Domain background:
A client of Arvato-Bertelsmann, a mail-order sales company, is interested in identifying segments of the population to target with their marketing. The objective is to identify the persons that are most likely to respond to the campaign and become customers.

Problem statement:
The problem statement is how the mail order company can acquire clients more efficiently.

Datasets and inputs:
It is provided demographics information for both general population and former customers of the company.

Solution statement
My approach for the problem will be:

- Perform data visualization to look for relevant feature information, possible number of groups, understand data and domain to engineer new features if possible.
- Look at the feature importance, firstly univariant, correlation between features and labels, and also multivariant using a simple model or a more complex one, depending on the computation cost.
- Perform Kmeans to segment the data into range(N:M) groups
- Perform XGBoost, Sklearn Random Forest and Pytorch
- Evaluate the results in a Matrix with AUC as the metric
- Submit the best ones to Kaggle and observe if the performance order is maintained or it is not (possible difference of variance between train and test data)

Benchmark model
To benchmark the model my idea is to create a matrix with unsupervised methods on x-axis and supervised on y-axis in order to have some insights about which are the more successful combinations and try from this point with different hyperparameters or models to evaluate if the results are improving.

Evaluation metrics
The model will aim for the highest possible AUC.

AUC is defined as the area under the ROC curve.

ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The graphic of the curve is done by plotting the true positive rate agains the false positive rate at various threshold settings.

Project design
As we have developed the Nanodegree with AWS Sagemaker I find interesting to continue improving my skills with this ecosystem.

- Create a new notebook instance
- Include the initial notebook and the data for training and test
- Add training and test scripts for the models that require them

- Sections of the notebook
  - Data load
  - Data visualization
  - Feature engineering
  - Feature importance
  - Model training
  - Model testing
  - Results saving