95-851 Fall 2020 - Making Products Count: Data Science for Product Management

Homework 4: Market Segmentation using *k*-means clustering (**due October 14, 2020**)

Introduction: *k*-means Clustering

k-means clustering is an algorithm commonly used for market segmentation analysis. *k*-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. *k*-means clustering is an unsupervised learning technique, which means the market segmentation problem, we can put customers that exhibit similar behaviors into the same cluster for further analysis. In this assignment, you will apply the *k*-means algorithm to the Cannondale bikes data set to perform some basic segmentation analysis. The example is taken from http://www.business-science.io/business/2016/08/07/CustomerSegmentationPt1.html which walks through the solution in R. For this assignment ,you should code the solution in Python.

Dataset

The dataset contains bicycle orders. The first data table `bikes.xlsx` contains information about various bike models. The second data table `bikeshops.xlsx` contains information about bike shops. The third data table `orders.xlsx` contains the actual order histories with the manufacturer Cannondale. Note, in `orders.xlsx`, customer.id refers to the IDs of bikeshops who ordered the bike and the product.id refers to the IDs of specific bike models.

Task

In this assignment, you will first go through some data preprocessing steps and then perform a k-means clustering analysis to analyze segmentation of bike purchasing behavior. Keep in mind that here the objects being segmented by purchasing behavior are bikeshops, not the people who are the actual end customers for the bikes.

Before delving into the analysis, we need to develop a hypothesis for purchasing trends. Developing a hypothesis is necessary, as the hypothesis will guide our decisions on how to formulate the data in such a way to cluster bikeshops. For the Cannondale orders, our hypothesis is that bike shops purchase Cannondale bike models based on features such as Mountain or Road Bikes and price tier (high/premium or low/affordable). The bike model features (e.g. price, category, etc.) will be used for assessing the preferences of the shop clusters (more on this later).

To start, we′ll need a unit of measure to cluster on. We can select quantity purchased or total value of purchases. We′ll select quantity purchased because total value can be skewed by the bike unit price. For example, a premium bike can be sold for 10X more than an affordable bike, which can mask the true buying habits of interest.

Step 1: Data preprocessing

- First, you will load all three datasets (bikes, bikeshops, and orders) and merge them together into one single data frame for further analysis. Remember, you can link the three datasets together by using the product.id and customer.id in the orders table.
- Next, convert the unit price to categorical high/low variables. To do this, you can divide the unit prices into two halves using the median as the split point.
- Spread the bikeshops by quantity of bike models purchased. To do this, you can group the data by model & model features (‘bikeshop.name', 'model', 'category1', 'category2', 'frame', 'price') and summarize by quantity purchased. Then, you can use a pivot table to allocate the aggregated quantities across different bikeshops. This transforms the data frame from one in which each row is an order from a bikeshop for a model, to one in which each row is a model, and one column for each of the model′s attributes and one column for each bikeshop, indicating what fraction of the shop′s orders are for that bike.
- Last, we need to scale (normalize) the quantity data. Unadjusted quantities presents a problem to the k-means algorithm. Some bikeshops are larger than others meaning they purchase higher volumes. Fortunately, we can resolve this issue by converting the order quantities to proportion of the total bikes purchased by a bikeshop.

Step 2: Perform *k*-Means Clustering

Now you are ready to perform *k*-means clustering to segment our bikeshops. Think of clusters as groups of bike shops that behave similarly. Prior to starting we will need to choose the number of groups, *k*, that are to be detected. The best way to do this is to think about the shops and our hypothesis. We believe that there are most likely to be at least four groups of shops because of mountain bike vs. road bike and premium vs affordable preferences. We also believe there could be more as some customers of some shops may not care about price but may still prefer a specific bike category. However, we′ll limit the clusters to eight as more is likely to overfit the segments.
- Fit a *k*-means Clustering model on preprocessed data. You can use the *k*-means package from the scikit-learn library in python. Refer to external resources if you need more information on the scikit-learn library.
- Clearly state your choice of the parameter *k* and your assumption behind choosing a specific *k* parameter. How many clusters do you think there should be and why?

Step 3: Analyze the result

Now that you have clustered the data, you can inspect the groups find out which shops are grouped together.
- Print out shop names that are in each segment
- Determine the preferences of the shop segments by inspecting factors related to the model (e.g. price point, category of bike, etc.). To do this, you need to combine cluster centroids with bike models for feature inspection. Then for each cluster, arrange top 10 bike models by cluster in descending order and print out this information.

What to submit

Save your .ipynb as an .html file and submit it to Canvas. Remember to run all your cells and display the output in readable format before submitting your code!

External Resources

http://www.business-science.io/business/2016/08/07/CustomerSegmentationPt1.html
http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html