Udacity Machine Learning Nanodegree 2020

Capstone Proposal

Customer Segmentation Report for Arvato Financial Services

Mark Anthony B. Dungo

February 2020

1 Domain Background

Knowing your customer characteristics, so that you can choose which among the people have a high probability to be our customers is a good market strategy, dividing customers and the general population into groups based on their common characteristics can help us analyze which on the population might have interest on our services to market to each group effectively and appropriately.

The goal of this capstone project, is to perform k-means clustering and choose the best classifier for Customer Segmentation Report, Supervised Learning Model and Kaggle Competition. Having a good supervised model will help us find a list of people that might have interest in our services if we have the database about the information of people.

2 Problem Statement

The main objective of this project will be to use unsupervised and supervised techniques.

I. Learn how analyzing and visualization of the groups of our customers and the population based on their common characteristics can help us market our business.
II. We want to compute and optimize the best relevant features to fit into our classifier to build a good supervised model for features and class labels that contain a lot of unstable NaN and zeros.

3 Datasets and Inputs

For this project we will use the datasets which are stored from the following files:

- 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).

For more information about the columns depicted in the files, you can refer to two Excel spreadsheets provided in the workspace. One of them is *"./DIAS Information Levels - Attributes 2017.xlsx"* a top-level list of attributes and descriptions, organized by informational category. The other is *"./DIAS Attributes - Values 2017.xlsx"* a detailed mapping of data values for each feature in alphabetical order.

The "CUSTOMERS" file contains three extra columns ('CUSTOMER_GROUP', 'ONLINE_PURCHASE', and 'PRODUCT_GROUP'), which provide broad information about

the customers depicted in the file. The original "MAILOUT" file included one additional column, "RESPONSE", which indicated whether or not each recipient became a customer of the company. For the "TRAIN" subset, this column has been retained, but in the "TEST" subset it has been removed; it is against that withheld column that the final predictions will be assessed in the Kaggle competition.

4 Solution Statement

The main proposed solution to this problem is to apply Machine Learning techniques that have proved to be reliable in creating a model for complicated training data features.

I. First I did a little Customer Segmentation Report's visualization.
    A. Get to know the data and check XML spreadsheets.
    B. Analysing the descriptions of the demographics datas.
    C. K-means clustering and visualizations. [1]
II. In the next part, I will train a supervised learning model.
    A. Familiarizing the demographics datas.
    B. Encode some incompatible data feature values.
    C. Select good relevant features using Pearson correlation. [2]
    D. Try to use the Nystroem method to speed up the process.
    E. Choose what classifiers perform the most to improve. [3]

I will use the evaluation metrics described in later sections to compare the performance of the II solution against the benchmark models in the next section.

5 Benchmark Model

For the benchmark model, I use different classifiers and minimum correlation targets for getting the relevant features to include on our training data features.

Tested Classifiers including Parameters

- sklearn.svm.SVC(gamma='scale', probability=True)
- sklearn.svm.SVC(kernel='poly', gamma='scale', probability=True)
- CalibratedClassifierCV(sklearn.svm.LinearSVC(dual=False), cv=5)
- sklearn.linear_model.SGDClassifier(loss='log')
- sklearn.linear_model.SGDClassifier(loss='modified_huber')
- ensemble.AdaBoostClassifier()

- ensemble.BaggingClassifier()
- ensemble.ExtraTreesClassifier(n_estimators=100)
- ensemble.GradientBoostingClassifier()
- ensemble.RandomForestClassifier(n_estimators=100)
- sklearn.linear_model.LogisticRegression(solver='lbfgs', max_iter=7600)
- sklearn.neighbors.KNeighborsClassifier()

There are only two classifiers that perform well. They are AdaBoostClassifier and GradientBoostingClassifier all with default parameters.

These are the top 3 on the result of my classifiers and minimum correlation targets' test.

Classifier: AdaBoostClassifier
Minimum correlation target: 0.0155000000
Accuracy: 98.74316303968347% | ROC AUC: 0.7961298806172112

Classifier: AdaBoostClassifier
Minimum correlation target: 0.0200000000
Accuracy: 98.59187710927499% | ROC AUC: 0.7888601440623073

Classifier: GradientBoostingClassifier
Minimum correlation target: 0.0155000000
Accuracy: 98.74316303968347% | ROC AUC: 0.7836427028088784

6 Evaluation Metrics

The evaluation metric for my classifiers and minimum correlation targets' test are the scores of their Area Under the Receiver Operating Characteristic Curves (ROC AUC) from their probability prediction scores.

7 Project Design

Data Preprocessing

First analyze the columns and the feature values in our datasets and what processing needs to be done to make them valid.

- Split the dataset to have our training and test features and class labels.

- Make sure there's no invalid values on our features and class labels.
- Check to use the Nystroem method to speed up the process. I later figured out that it's producing inconsistent output values and removed them from my code.
- Consider optimizing my codes, such as an instant multiple classifiers and minimum correlation targets' testing per single code run.
- Train repeatedly and only save if the result scores are good.

Data Splitting

Split the data into a training set and test set with an 80%-20% to 70%-30% split.

Model training and evaluation

I will create a script to train and evaluate multiple model architectures with different hyper-parameters. The script iterates this process with different minimum correlation targets. Then I store the best performing model and train it very well.

8 References

[1] Chinna Naidu, "Customer Segmentation using SVM | Kaggle"
https://www.kaggle.com/chinnanaidu/customer-segmentation-using-svm

[2] Abhini Shetye, "Feature Selection with sklearn and Pandas - Towards Data Science"
https://towardsdatascience.com/feature-selection-with-pandas-e3690ad8504b

[3] Jason Brownlee, "How to Use ROC Curves and Precision-Recall Curves for Classification in Python"
https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/