Arvato-Financial-Services

Machine Learning Engineer Nanodegree

Capstone Project

Joe Abraham

September 20, 2020

Project Motivation :

This competition is connected to one of Udacity's capstone project options for the Data Science Nanodegree program, in connection with Arvato Financial Solutions, a Bertelsmann subsidiary.

In the project, a mail-order sales company in Germany is interested in identifying segments of the general population to target with their marketing in order to grow. Demographics information has been provided for both the general population at large as well as for prior customers of the mail-order company in order to build a model of the customer base of the company. The target dataset contains demographics information for targets of a mailout marketing campaign. The objective is to identify which individuals are most likely to respond to the campaign and become customers of the mail-order company.

As part of the project, half of the mailout data has been provided with an included response column. For the competition, the remaining half of the mailout data has had its response column withheld; the competition will be scored based on the predictions on that half of the data.

The whole project is divided into 4 parts:

- Part 0 : Know the data, where you try to understand the data
- Part 1 : Customer Segmentation (Unsupervised Learning)
- Part 2 : Supervised Learning
- Part 3 : Kaggle competition

The main prerequisite for applying some machine-learning algorithm is that you need a clean dataset. Understanding and cleaning the dataset is the most critical task that I did. While processing data we should make sure that we lose important information. So data preprocessing is the most important stage.

Data Pre-Processing

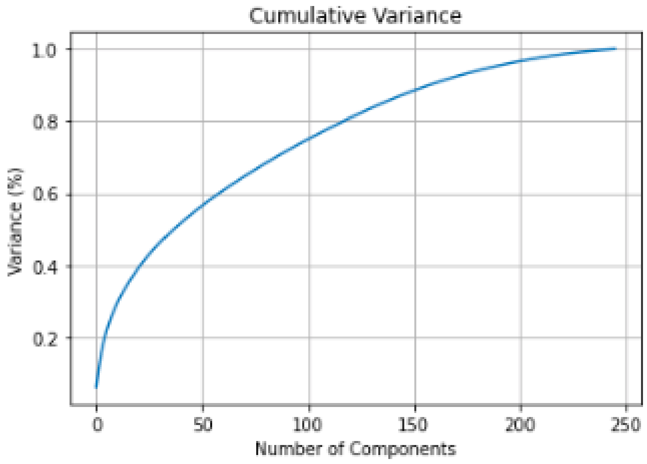
I have done the following steps in pre-processing

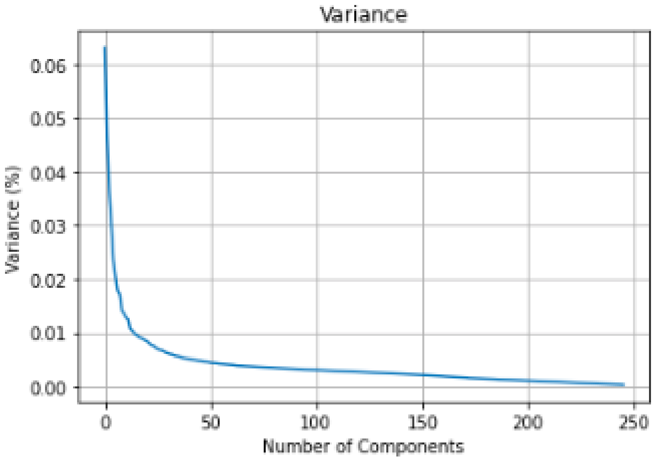
- Took the help of excel sheet to exactly know how unknown data is represented
- Filled the unknown data with NaNs
- Replace birth year with NaN, if it is zero.
- Dropped the columns, if most of the values are NaNs(only for unsupervised learning)
- Replaced NaN with most frequent value in that particular column
- Performed feature scaling

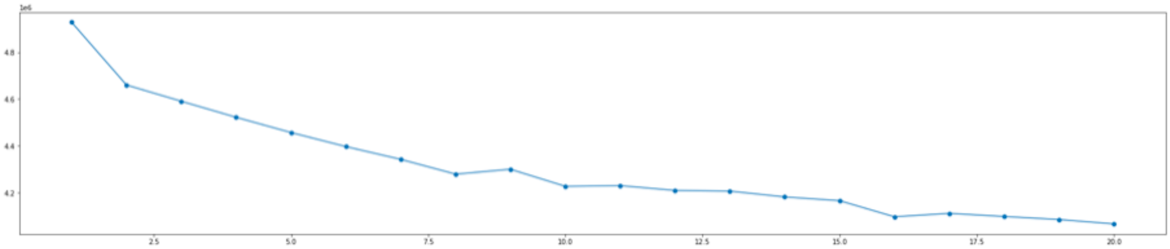
Un-supervised Learning

There was no constraints of elements in PCA and after doing PCA on the cleaned data, I found that variance almost become 0 after 200 components (graph available inside the notebook)

To find the right amount of cluster KMeans was applied in a loop from 1 to 19 and after that elbow method was used and 10 was my best choice to differentiate between customer and general public.

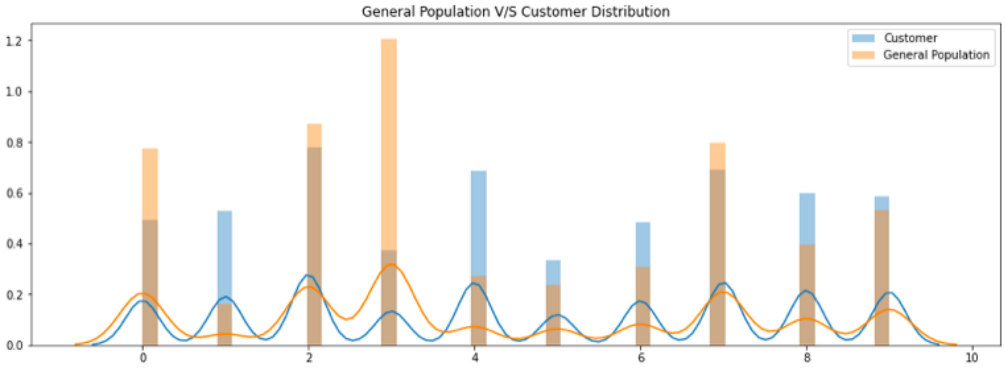






Graph with different centeroids for choosing best K-value in elbow method.

Final Conclusion in K-Means clustering

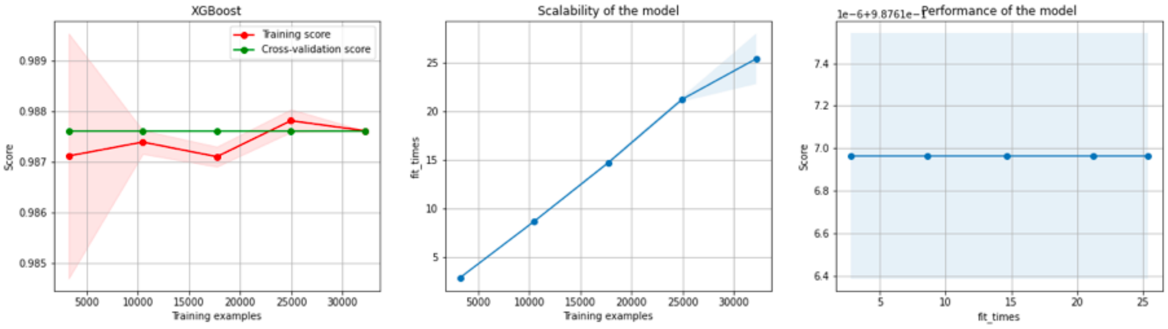


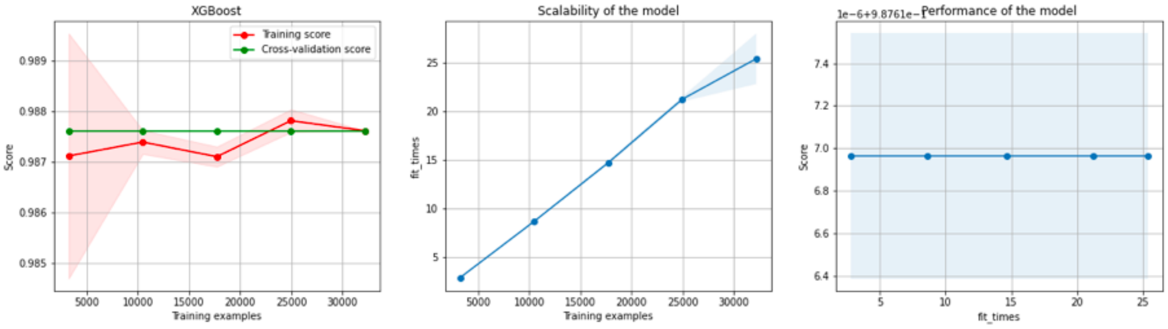
The above graphs shows that K-Means clustering with K = 10 helps to find customer from general public, it has very small difference, compared to others

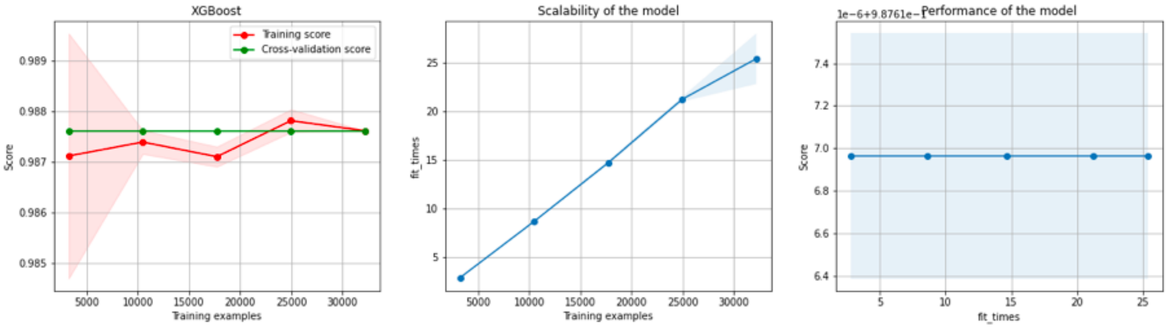
Supervised Learning

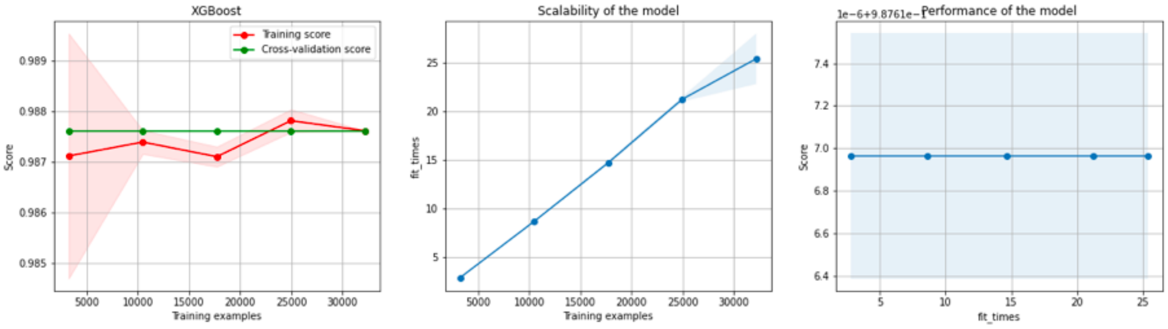
plot_learning_curve was taken from sklearn's official site. And plotters for XGBoost, Random Forest, AdaBoost and Decision Tree. And I found that XGBoost is performing well. Hyperparameter tuning is done to XGBoost and it was giving a cross validation score of 0.987

Learning curve





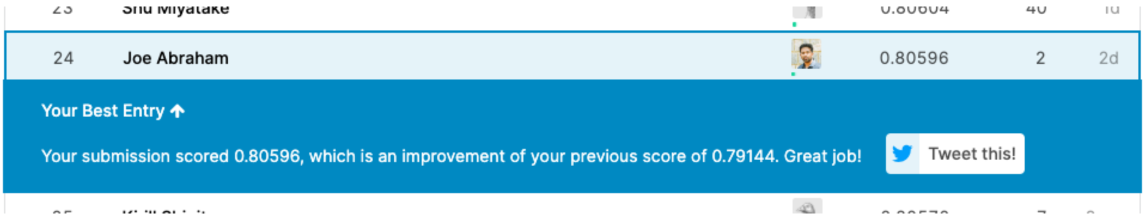




After hyperparameter tuning cross validation score came out to be 0.9876169638016635

Kaggle competition

The model was run on a provided dataset and then submitted on Kaggle and my model performed with a score 0.80596. And I grabbed 24th place in leadership board as on 20-09-2020



23	Shu Miyatake	0.80604	40	1d
24	Joe Abraham	0.80596	2	2d

Your Best Entry ↑

Your submission scored 0.80596, which is an improvement of your previous score of 0.79144. Great job!

Evaluation on Kaggle competition

The evaluation metric for this competition is AUC for the ROC curve, relative to the detection of customers from the mail campaign. A ROC, or receiver operating characteristic, is a graphic used to plot the true positive rate (TPR, proportion of actual customers that are labeled as so) against the false positive rate (FPR, proportion of non-customers labeled as customers).

The line plotted on these axes depicts the performance of an algorithm as we sweep across the entire output value range. We start by accepting no individuals as customers (thus giving a 0.0 TPR and FPR) then gradually increase the threshold for accepting customers until all individuals are accepted (thus giving a 1.0 TPR and FPR). The AUC, or area under the curve, summarizes the performance of the model. If a model does not discriminate between classes at all, its curve should be approximately a diagonal line from (0, 0) to (1, 1), earning a score of 0.5. A model that identifies most of the customers first, before starting to make errors, will see its curve start with a steep upward slope towards the upper-left corner before making a shallow slope towards the upper-right. The maximum score possible is 1.0, if all customers are perfectly captured by the model first. (It should be noted that this particular task is very difficult with a lot of noise, and so you should not expect extremely high scores!)