explained in the previous section. After all the preprocessing I divided the dataset into train and test in a 75:25 ratio. I created a simple logistic regression model and trained it using a 5-fold cross validation.
Cross validation split the dataset into k groups. For each group, take the group as test dataset and fit the model on the remaining of the groups. Cross validation is a resampling procedure used to evaluate machine learning models on a limited data sample[4].

5) Since the data is unbalanced, I also used Synthetic Minority Oversampling Technique (SMOTE). SMOTE oversample the examples in the minority class by generating new

examples. Specifically, a random example from the minority class is first chosen. Then *k* of the nearest neighbors for that example are found (typically *k=5*). A randomly selected neighbor is chosen, and a synthetic example is created at a randomly selected point between the two examples in feature space[5].

IV. Results

There are two kinds of tasks performed in this project: unsupervised learning and supervised learning. In unsupervised learning, I used the k-means algorithm to identify the segments in the general population that looks similar to the customers. Below is the snippet of the clustering:

cluster	population	customer
0	11.572226	9.304628
1	10.689262	7.344110
2	12.581473	17.300582
3	13.981282	12.917198
4	9.299637	9.765550
5	5.569191	4.179638
6	6.201625	6.554773
7	11.329844	11.109132
8	12.055463	15.616473
9	6.719998	5.907915

Cluster 2,3,7 and 8 represent the biggest chunk for the customer dataset. The mail order company can target people in the same cluster from the general population. As general population of cluster 2,3,7 and 8 is similar in characteristics with the customers, chances are more they will respond to the promotion.

For supervised learning, I created different machine learning models. As we are trying to predict that the customer will respond to the offer or not, classification algorithms are used. Below is the summary of the performance of various models:

		Best Cross Validation Score	Test set score
1	Logistic Regression	0.67	0.50
2	Logistic Regression with SMOTE	0.89	0.55
3	XG-Boost with 5-fold CV	0.99	0.92
4	XG-Boost	-	0.55

5	Light GBM with 5-fold CV	0.99	0.54
6	Light GBM	-	0.74

Model 3,4,5 and 6 all are using SMOTE. Light GBM with 5-fold CV (model 5) is overfitting the dataset, as it is performing well in cross validation but failed poorly in test set.

V. Conclusion

Cluster algorithm and trained XG boost model could be stored and used to make inference in real time or in batch to assist the team.

Improvement:

1) Hyper tuning of models is not performed. I ran into memory issues while hyper tuning the parameters. Hyper tuning of models could improve the quality of predictions
2) We can create a few features based on the clustering result. For example, if a data point belongs to cluster to 2,3,7 or 8 then 1 or else 0.
3) An in-depth understanding of clusters might have helped in the generation of essential features.

Reference:

[1] – As One-Stop Shop for Principal Component Analysis, Matt Brems, https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c

[2] – Understanding K-means Clustering in Machine Learning, Dr. Michael J. Garbade https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

[3] – scikit learn documentation https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

[4]- A gentle introduction to k-fold cross-validation https://machinelearningmastery.com/k-fold-cross-validation/

[5] – SMOTE oversampling for imbalanced classification with python https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/