Project Definition

Project Overview

The aim of the project is to analyze demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population.

The data for this project includes general population dataset, customer segment data set, dataset of mailout campaign with response and test dataset that needs to make predictions. The data is not publically available. It was provided by Udacity partners at Bertelsmann Arvato Analytics only to those participating in the Udacity nano degree program.

The solution approach includes the following 3 phases:

- Data pre-processing: clean and re-encode data.
- Segmentation: use unsupervised learning techniques to create clusterings of customer and general population, and then - identify the difference.
- Prediction: use the demographic features to predict whether or not a person became a customer after a mailout campaign.

At the end a web app was developed to enable realtime prediction of possible customers based on the model generated.

Problem Statement

The problem statement for this project is to analyze demographics data for customers of a mail-order sales company in Germany, comparing it against demographics information for the general population.

The idea is to 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, applying what we have 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.

Metrics

Area under the receiver operating characteristic curve (ROC_AUC) from predicted probabilities will be used to evaluate performances of the models. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. 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 target person more highly than a random non-target person. Thus, ROC analysis provides tools to select possibly optimal models for customer prediction task.

AUC is desirable for the following two reasons:

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

We applied this metric to the 3 classifiers. The Adaptive Boosting Classifier (with 76% score) gave better predictions than the other 2 classifiers (Gradient Boost - 71% and XGBoost - 50%).

Analysis

- ### Data Exploration & Visualization

While preparing the data, the first steps was to collect information about missing data in the dataset as this is one of the key elements that affects Machine Learning models significantly. Identifying and understanding missing data information at an early stage will help us deploy a robust and efficient data processing routine.

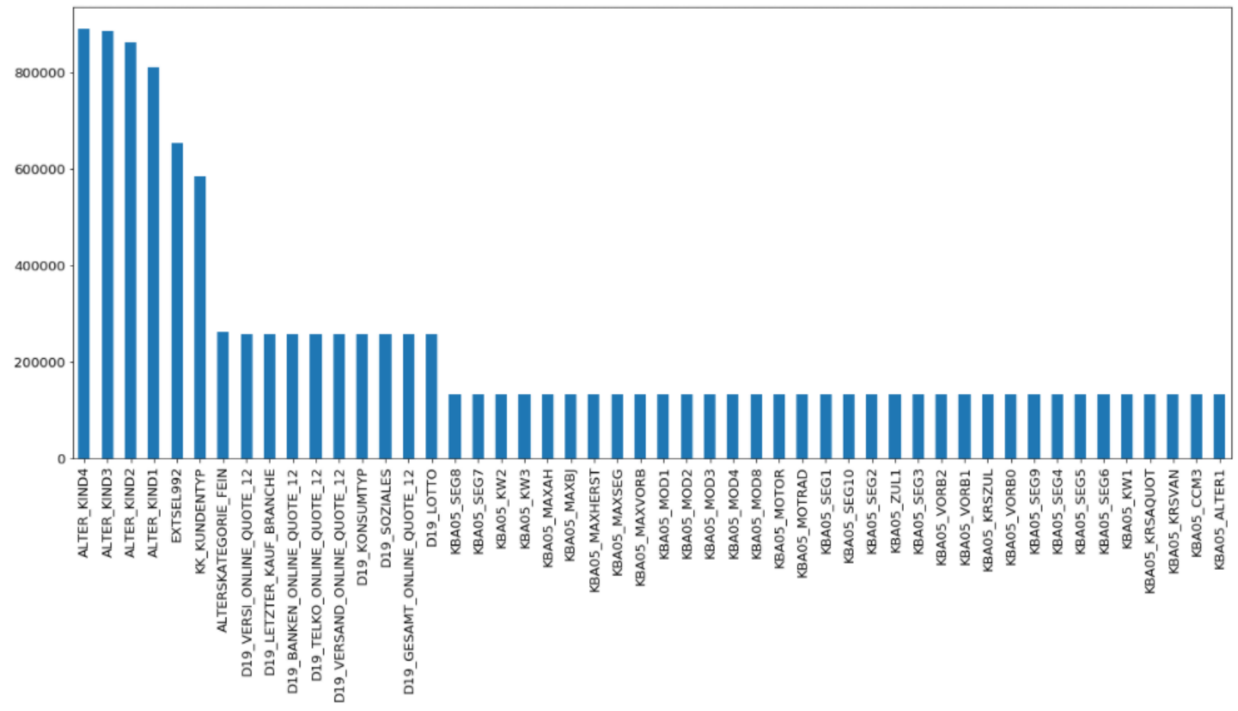


Figure 1 Top 50 missing data

The analysis demonstrates that most of the columns have less than 30% of missing data while there are 41 attributes with significant/high (more than 30%) missing data.

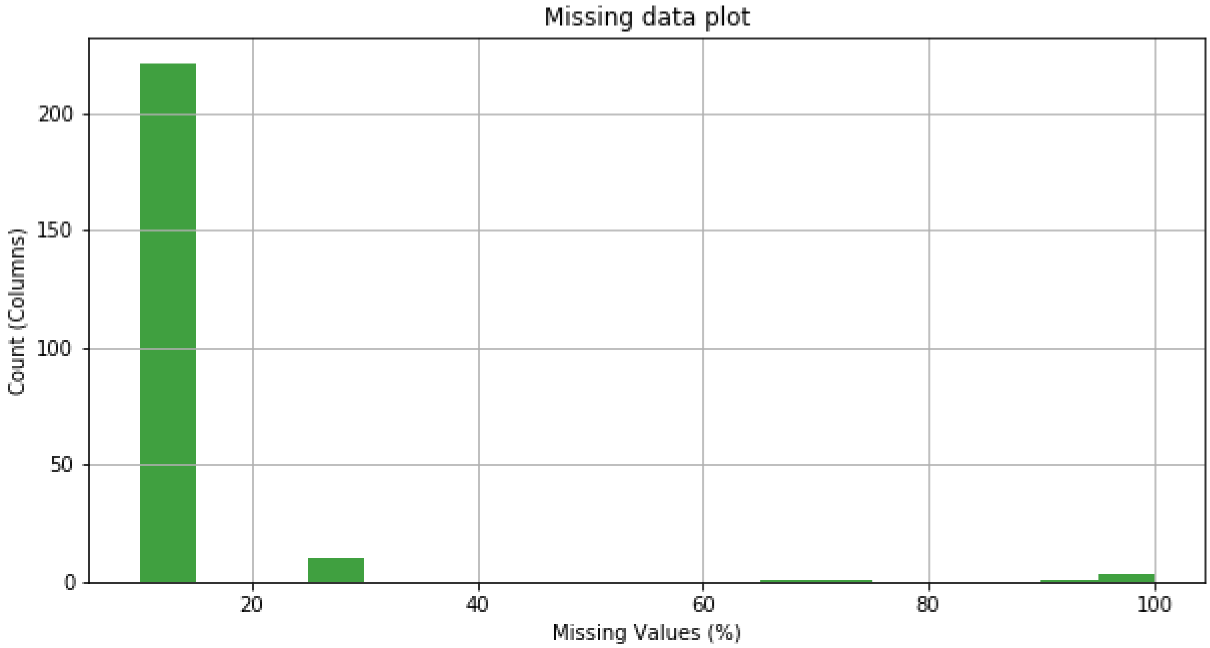


Figure 2 Proportion of missing values

The analysis of missing data by rows also revealed that the maximum number of missing data in each row is 233 attributes out of 303 attributes left after dropping columns.

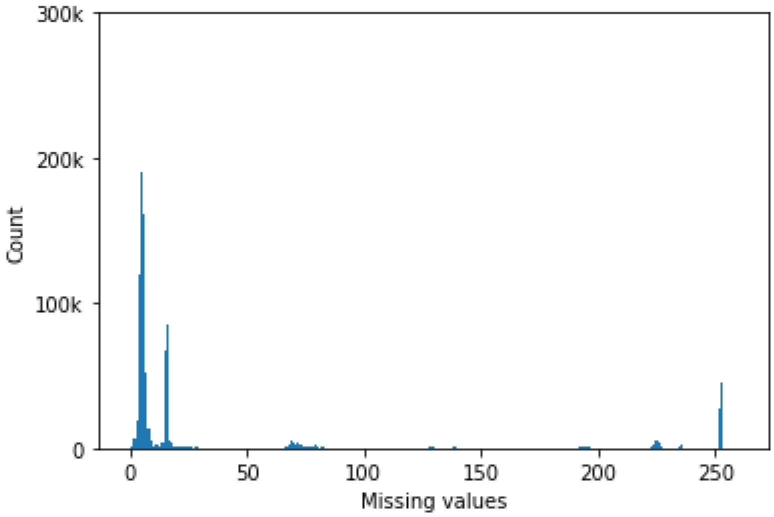


Figure 3 Missing Data in Rows

Next, we analyzed the extent of outlier in the data. Rows with outlier data were removed.
Finally, we dealt with categorical data by applying one hot encoder. There are four types of data in the dataset: ordinal, numeric, categorical and mixed. For ordinal variables, although the values might not be linearly related, we assumed the ordinal variables are interval variables here. The remaining categorical and mixed variables were re-encoded and selected.

This culminated in the creation of a data cleaning function which was used through out the project to clean and prepare the data used.

Methodology

- **Data Preprocessing**

Post data-preprocessing we discovered that the general population data (azdias) now has 100341 rows and 303 columns. This was after we had discarded less important features and outlier data. Since the data dimension is still high, we decided to use Principal Component Analysis (PCA) to reduce dimension to use Unsupervised Learning efficiently in the subsequent steps.

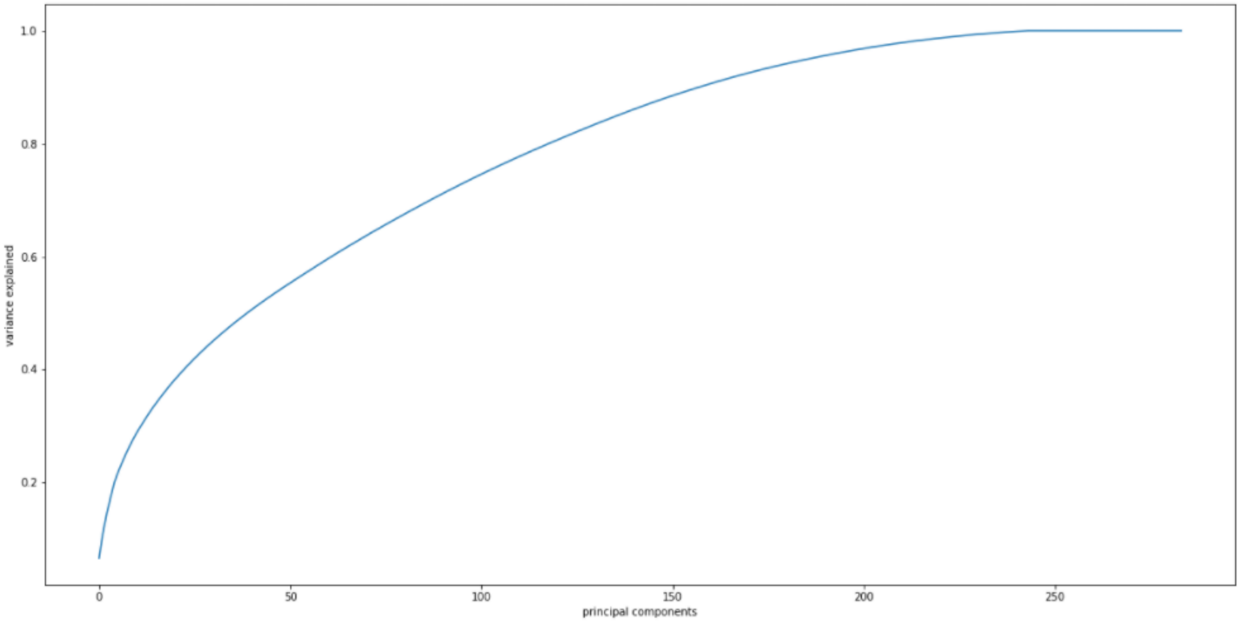


Figure 4 Principal Components Variance

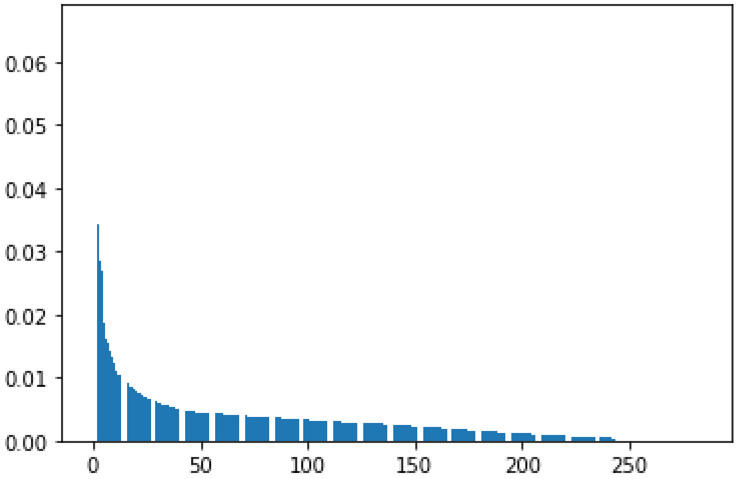


Figure 5 Variance investigation on each principal componen

- **Implementation**

With dimension now reduced, we proceeded to unsupervised learning (clustering). For this we are using the KMeans cluster algorithm. We used the “elbow” graph to determine optimal number of clusters (k) for our data set.

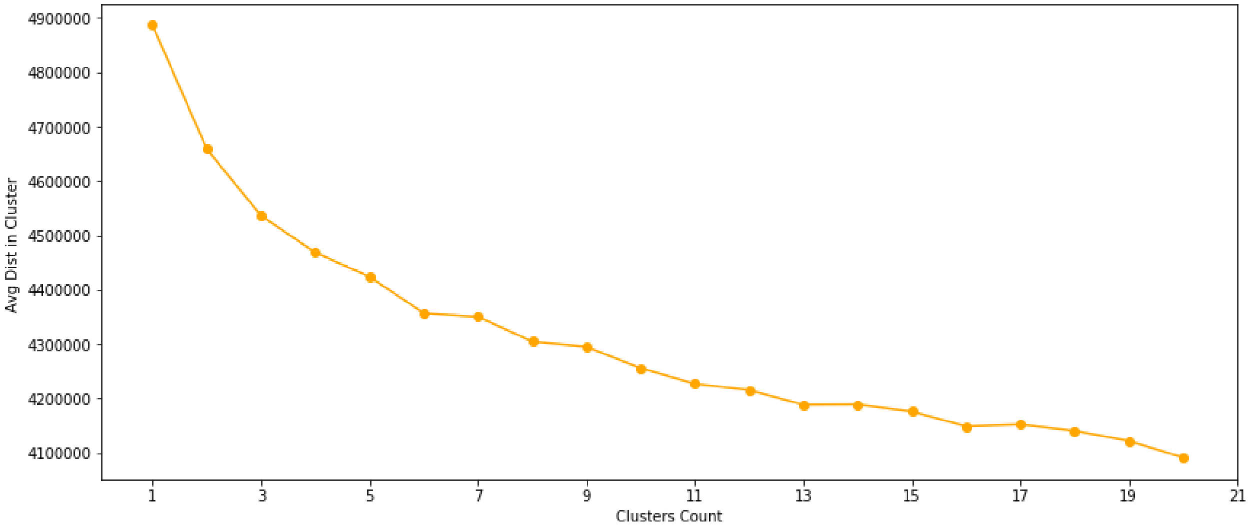


Figure 6 Elbow graph

Using the chart above we can see that the average distance within clusters starts getting flat as we approach 12 clusters. Informing the k value for our kmeans algorithm as 12

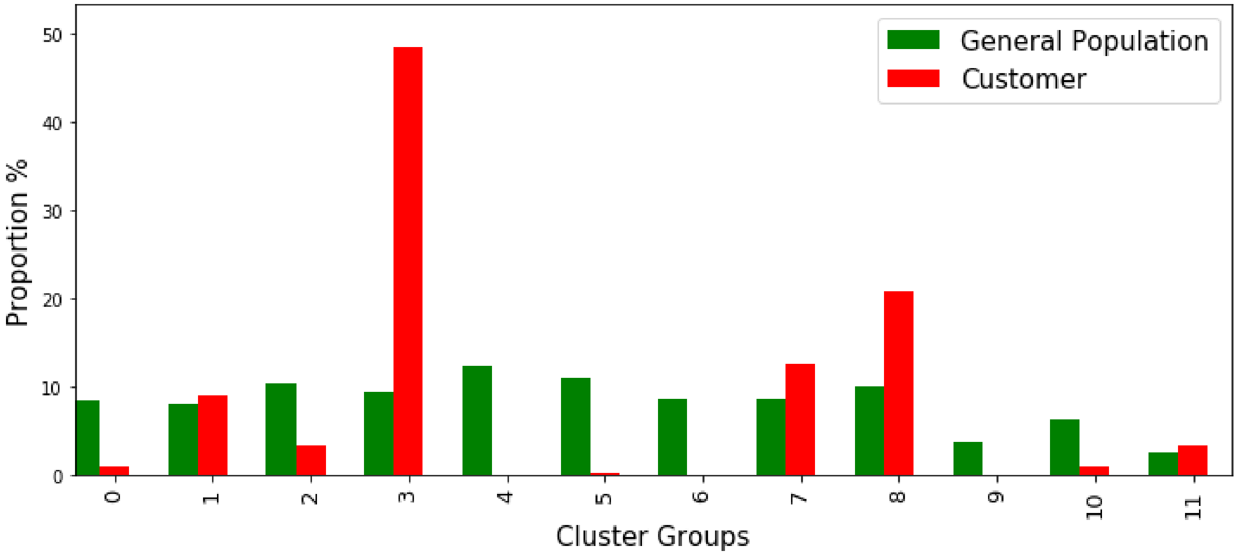


Figure 7 Proportion of customer segment data to general population.

Refinement

Supervised Learning Model

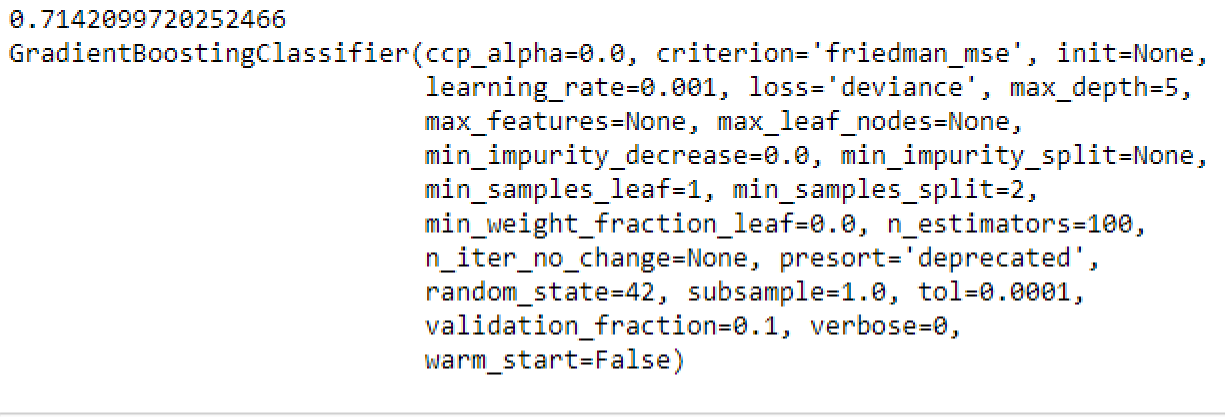
Having evaluated which parts of the population are more likely to be customers of the mail-order company, we proceeded to build a prediction model.

The dataset was cleaned using same preprocessing pipeline built earlier. The data was then split into training and validation datasets.

Results

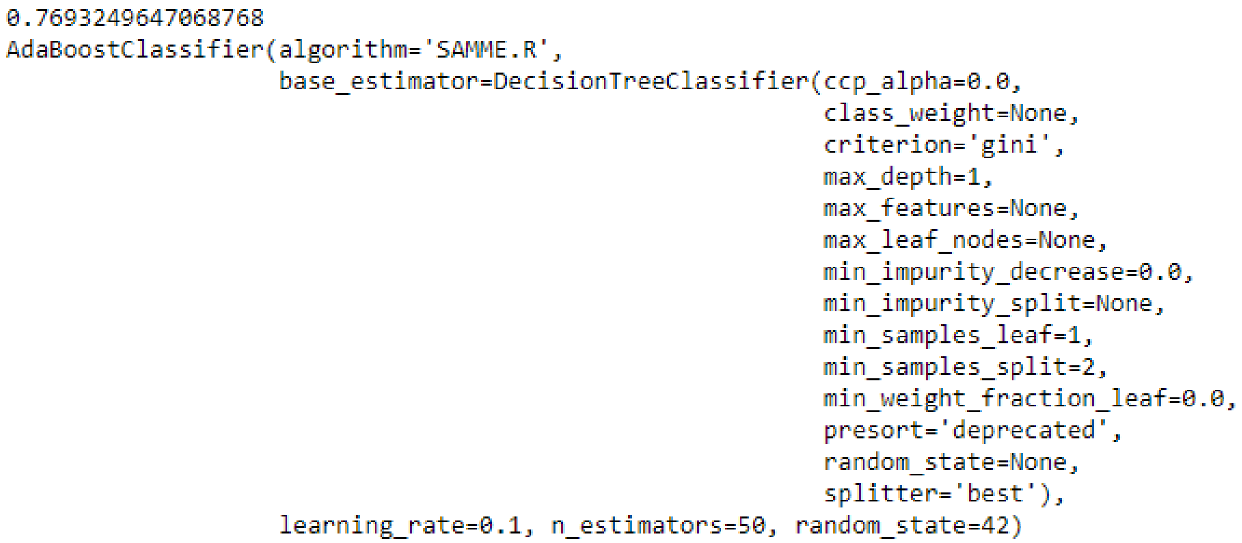
Model Evaluation and Validation

In order to evaluate and select best performing algorithm, we evaluated 3 algorithms (AdaBoostRegressor, GradientBoostingRegressor and XGBRegressor)and used ROC AUC evaluation metrics to select the best algorithm to use. Various hyper parameters were evaluated and the best performing parameters selected



```
0.7142099720252466
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.001, loss='deviance', max_depth=5,
                           max_features=None, max_leaf_nodes=None,
                           min_impurity_decrease=0.0, min_impurity_split=None,
                           min_samples_leaf=1, min_samples_split=2,
                           min_weight_fraction_leaf=0.0, n_estimators=100,
                           n_iter_no_change=None, presort='deprecated',
                           random_state=42, subsample=1.0, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)
```

Figure 8 Gradient Boosting Classifier



```
0.7693249647068768
AdaBoostClassifier(algorithm='SAMME.R',
                   base_estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                                         class_weight=None,
                                                         criterion='gini',
                                                         max_depth=1,
                                                         max_features=None,
                                                         max_leaf_nodes=None,
                                                         min_impurity_decrease=0.0,
                                                         min_impurity_split=None,
                                                         min_samples_leaf=1,
                                                         min_samples_split=2,
                                                         min_weight_fraction_leaf=0.0,
                                                         presort='deprecated',
                                                         random_state=None,
                                                         splitter='best'),
                   learning_rate=0.1, n_estimators=50, random_state=42)
```

Figure 9 Adaptive Boosting Classifier

Justification

We applied AUC_ROC score metric to the 3 classifiers. The Adaptive Boosting Classifier (with 76% score) gave better predictions than the other 2 classifiers (Gradient Boost - 71% and XGBoost - 50%).

See the Metrics section above for justification for this metric.

Conclusion

Reflection

This project has been quite enlightening and challenging. At the end we were able to deliver a model that can be used through a web app.

In the Customer Segmentation Report part, We caried out data pre-processing and used Principal component analysis (PCA) to compute the principal components and used them to perform a change of basis on the data, this led to the selection of a few principal components for our model.
We also used the K-means clustering to partition the data into 12 clusters in a way that data points in the same cluster are similar and data points in the different clusters are farther apart. This enabled us obtain the clustering of different population. By this we

were able to differentiate potential customer population from non-potential customer population. With this understanding, a company can better tell their target audience, and increase conversion rate or reduce their marketing/promotion cost.
The final part focused on building a model for predicting individuals that are more likely to respond and potentially become the company customers using the mailout campaign. Gradient Boosting, Adaptive Boosting and XGBoost classifiers were evaluated and AdaBoost was selected because it gave better results.

- **Improvement**

Increase PCA components can help improve the results. Using only the PCA components as independent variables.
It will also be helpful to have a clearer understanding of the features in order to determine the features to drop or re-engineer since the data dictionary does not include all columns