Arvato Capstone Project: Report

Wanli Pu

Abstract

This project is a data analytics study of demographics data for customers of a mail-order sales campaign in Germany. Data cleaning, preprocessing, transformation, unsupervised and supervised learning techniques were applied for this study. Unsupervised learning is used for customer segmentation purpose, and supervised modeling is used to predict the likelihood of individuals becoming customers for the mail-order campaign.

Background

The data for this project is provide by Bertelsmann Arvato Analytics, and represents a real-life data science task. Demographics data are provided for both the customer pool and the general population. The goal is to use the available data to perform customer segmentation and identify the parts of the population that best describe the core customer base of the company. Addition datasets provided to train a machine learning model which can predict which individuals are most likely to convert into customers for the company.

Datasets

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)

In addition, two more complimentary data information summary files associated with the project:

- DIAS Information Levels - Attributes 2017.xlsx: Attribute description for the four main data files
- DIAS Attributes - Values 2017.xlsx: Attribute value explanation for each attribute, and it could help for data preprocessing steps.

Problem Statement

There are two major problems to be solved. The first problem is to identify the parts of the population that best describe the core customer base of the company. The data from the ‘Udacity_AZDIAS_052018.csv’ and ‘Udacity_CUSTOMERS_052018.csv’ are the main source for the information.

The second problem is to train a machine learning model and predict which individuals are most likely to convert into customers for the company's mail-order campaign. The data from the 'Udacity_MAILOUT_052018_TRAIN.csv' and 'Udacity_MAILOUT_052018_TEST.csv' are the main source for the information.

In addition, the data from the 'DIAS Information Levels - Attributes 2017.xlsx' and 'DIAS Attributes - Values 2017.xlsx' are important resources to learn more information about the dataset and could help to process data.

Data Exploration Analysis

The general population data from 'Udacity_AZDIAS_052018.csv' contain 891,221 rows and 366 columns. Table 1 shows only a small portion of the data, and there are lots of missing values present.

Table 1: General Population Data

	LNR	AGER_TYP	AKT_DAT_KL	ALTER_HH	ALTER_KIND1	ALTER_KIND2	ALTER_KIND3	ALTER_KIND4	ALTERSKATEGORIE_
0	910215	-1	NaN	NaN	NaN	NaN	NaN	NaN	
1	910220	-1	9.0	0.0	NaN	NaN	NaN	NaN	
2	910225	-1	9.0	17.0	NaN	NaN	NaN	NaN	
3	910226	2	1.0	13.0	NaN	NaN	NaN	NaN	
4	910241	-1	1.0	20.0	NaN	NaN	NaN	NaN	

'DIAS Information Levels - Attributes 2017.xlsx' and 'DIAS Attributes - Values 2017.xlsx' could be used to better understand the general population data. Table 2 shows part of the 'DIAS Attributes - Values 2017.xlsx' information, and the meanings of each feature values are important and lots of data processing decisions should be based on this document.

Table 2: DIAS Attributes - Values 2017.xlsx

Attribute	Description	Value	Meaning
AGER_TYP	best-ager typology	-1	unknown
		0	no classification possible
		1	passive elderly
		2	cultural elderly
		3	experience-driven elderly
ALTERSKATEGORIE_GROB	age classification through prename analysis	-1, 0	unknown
		1	< 30 years
		2	30 - 45 years
		3	46 - 60 years
		4	> 60 years
		9	uniformly distributed
ALTER_HH	main age within the household	0	unknown / no main age detectable
		1	01.01.1895 bis 31.12.1899
		2	01.01.1900 bis 31.12.1904
		3	01.01.1905 bis 31.12.1909

The raw information from Table 2 is hardly applicable for large dataset processing. In order to facilitate automate data processing, a new file is constructed, 'AZDIAS_Attributes_Info.csv' as showed in Table 3. In this file, data types are assigned based on information provided in 'DIAS Information Levels - Attributes 2017.xlsx' and 'DIAS Attributes - Values 2017.xlsx', missing

values groups are compiled as a list, and several attributes contain values 10 and need to be converted to 0.

Table 3: AZDIAS_Attributes_Info.csv

attribute	type	missing_or_unknown	information_level	convert_10_to_0
AGER_TYP	categorical	[-1,0]	person	NaN
ALTERSKATEGORIE_GROB	ordinal	[-1,0,9]	person	NaN
ALTER_HH	interval	[0]	household	NaN
ANREDE_KZ	categorical	[-1,0]	person	NaN
ANZ_HAUSHALTE_AKTIV	numeric	[]	building	NaN
ANZ_HH_TITEL	numeric	[]	building	NaN
ANZ_PERSONEN	numeric	[]	household	NaN
ANZ_TITEL	numeric	[]	household	NaN
BALLRAUM	ordinal	[-1]	postcode	NaN
CAMEO_DEUG_2015	categorical	[-1,X]	microcell_rr4	NaN

Processing Missing Values and Select Features

Lots of missing values are not recognizable; first, all missing numbers and codes need to be converted to the right missing value codes. This step could be done with the help of 'AZDIAS_Attributes_Info.csv' files. Figure 1 shows the results after this step, and only a handful of attributes containing missing values more than 30%.

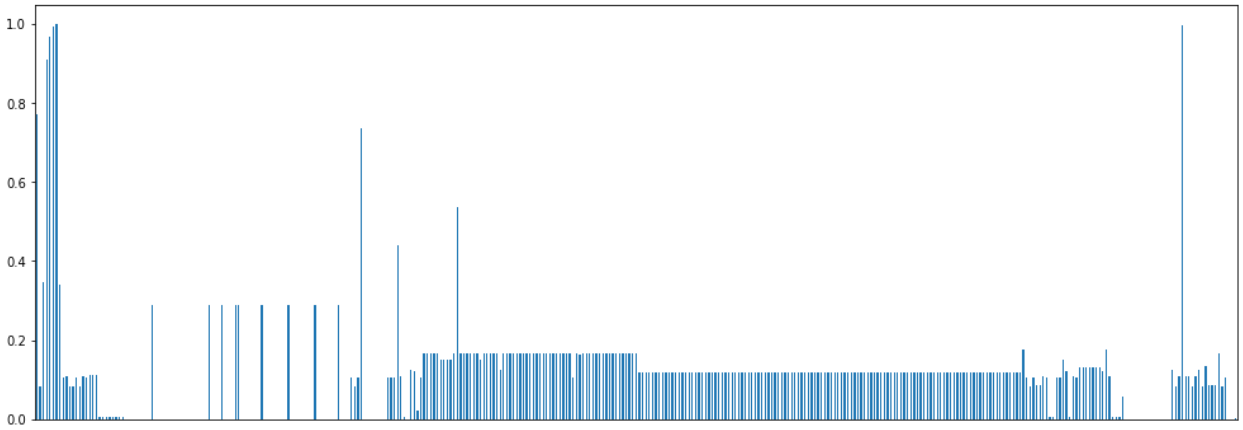

Figure 1: Missing Value Fractions for Each Attribute of General Population Data (Before cutoff)

Figure 2 shows the missing values fractions for each attribute after removing 11 attributes with more than 30% missing values.

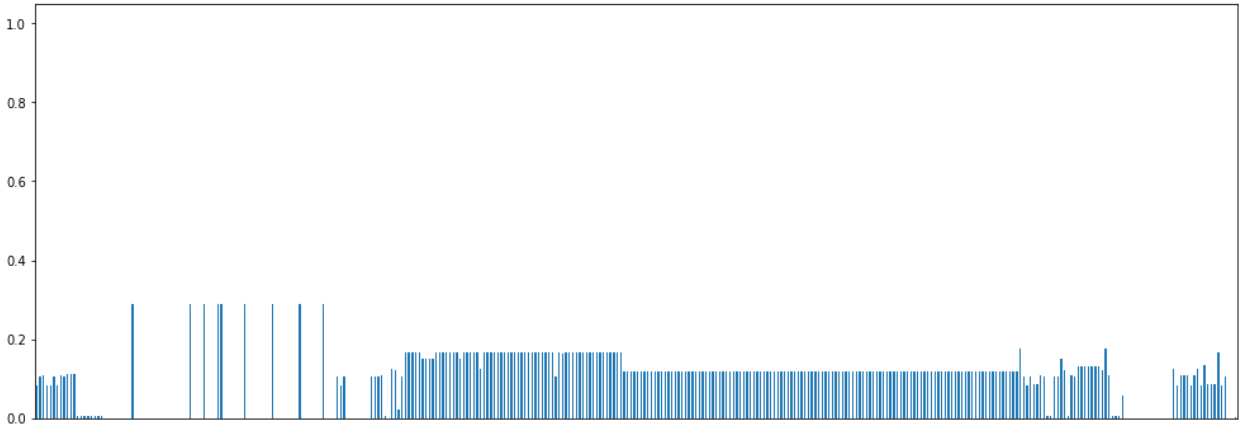

Figure 2: Missing Value Fractions for Each Attribute of General Population Data (After cutoff)

Re-encode Features

After selecting features with acceptable missing values counts, Figure 3 show data type counts for all features: 49 categorical features, 7 mixed typed features, 13 numerical features, 1 onehot feature and 283 ordinal features. I will go through feature processing steps for each feature type.

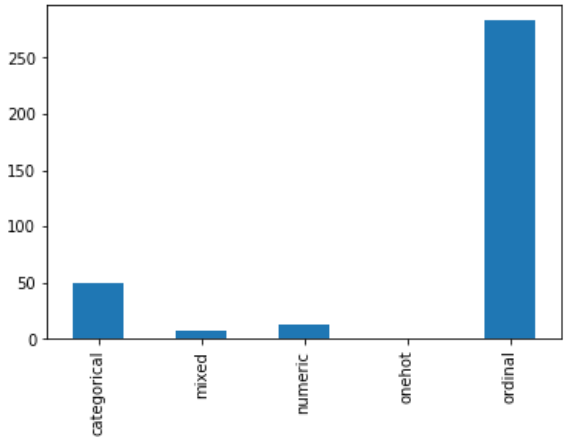

Figure 3: Data Types Bar chart

1. Categorical Features

For categorical features, lots of them are numerical coded, only 'CAMEO_DEU_2015' and 'OST_WEST_KZ' are coded with strings. 'CAMEO_DEU_2015' contains duplicated information with 'CAMEO_DEUG_2015', and 'CAMEO_DEU_2015' is dropped. 'OST_WEST_KZ' is re-encoded with binary numerical values. For all the remaining categorical features, if there is any missing value present, missing values are replaced with the most frequent values for the features.

2. Mixed-type Features

There are several mixed typed features, and they contain loaded properties, and need to be separated out help the analysis process. ‘CAMEO_INTL_2015’ combines both wealth and life stage. I decided to split this attribute into two features ‘CAMEO_INTL_2015_WEALTH’ and ‘CAMEO_INTL_2015_STATUS’. ‘PRAEGENDE_JUGENDJAHRE’ is handled similarly. For all the remaining mix-typed features, if there is any missing value present, missing values are replaced with the most frequent values for the features.

3. Ordinal Features

   All the ordinal features are numerically coded, and missing values are replaced with the most frequent values for the features. In addition, for several ordinal attributes, value 10 needs to be converted to 0 to make the value meaningful as ordinal variables.

4. Onehot Features

   Traditinally, categorical features need to be onehot-encoded; however, for this dataset, most of the categorical features are semi-oridinal, and we may come through without onehot encoding. Meanwhile, there is one feature ‘D19_LETZTER_KAUF_BRANCHE’ which is necessary to be onehot-encoded or dropped. I specify this one as onehot_feature for this project. Missing values are replaced with the most frequent values for the features.

5. Numerical Features

   Missing values are replaced with median value for the feature, and the resulted values for each feature are standardized to achieve consistent magnitude.

For this project, three feature encoding schemes was tested as showed in Table 4. For scheme #1, the onehot feature (D19_LETZTER_KAUF_BRANCHE) will be dropped, some information will be lost; for scheme #2, onehot feature and categorical features will be all onehot encoded, lots of additional sparse features will be added to the data matrix; for scheme #3, keep the categorical features the same, and onehot encode the onehot feature (D19_LETZTER_KAUF_BRANCHE).

Table 4: Feature Encoding Schemes

#	Encoding Scheme
1	Drop onehot features (only one)
2	Onehot encode all categorical features and onehot features
3	Only onehot encode onehot features

Customer Segmentation Analysis

Unsupervised clustering analysis was used to find segment among general population and describe the relationship between general population and customer pool. Before implementing

clustering analysis, Principal Component Analysis (PCA) was first applied to the dataset to rescue the dimensionality.

1. PCA Dimensionality Reduction

After previous data processing and encoding step, there are 585 features with encoding scheme #2, it's a good idea to apply PCA to analyze data variance and reduce feature dimensions while retaining most of information. After applying PCA, Figure 4 was plotted to show the cumulative explained variance against the number of components included. Based on Figure 4, if 150 components are kept, these 150 components could account for more than 90% of total explained variance.

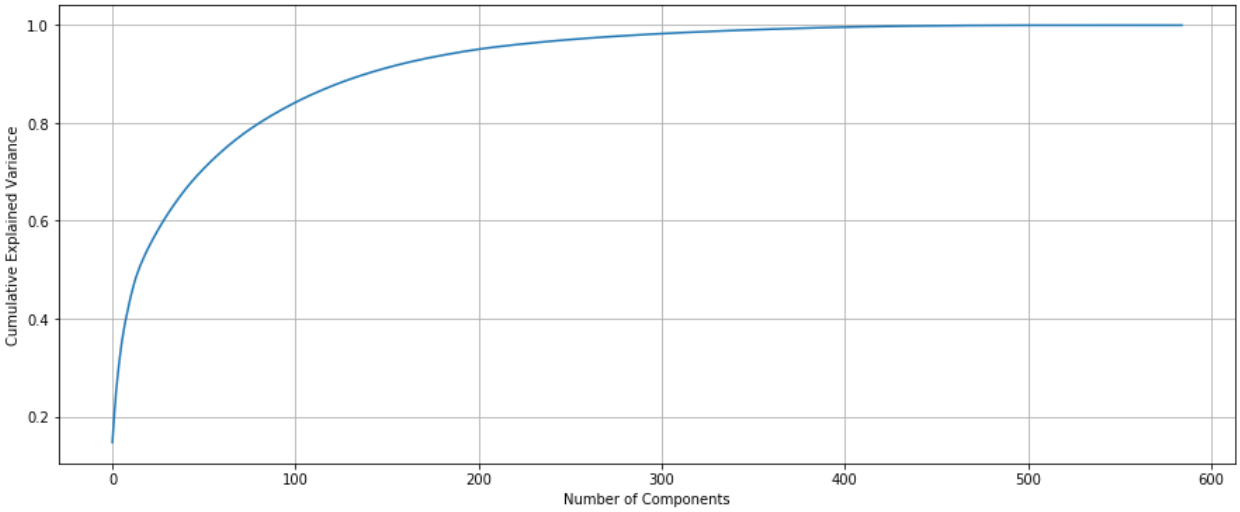

Figure 4: Cumulative Explained Variance

After choosing 150 as the magic number for the following data transformation, datasets are transformed to 150 components. For each of the top 5 principal components, top 6 contributors are list in Table 5.

Table 5: Top 6 Contributors for Principal Component #1-5

Principal Component # 1	contribution
D19_GESAMT_ONLINE_QUOTE_12	0.30514
D19_VERSAND_ONLINE_QUOTE_12	0.28691
D19_GESAMT_OFFLINE_DATUM	0.27746
D19_KONSUMTYP	0.25384
D19_VERSAND_OFFLINE_DATUM	0.25033
D19_VERSAND_DATUM	0.19096
Principal Component # 2	contribution
D19_VERSAND_ONLINE_QUOTE_12	0.41015
D19_GESAMT_ONLINE_QUOTE_12	0.40931
D19_GESAMT_DATUM	0.36253
D19_VERSAND_DATUM	0.35171
D19_GESAMT_ONLINE_DATUM	0.30124
D19_VERSAND_ONLINE_DATUM	0.26437
Principal Component # 3	contribution
ORTSGR_KLS9	0.21679
D19_GESAMT_ONLINE_DATUM	0.20214
D19_VERSAND_ONLINE_DATUM	0.17914
INNENSTADT	0.16417
D19_TELKO_DATUM	0.16037
D19_GESAMT_DATUM	0.15956
Principal Component # 4	contribution
ORTSGR_KLS9	0.25866
SEMIO_REL	0.20956
BALLRAUM	0.20188
INNENSTADT	0.19438
EWDICHTE	0.17748
SEMIO_ERL	0.1768
Principal Component # 5	contribution
KBA13_KMH_211	0.17682
KBA13_KMH_250	0.17432
KBA13_SEG_SPORTWAGEN	0.16296
KBA13_KW_121	0.16256
KBA13_CCM_2501	0.15899
KBA13_CCM_3001	0.1561

2. K-means Clustering Analysis

After transforming data into 150 principal components, the dimension of the data is reduced from 585 to 150 while retaining more than 90% of total variance. Next step is to apply k-means with sklearn package to cluster data in general population.

How to determine the number of clusters (k) is important, common practice is to try multiple k values for k-means clustering and calculate the clustering score for each k value. Figure 5 shows clustering scores against cluster numbers. The score of clustering is the sum of squared distances of points to nearest cluster center [1]. 10 was chosen as the number of clusters for following analysis.

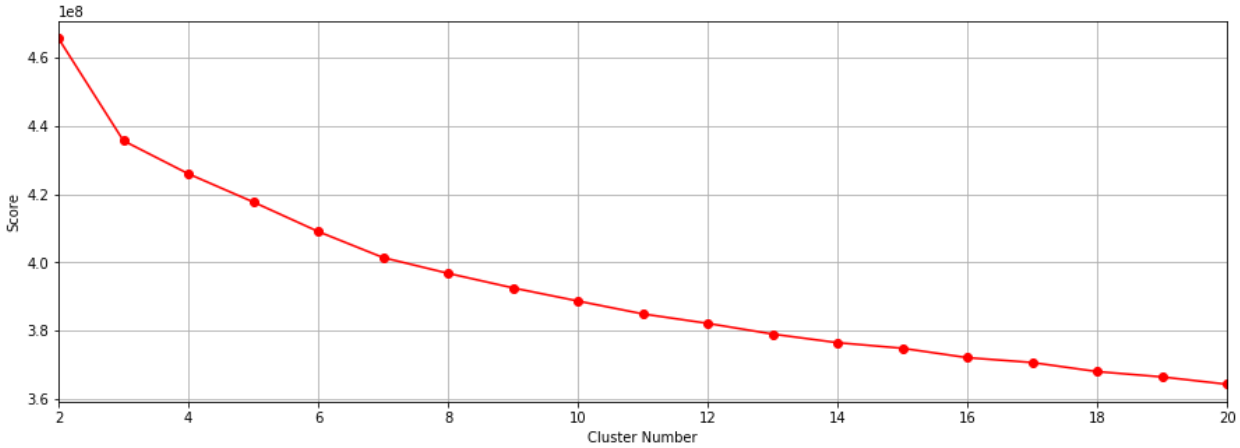

Figure 5: Clustering Score vs. Cluster Number

3. Compare general population data with customers data

Then the apply the same k-means model to customer data, and we have the cluster labels for both general population data and customer data. By calculating the cluster fraction, we could plot cluster fractions for general population data (Figure 6) and cluster fractions for customer data (Figure 7).

In addition, the fraction difference between general population and customer pool was plotted in Figure 8. Two clusters stand out, one is cluster # 8 and #5. Cluster #8 is the most over-represented cluster in customer pool compared to general population, while cluster #5 is the most under-represented cluster in customer pool compared to general population.

To further investigate the difference between cluster #8 and cluster #5. Figure 9-13 were plotted. Five features were used to illustrate the difference: `ONLINE_AFFINITAET`: online affinity, `INNENSTADT`: distance to city center, `EWDICHTE`: density of inhabitants per square kilometer, `ORTSGR_KLS9`: classified number of inhabitants, `SEMIO_REL`:affinity indicating in what way the person is religious.

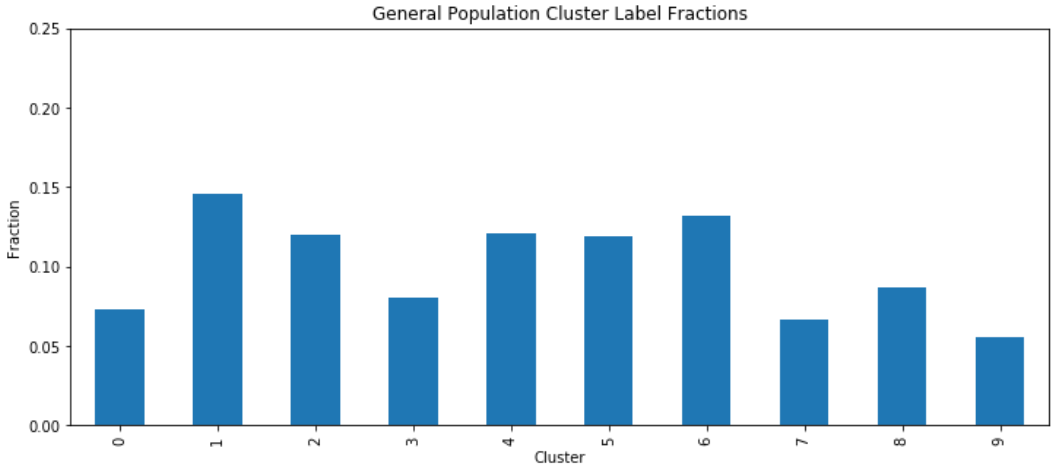

Figure 6: General Population Cluster Label Fractions

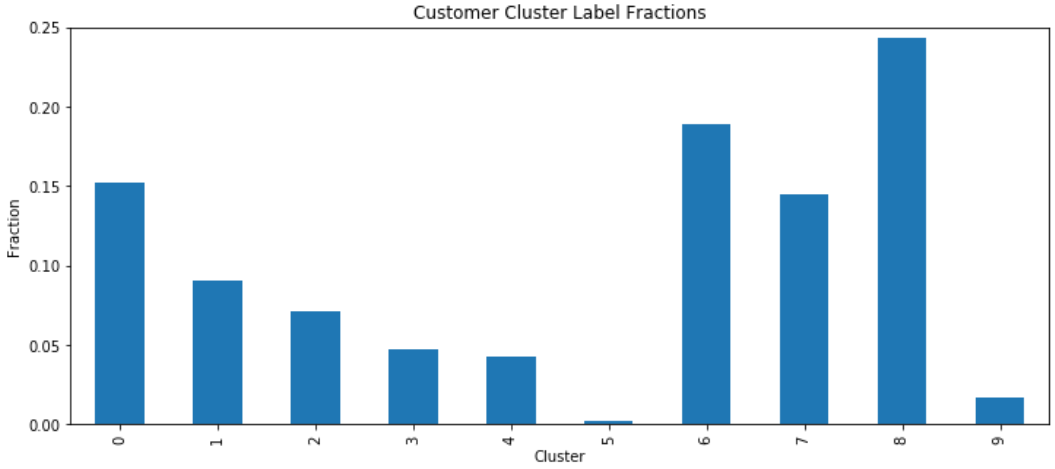

Figure 7: Customer Cluster Label Fractions

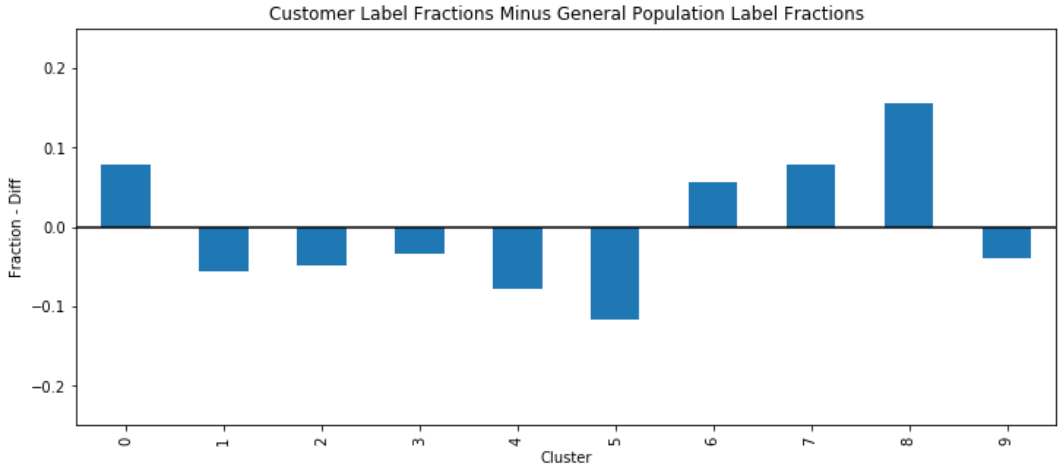

Figure 8: Customer Cluster Label Fractions Minus General Population Label Fraction

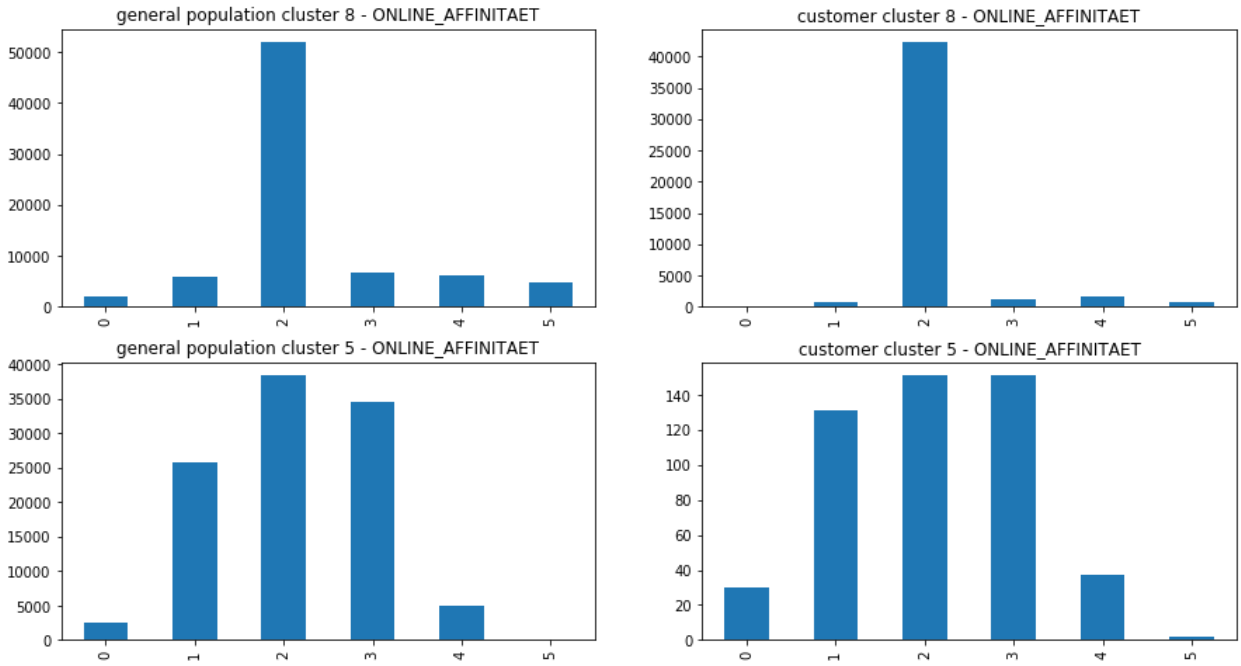

Figure 9: Cluster #8 and #5 comparison (ONLINE_AFFINITAET)

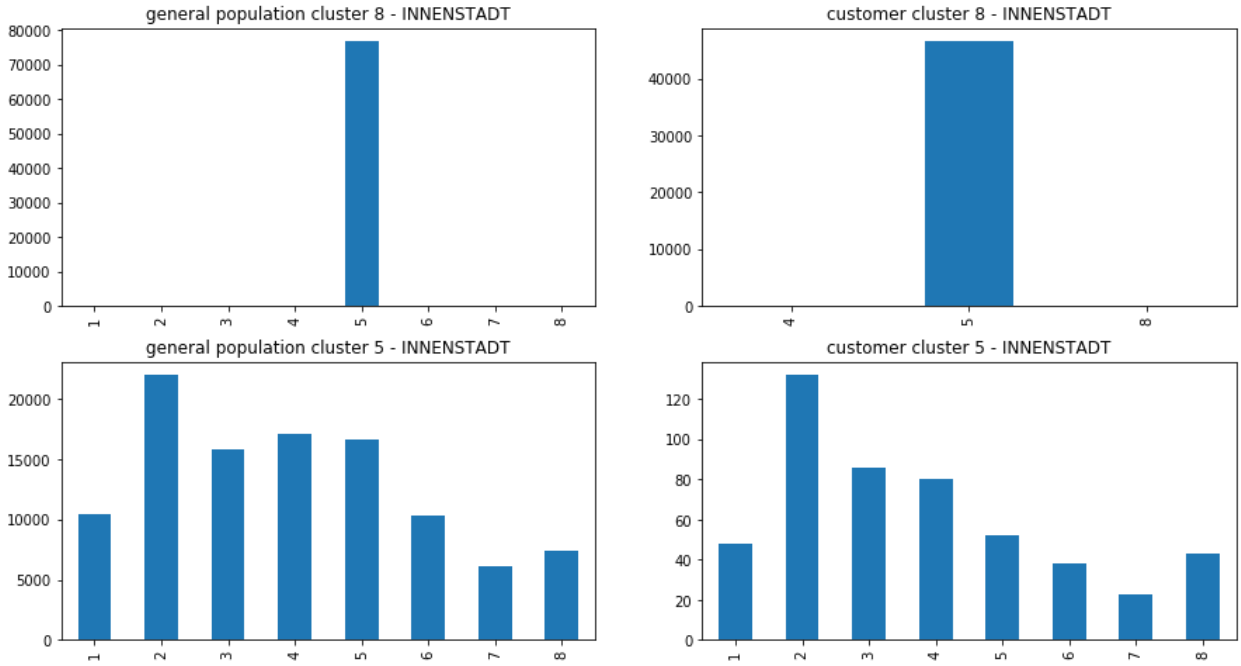

Figure 10: Cluster #8 and #5 comparison (INNENSTADT)

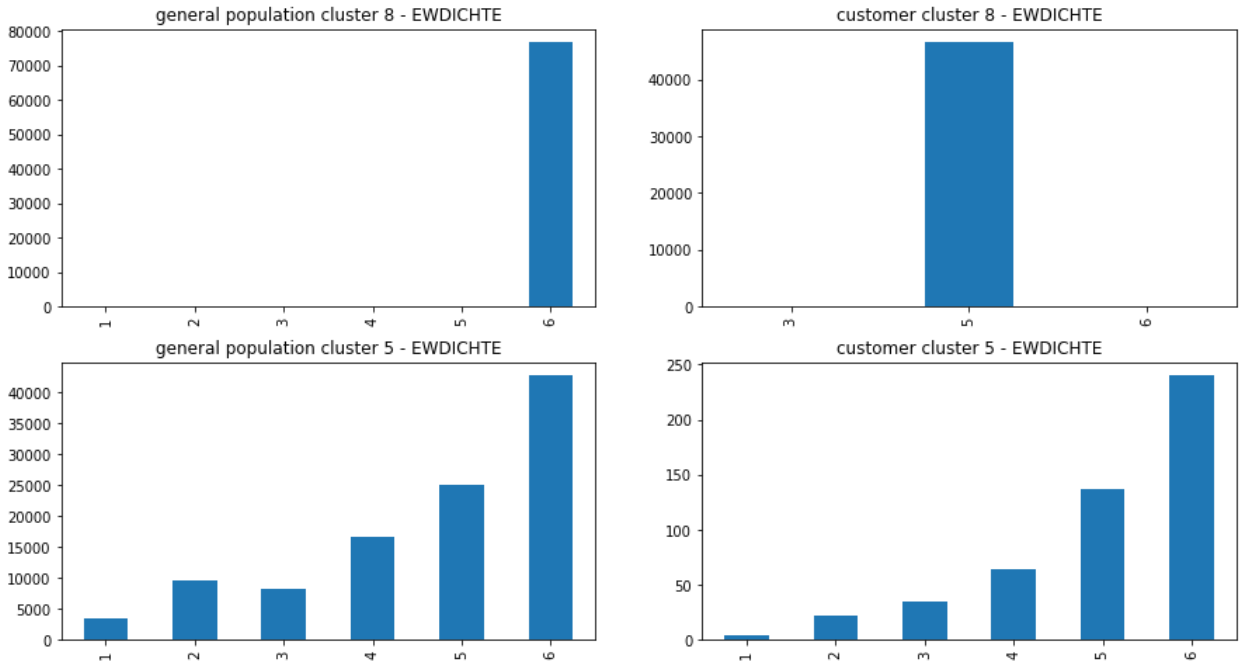

Figure 11: Cluster #8 and #5 comparison (EWDICHTE)

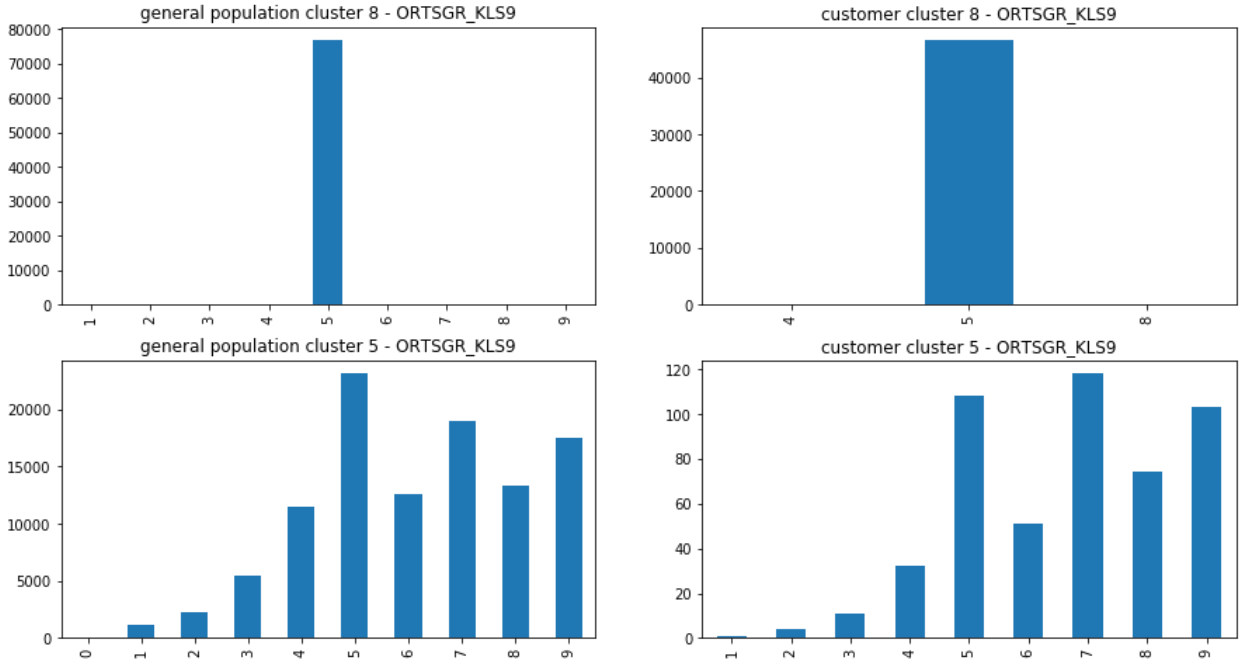

Figure 12: Cluster #8 and #5 comparison (ORTSGR_KLS9)

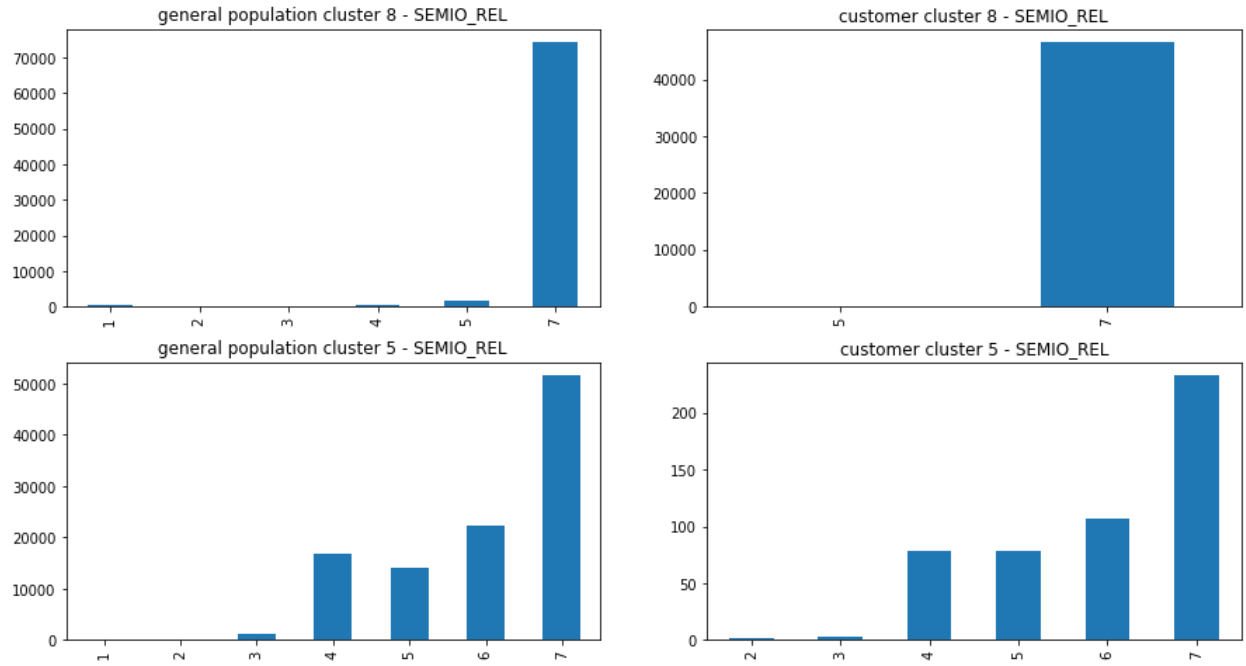

Figure 13: Cluster #8 and #5 comparison (SEMIO_REL)

Supervised Learning Modelling

Now that we have 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; training data will be used to train predictive model and verify the model, while testing data will be used to generate prediction and submitted to kaggle competition.

1. Data Processing

   Same data processing steps (imputing, normalization, encoding) were applied to the training data and testing data.

   However, the label distribution for ‘RESPONSE’ is imbalanced, and the majority of the training cases are ‘0’s (Figure 14). To ease the severe imbalance issue, positive samples (‘1’s) were resampled without replacement, and now the ratio between ‘0’s and ‘1’s is 10:1 (Figure 15).

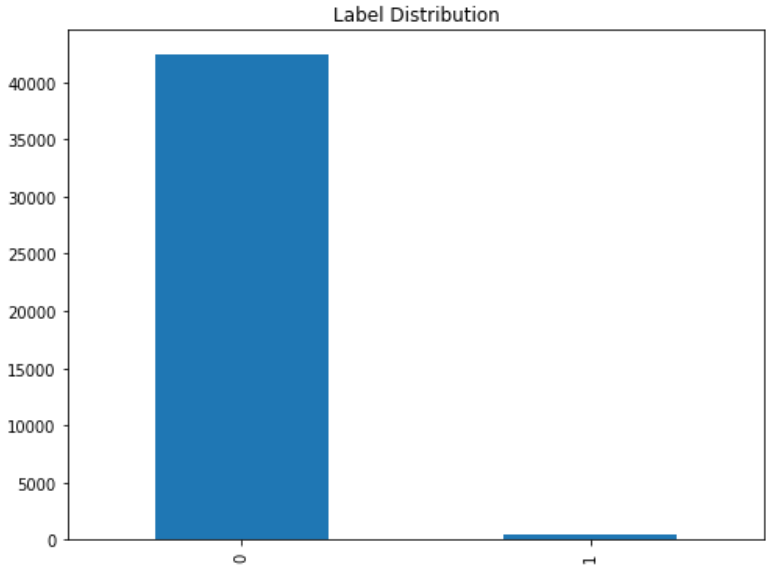

Figure 14: Training Label Distribution

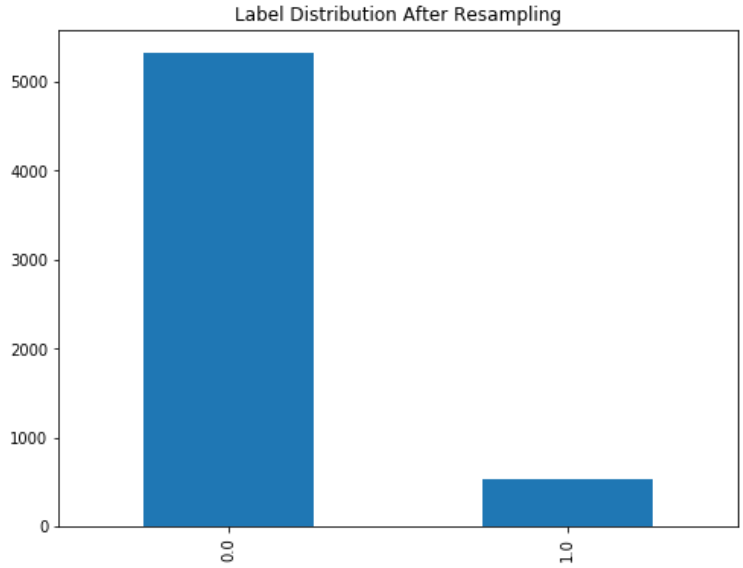

Figure 15: Training Label Distribution After Resampling

2. Metrics

For binary classification problems, there are certain number of true positive samples and certain number of true negative samples, and they are called ground truth. A trained binary predictive model will predict positive or negative for every sample, and now we have certain number of predicted positive samples and certain number of predicted negative samples. With ground-truth and predicted values, there are four possibilities for each sample:

a. True Positive: the model predicts positive and it's true
b. True Negative: the model predicts negative and it's true
c. False Positive (Type I Error): the model predicts positive but it's false (negative)
d. False Negative (Type II Error): the model predicts negative but it's false (positive)

With these four possibilities, a confusion matrix could be put together as Figure 16 [2], and a pregnancy analogy is used to help the understanding of the concepts.

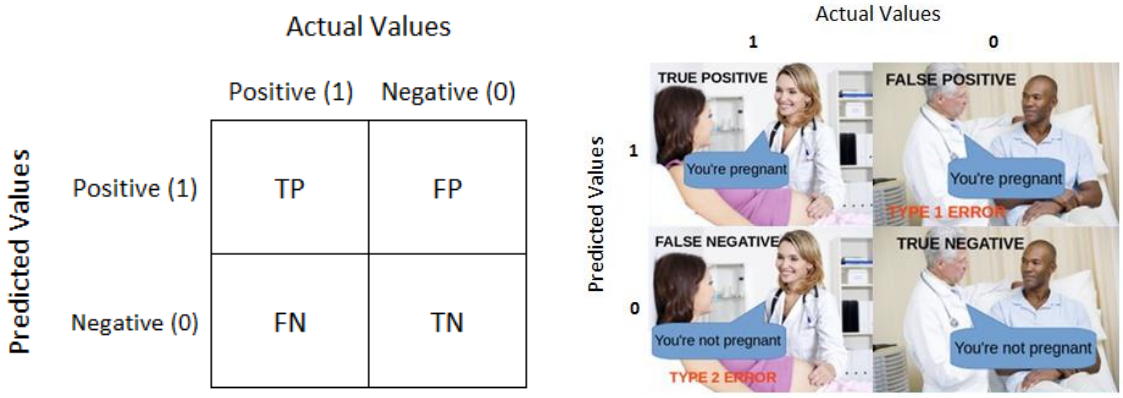

Figure 16: Confusion Matrix [2]

A graph plotting TPR against FPR is called ROC curve. Here are the definition of TPR and FPR.

$$TPR\ (True\ Positive\ Rate, Recall, Sensitivity) = \frac{TP}{TP + FN} = \frac{TP}{Positive(Actual\ Values)}$$

$$FPR\ (False\ Positive\ Rate) = \frac{FP}{FP + TN} = \frac{FP}{Negative(Actual\ Values)} = 1 - \frac{TN}{FP + TN}$$

For a specific model, TPR and FPR could be calculated with a range of different threshold and then a ROC curve could be generated with these values as in Figure 17 [3]. The area under the ROC curve is called AUC-ROC score, and the perfect score is 1. For this project, the corresponding Kaggle auc-roc score is the evaluation metrics.

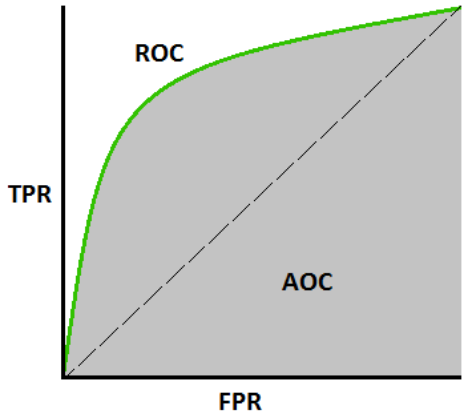

Figure 17: AUC-ROC [3]

3. Baseline Model

For typical binary classification problems, a vanilla logistic regression model could serve as a good benchmark. 5-fold cross validation was used to investigate logistic regression model (Figure 18); average roc_auc score is 0.897 (training) and 0.7126 (cross-validation). With this model, the kaggle public score is 0.73018 (Table 7)

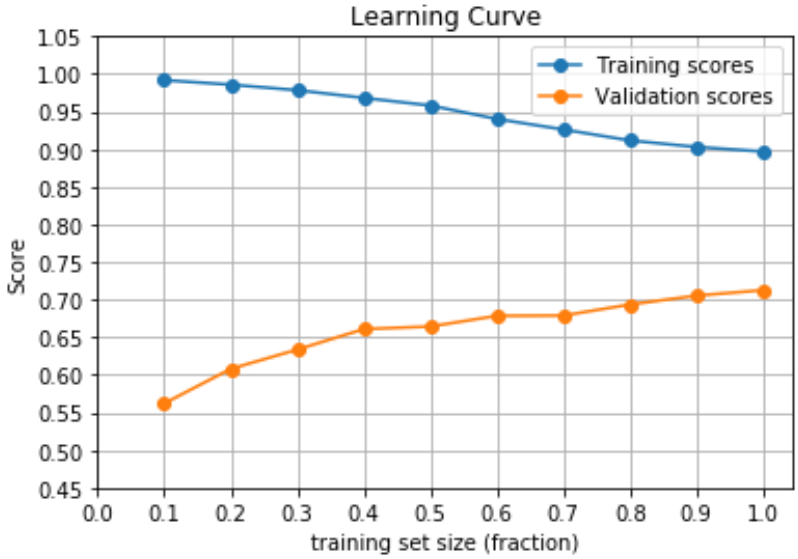

Figure 18: Logistic Regression Training Curve

4. Gradient Boosting Classifier and Refinement

Gradient boosting classifier with default hyperparameters and 5-fold cross validation was used as well (Figure 19); average roc_auc score is 0.9213 (training) and 0.7535 (cross-validation).

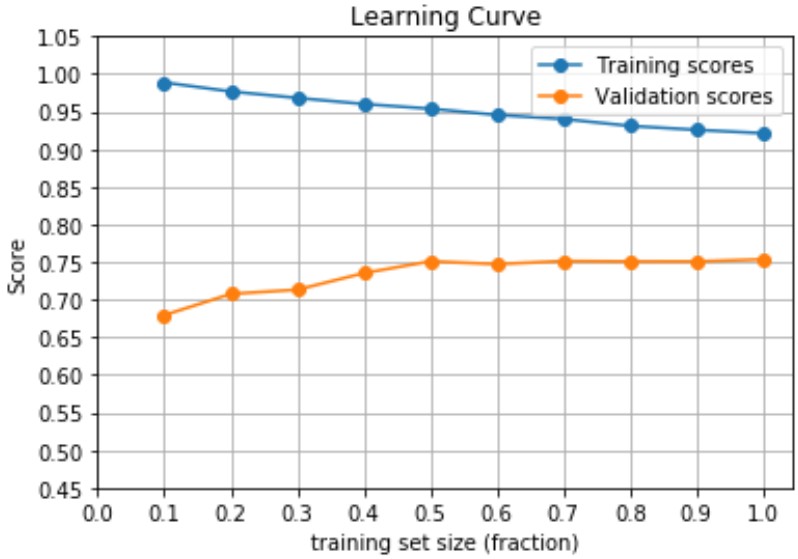

Figure 19: Gradient Boosting Classifier Training Curve

To optimize model performance, grid-search was used. Top 5 validation scores and corresponding hyperparameter were shown in Table 6.

Table 6. Top 5 Validation Scores for Grid Search

params	mean_test_score	rank_test_score
{'learning_rate': 0.01, 'max_depth': 3, 'min_samples_split': 4, 'n_estimators': 100}	0.762529	1
{'learning_rate': 0.01, 'max_depth': 3, 'min_samples_split': 4, 'n_estimators': 200}	0.762509	2
{'learning_rate': 0.01, 'max_depth': 5, 'min_samples_split': 4, 'n_estimators': 100}	0.762297	3
{'learning_rate': 0.05, 'max_depth': 5, 'min_samples_split': 6, 'n_estimators': 100}	0.762050	4
{'learning_rate': 0.01, 'max_depth': 5, 'min_samples_split': 2, 'n_estimators': 200}	0.761917	5

In addition, after parameter optimization, the entire dataset was used as well.

Table 7. Model summary and Comparison

Model	**Parameters**	**Data**	**Kaggle Public Score**	**Validation Score**
Logistic	solver:‘ liblinear’, penalty: ‘l2’	10：1	0.73018	0.71257
GradientBoostingClassifier	learning_rate: 0.01, n_estimators: 100, max_depth: 3, min_samples_split: 4	10：1	0.79129	0.76253
GradientBoostingClassifier	learning_rate: 0.01, n_estimators: 100, max_depth: 3, min_samples_split: 4	Orig	0.80153	0.76536

5. Kaggle Competition

Now, several models have been trained and cross-validated. Finally, these models were applied to the test data, kaggle submission files were generated with these models and submitted through kaggle. The best kaggle public score is 0.80153, which is at 31% currently.

Conclusion

1. Reflection

This project provides a great opportunity to apply data processing and modeling techniques to the real-world dataset. It took great effect to digest all the materials which have been taught in this Nanodegree program.

2. Improvements

   Given more time, there are several areas which can be improved.

   i. Better understand the meanings for each feature and use this information to better select features and do data processing.
   ii. Apply cluster analysis results to supervised learning
   iii. Build data processing pipelines to standardize data processing steps.

Reference

1. https://scikit-learn.org/stable/modules/clustering.html#k-means (k-means guidance)
2. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (understanding confusion matrix)
3. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (understanding auc-roc curve)