Customer Segmentation

Mingrui Zhang

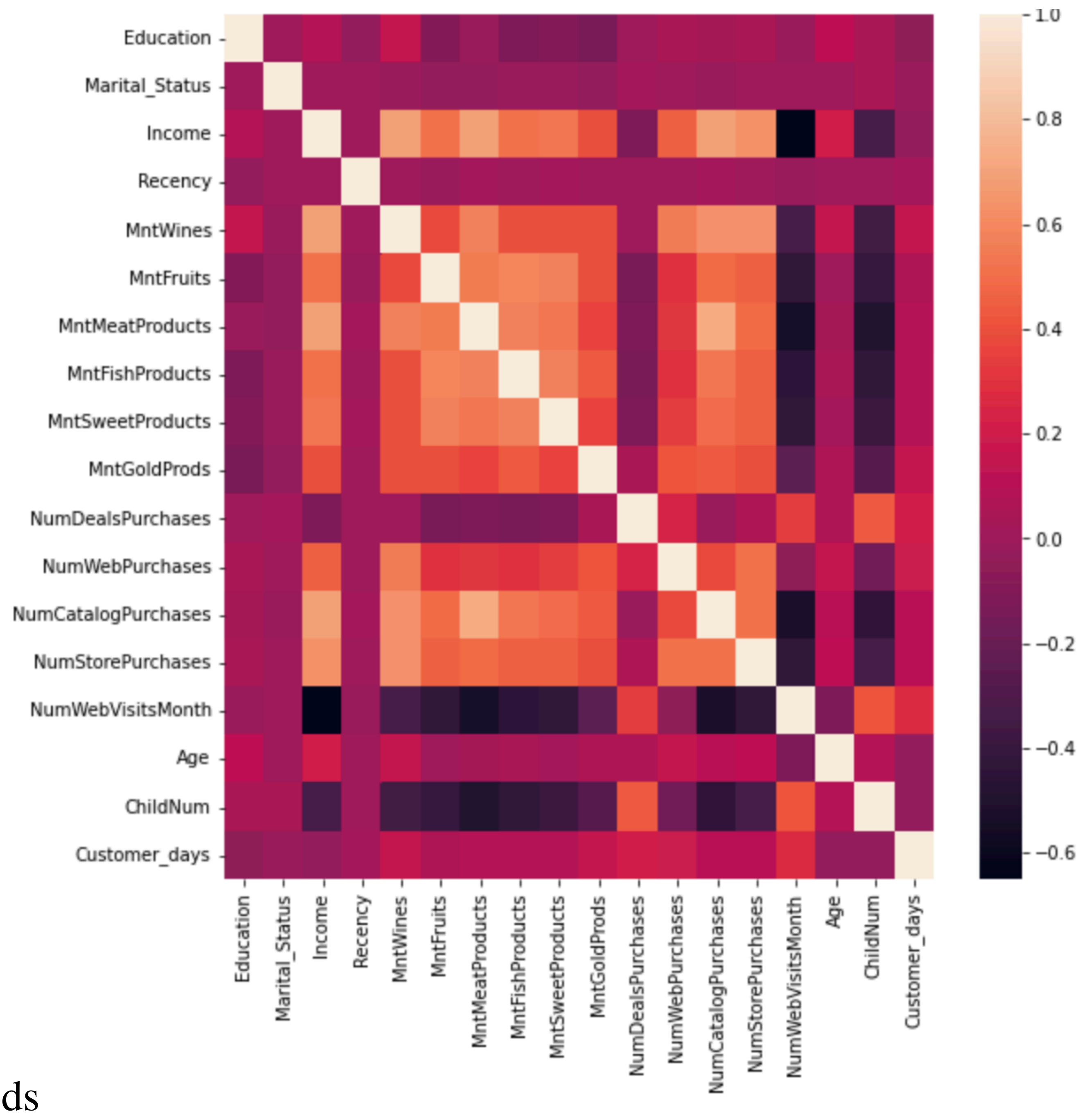
zhang.mingr@northeastern.edu

Problem Definition

Customer segmentation is a way to divide customers into groups based on common characteristics so that companies can make market decisions and develop specialized marketing strategies to different groups. Given a customer dataset, we need to use unsupervised machine learning methods to cluster customers into some number of groups and analyze characteristics of different groups. Meanwhile, the quality of clustering also need to be taken into consideration.

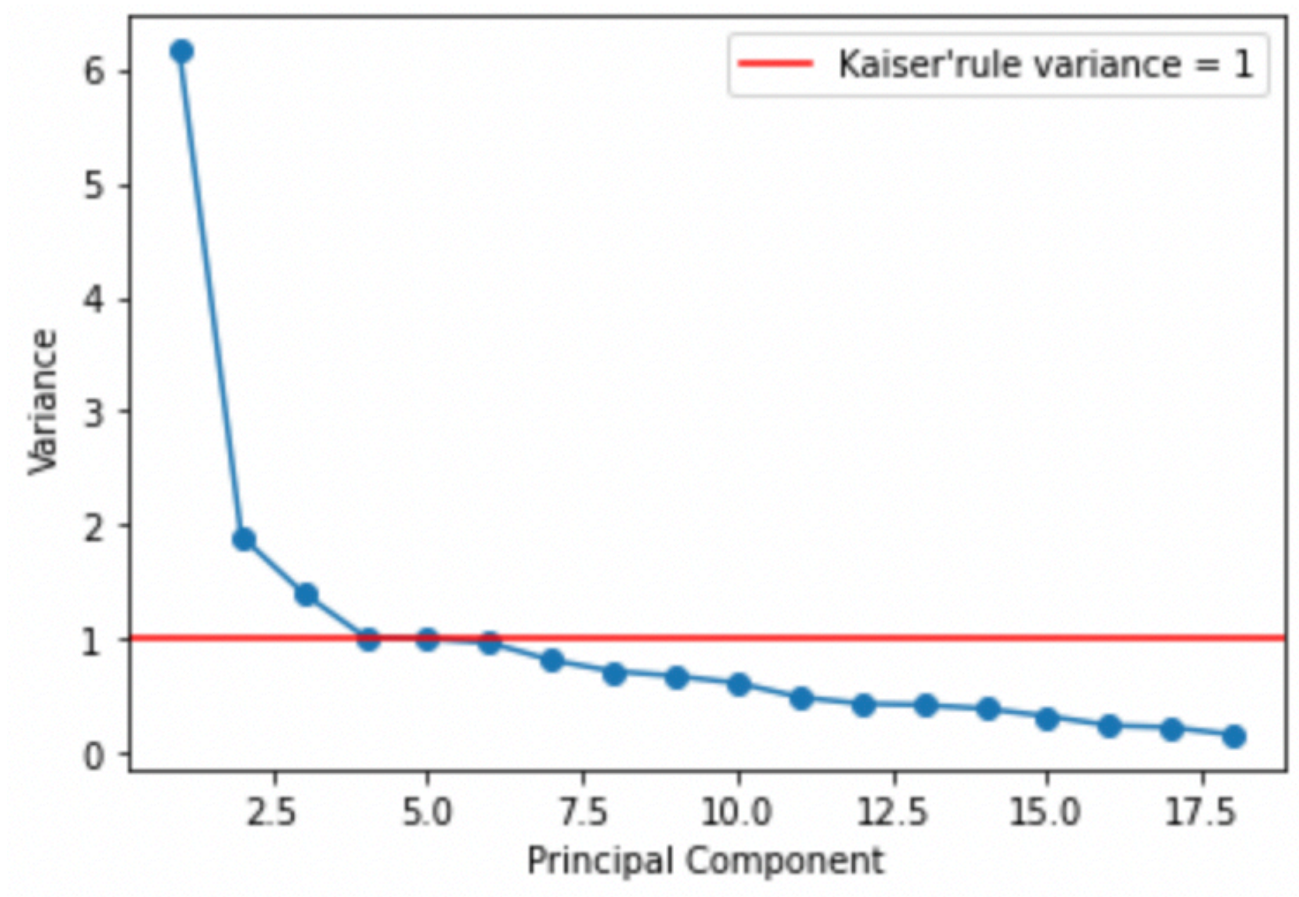
Existing Methods

Use simple PCA with fixed number of principal components to do dimensionality reduction method. Then implement Kmeans and the Elbow method to find optimal clustering.



Proposed Method

1. Use two dimensionality reduction methods to make a comparison: PCA and Autoencoder applied into a dataset containing customer information. Both methods we find its optimal dimensions to reduce.
2. Implement Kmeans to cluster.
3. Choose Silhouette score and DB index to evaluate the quality of clustering.
4. Plot multiple distributions between clusters and original features to achieve customer segmentation and then conclude some significant characteristics of each cluster.

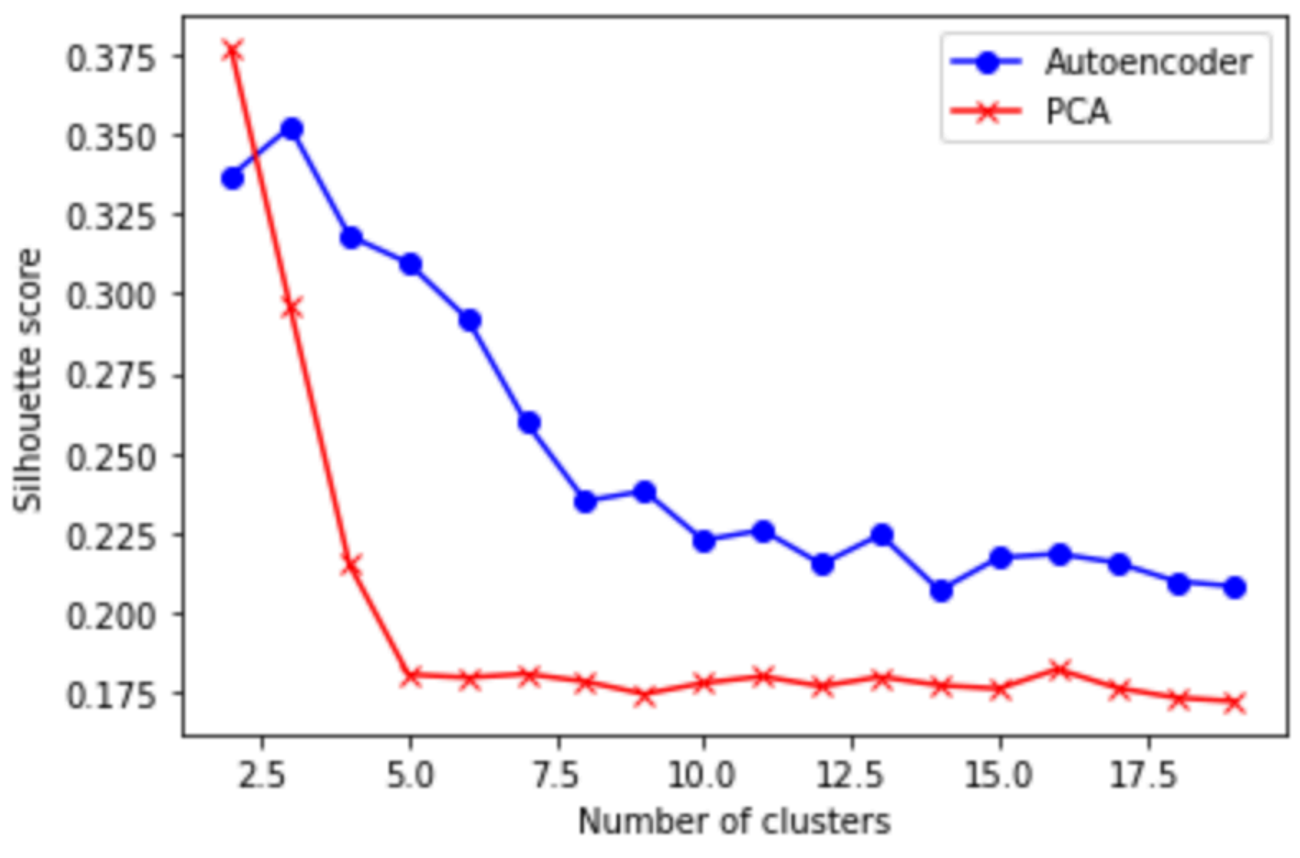


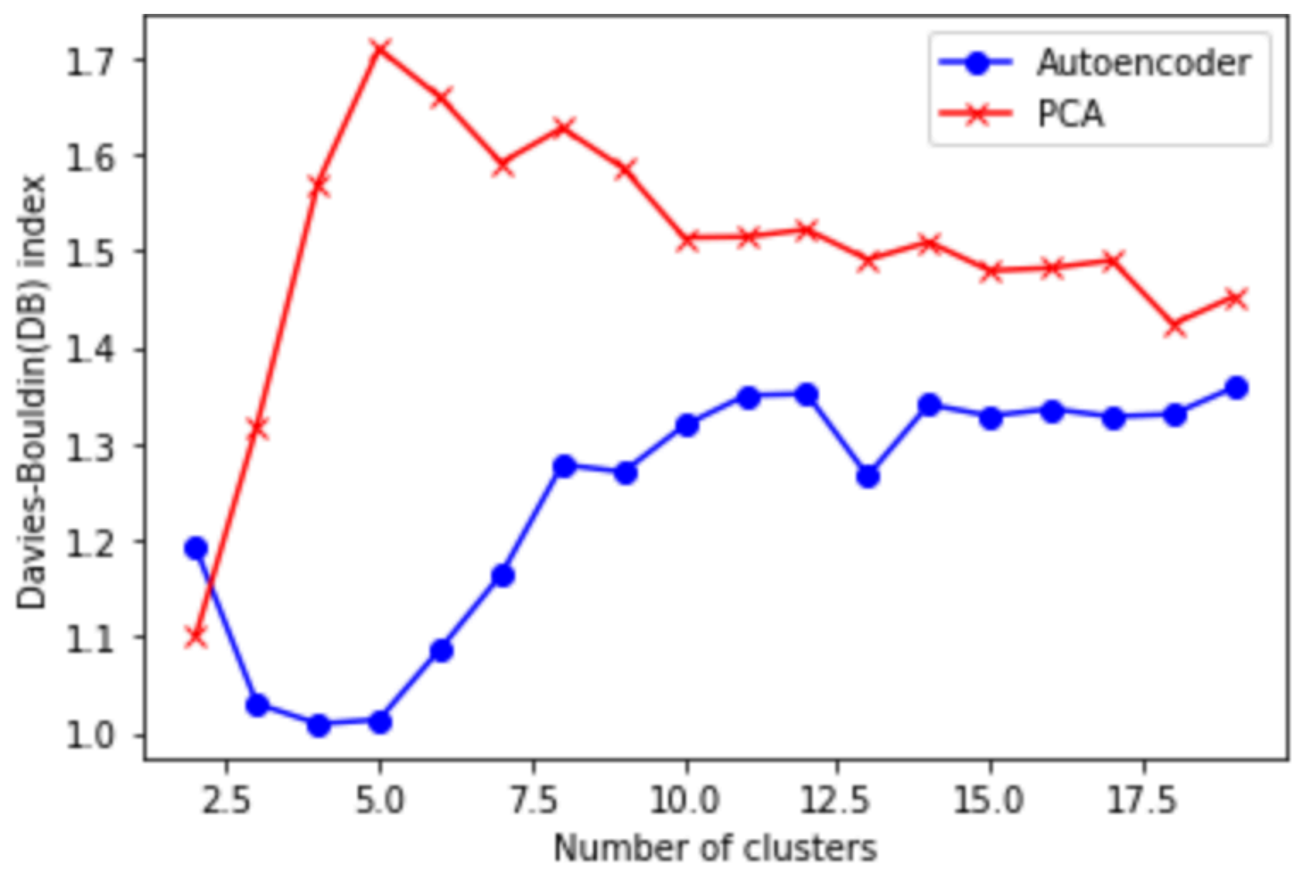
Data Description & Experimental Setup

The dataset is from Kaggle. The URL is https://www.kaggle.com/imakash3011/customer-personality-analysis. It has 2240 customer records for us to analyze and each customer has 29 features within four main parts: customer's basic information, the total quantity a customer spent in six kinds of products, promotion that customer ever used and the number of times customer bought products in four purchasing ways such as online, in store etc.

The experiments are run under google colab, a platform similar to jupyter notebook that anyone can write and execute arbitrary python code through the browser.

For training Autoencoder, we set loss=mean square error, optimizer=Adam, epochs=500, batch_size=256 with the other default Keras parameters.





Results & Discussion

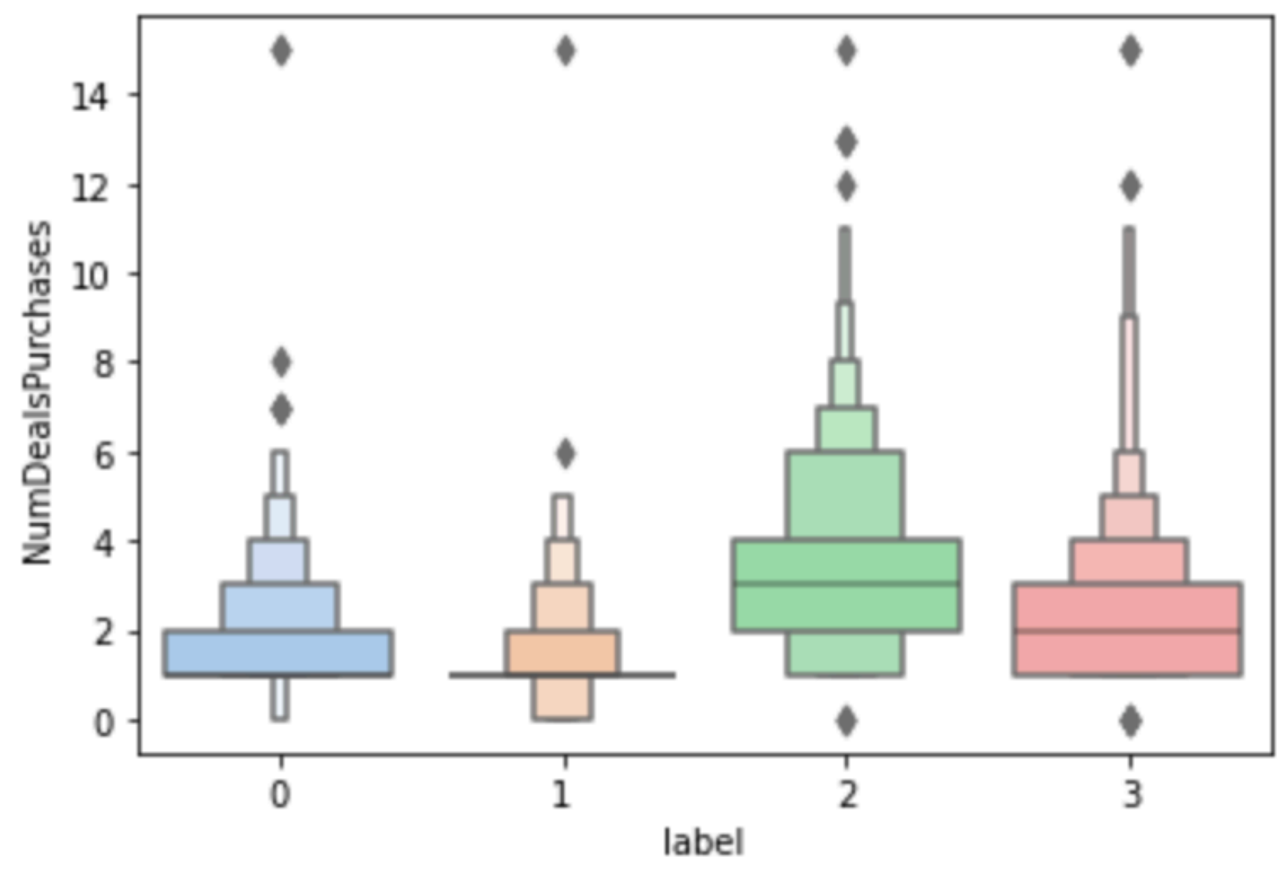
1. Feature correlation: Income has a negative correlation with NumWebVisitMonth and ChildNum and a positive correlation with all the features about the number of purchasing especially MntWines and MntMeatProducts. It means that people with higher income tend to have less children, buy more products like wines and meat and seldom go shopping online.
2. PCA: Variance of Top 5 principle components are above 1 and they take about 65% of overall variation. Feature Income contributes the most to Top 1 component.
3. Autoencoder: model with hidden layer of 5 nodes has relatively low MSE and better overall performance.
4. Kmeans: For both methods, the 'elbow value' is around 5. Distortion of PCA is about only 10% of autoencoder's. It shows that the samples reduced by autoencoder into 5-dimensions are more sparse and far away from each other.
5. Evaluation metrics: Compared with PCA, autoencoder performs better on both metrics which means dimensionality reduction of autoencoder is better on our dataset.
6. Customer segmentation:

Cluster 0: most only have one child, basic education degree, not usually use coupons, have middle-upper level income, love spending money in meat and gold products.

Cluster 1: most don't have child, have a high education degree, seldom use coupons, have a high level income, love spending money on meat but not gold products.

Cluster 2: have multiple children, basic education degree, usually use coupons, low level income, spend little money on meat but love buying gold products.

Cluster 3: have multiple children, middle education degree, often use coupons, very low level income, spend little money on meat and gold products.



Takeaway Points & Future Work

1. Make a baseline: The processed dataset has only 18 features which is not a large dimension. We can have a baseline of clustering result on the dataset without dimensionality reduction compared with reduced one to show if reducing dimensions is a good choice.
2. Build a more complicated Autoencoder: We can train Convolutional autoencoder to achieve a better performance.