Customer segmentation using clustering analysis on data collected from mobile shopping app

1. Definition

Project Overview

A key component to any strategic marketing, branding or business growth is market segmentation. The data set analyzed in this Capstone project is a user data collection from a mobile marketplace app for used goods. By segmenting the sellers into multiple groups, the company could provide a better customer support by training support staffs accordingly for each seller group when sellers reach out for assist throughout the process of selling their items.

Problem Statement

The goal of the project is to segment the sellers into number of groups and investigate the characteristics and uniqueness of each group. The preliminary assessment by the data provider suggests that there are four distinct seller groups: top sellers, business sellers, casual sellers and new sellers.
This capstone project will further investigate the data using various techniques of clustering analysis and will determine the number of unique seller groups based on the given features of the data set.

Metrics

Silhouette analysis will be used to test the number of clusters after KMeans clustering analysis was conducted on the dataset. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. (scikit learn webpage)

2. Analysis

Data Exploration

The dataset analyzed in this project contains sellers' activities which are shown as columns (features) in the form of csv file. The descriptions of features are shown in Table 1.

Table 1 Features of dataset

Column name	Description
id	user ID
install_date	the user install date
time_on_site	days since the user install date
positive_rating	number of positive ratings the user received as a seller
neutral_rating	number of neutral ratings the user received as a seller
negative_rating	number of negative ratings the user received as a seller
listing	number of items the user has listed for sale
listing_gmv	total dollar amount of the listed items from the user
sale	number of sales the user made
buyers	number of unique buyers of the user's items
gmv	total dollar amount of the user's sold item

Explain about data value of zero

Though the length of the dataset contains over 1.2 million historic seller records, majority of them implies new users who did not show any sale history using the app ('sale' column in the data set is zero and therefore other columns as well) at the time of this dataset. Only 8.07% of the dataset represents the sellers with at least one item listed for sale (97,423 out of total 1,207,774 sellers). These users are referred to as 'active users' in the following sections throughout the analysis.

Distribution of active sellers

Scatter plots and histograms are shown in Figure 1, which shows very wide ranges of the users across plotted features. The plots also represent the skewedness of data toward lower quantities for the features such as 'listing,' 'sale' and 'gmv' (total revenue). We can deduce based on these plots that majority of the sellers in the dataset represents new or relatively casual sellers. Throughout this project, the focus will be given to identify sellers beyond the new and casual levels, i.e. professional sellers or business sellers. The definitions of grouping the sellers will be determined by further analysis.

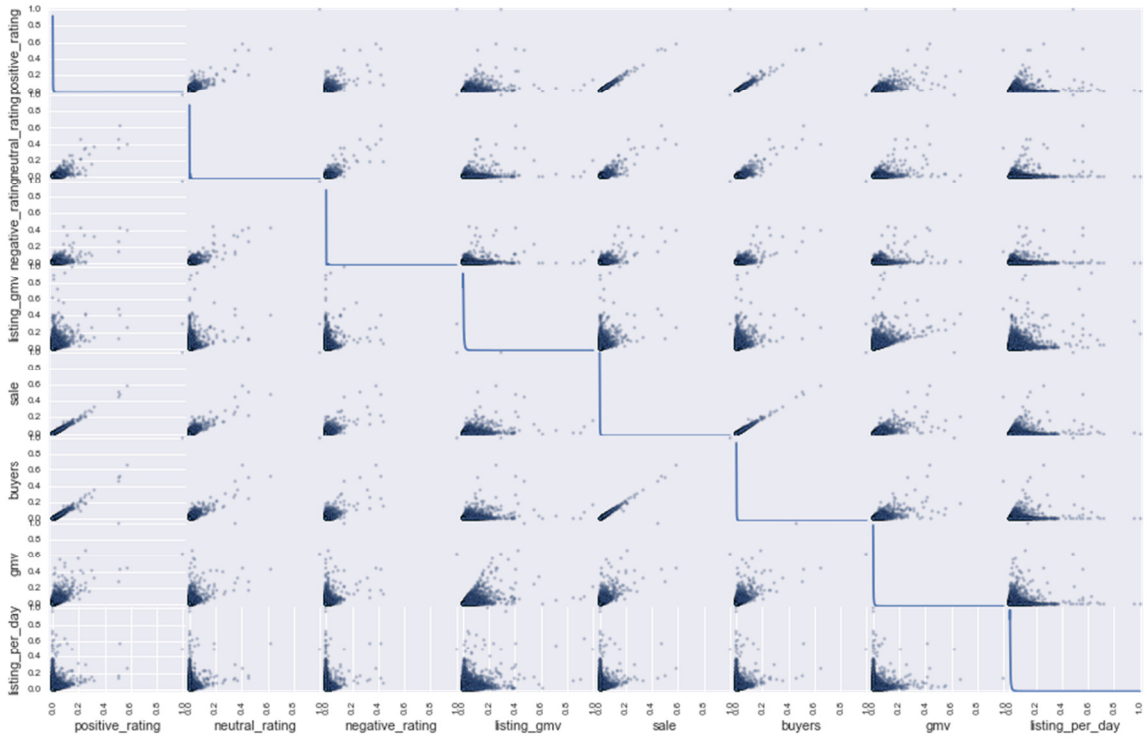

Figure 1 Feature correlations

Algorithms and Techniques

Since it is shown that only 8% of the dataset contains non-zero data, we can assume that the rest 92% can be assumed to indicate a group of new or inactive users. These non-active users could be separated into subgroups based on the feature 'time_on_site,' but I leave that task for a future analysis. In this project, I focus on segregating the active users applying unsupervised clustering analysis.

The algorithms used in this report are mainly in two folds; 1) principal component analysis (PCA) and 2) clustering analysis. Through PCA, the eleven features of dataset will be reduced to a manageable number and be used for clustering analysis. The detailed procedures of feature selection and dimensionality reduction will be discussed in the subsequent Data Processing section.

With reduced features via PCA, several clustering techniques were applied. Despite I chose to analyze 8% of the total data set, the number of entry exceeds 97,000. Most of sklearn clustering analysis algorithms except K-means clustering analysis either took very long time to process or crashes the iPython Notebook session when the data with greater than 50,000 entries were used as input. This will be covered later in the Results section.

Benchmark

If the number of clusters is unknown or labels for clusters are unknown, evaluation after clustering analysis must be performed. The Silhouette Coefficient is an example of the evaluation, where a higher Silhouette Coefficient score corresponds to a model with better defined clusters. The Silhouette Coefficient s for a single sample is then given as:

$$s = \frac{b - a}{\max(a, b)}$$

Where a is the mean distance between a sample and all other points in the same class and b is the mean distance between a sample and all other points in the next nearest cluster. (http://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient)

3. Methodology

Data Preprocessing

As presented in the previous section, the dataset contains 11 features. By using the principal component analysis (PCA), we can reduce the number of features to a reasonable number without losing too much of the integrity of the data. It is an engineer's discretion to select number of principal components after process the original data using PCA, but it is conventional to select up to three PCs as three dimensional data can be easily visualized and therefore its clusters can also be visually inspected and assessed.

1. Feature selection
   Before applying the PCA directly to the original dataset, however, it is recommended to remove or reduce the number of columns by understanding the meaning of each column and the correlations between columns. If any number of columns are strongly correlated, than they can be combined into a single feature. For the dataset in this study, the first and second column, ID and Install Date could be dropped as they only represents random identification numbers of sellers and the usage start date of each seller. Three rating columns (positive, neutral and negative) were initially considered to be irrelevant for analysis because not all active sellers receive ratings and they are given by their buyers and hence they tend to be somewhat inconsistent, but they were added to the final assessment of PCA. The columns 'time_on_site' and 'listing' were combined into a new column called 'listing_per_day' to provide how active and how frequent the seller list the materials onto the app.
   Now the original 11 features are reduced to 8. (Table 2)

Table 2 List of features after feature selection

positive_rating	neutral_rating	negative_rating	listing_gmv	sale	buyers	gmv	listing_per_day

2. Feature scaling
   As the final step before the PCA, the data were normalized using sklean MinMaxScaler. This estimator scales and translates each feature individually such that the values range from zero to one.

Implementation

Principal Component Analysis

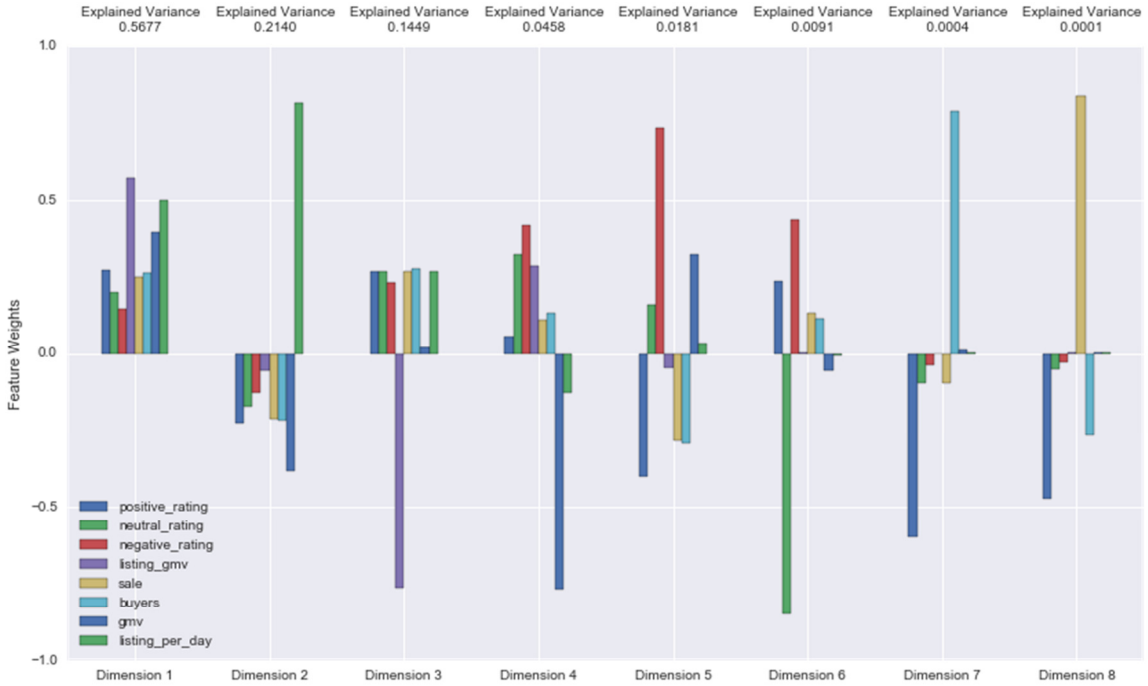

Figure 2 Explained Variance

I applied sklearn PCA algorithm with same number of principal components as outputs and assess the sum of total variance that can explain the original data. As seen in the Figure 2, first five principal components explain 99.05% of the original data suggesting that I need five principal components without losing too much of original data integrity.

Clustering Analysis

Clustering analysis is conducted with three different sklearn clustering algorithms:

a. K-means

b. DBSCAN
c. Agglomerative clustering

Since the new seller was assumed to be the ones without any listings in the given datasets, I need to separate the active users into three remaining groups. (Casual, Business and Top seller category)

Although the provider of this dataset intended to segment the sellers into four groups, such as: (a) New seller, (b) Casual seller, (c) Business seller and (d) Top seller, I have tested the cluster analysis results using silhouette score calculations by varying the number of clusters from two to six.

NOTE: The calculation using sklean's silhouette_score function, however, takes long time and it crashes the iPython Notebook kernel when more than 40000 data entries were provided.

Refinement

As mentioned in the previous section, I have used three different clustering algorithms.

K-means

K-means clustering algorithm is the only one that was able to process all 97,423 dataset and successfully finish the clustering. The silhouette score computation, however, was not able to process all data, so I had to randomly choose 40,000 datasets from the total and computed the score. I have varied the length of data for calculation with 20000, 30000 and 40000 data entries and the silhouette scores are same up to two digits below decimal point.

DBSCAN

As seen in the figure and number of users in each clusters, clusters are not evenly distributed and one or two clusters contain most of the datasets. DBSCAN views clusters as areas of high density separated by areas of low density. It is known that the clusters found using DBSCAN can be any shape, whereas k-means assumes the clusters to be convex shapes. Having negative silhouette scores in many smaller clusters found using k-means may indicate that the ground truth clusters may not be convex shapes.

DBSCAN is known to be good for data which contains clusters of similar density. There are two parameters to the algorithm, min_samples and eps, which define formally what we mean when we say dense. Higher min_samples or lower eps indicate higher density necessary to form a cluster. DBSCAN clustering did not work

well with the entire datasets. I was able to process only 60000 of data using this algorithm.

Hierarchical clustering

Hierarchical clustering algorithm also didn't work well with the entire dataset. I managed to make it work with only 30000 data and it took very long time to process.

4. Results

Model Evaluation and Validation

K-means clustering

After the datasets were clustered using K-Means clustering algorithm, the silhouette scores were computed with randomly selected 20000 data (Table 3).

Table 3 Silhouette score with 20000 random dataset

cluster number	silhouette score	computation time (seconds)
2	0.9057	121
3	0.8563	121
4	0.8377	124
5	0.7660	100
6	0.7049	103

I have checked the silhouette scores with three clusters with varying size of the data used in silhouette score computations (Table). It is shown that scores are similar regardless of the number of data entries, but the computation times increase greatly as more datasets are used in computation.

Table 4 Silhouette score with 3 clusters

data size	silhouette score	computation time (seconds)
30000	0.8551	961
40000	0.8565	2795

It is previously known that there would be three active seller groups within the dataset. The silhouette score for two clusters is greater than three clusters (from **Error! Reference source not found.**). The reason being is possibly because the sizes of elements in each cluster are very highly skewed. The numbers of elements in each cluster excluding the non-active sellers are as following (Table).

Table 5 Number of users in the clusters (30000 data used)

cluster number	number of sellers
0	29004
1	22
2	974

The Figure 3 shows the highly uneven numbers of elements in clustered seller data. The

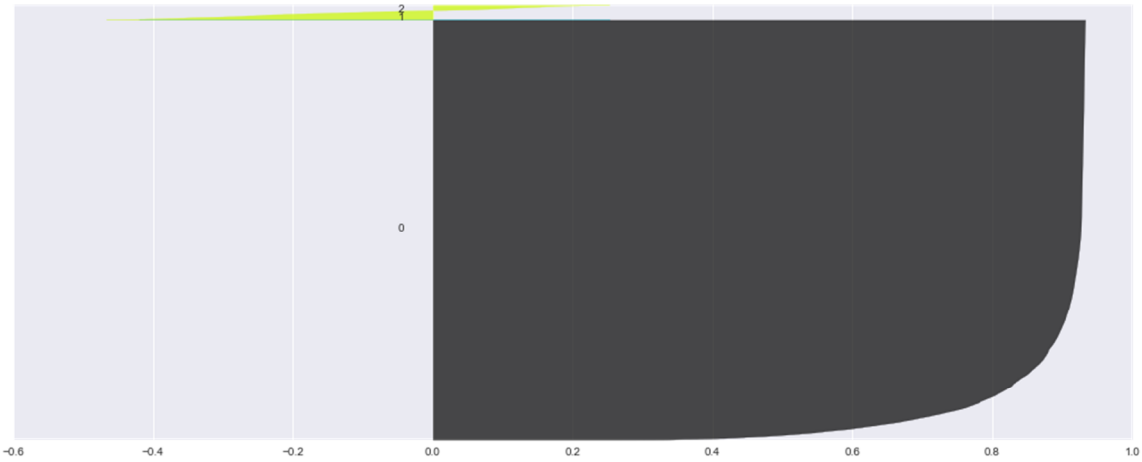

Figure 3 Silhouette score visualization (K-means)

DBSCAN

As mentioned in the previous section, DBSCAN algorithm only allowed 60000 data for its computation. The Problem with DBSCAN with this dataset is that the algorithm generated only one big cluster. Table 6 shows the results from DBSCAN analysis that all 60000 data grouped into one except only two.

Table 4 DBSCAN cluster results (60000 data used)

cluster number	number of sellers
0	59998
-1 (invalid)	2

Hierarchical clustering

As the third method, hierarchical clustering algorithm was applied. This algorithm also limits the number of data entry to be computed, but it took even longer time to run with 20000 and 30000 data entries than DBSCAN. Table 7 shows the computational time when number of clusters was predefined with three.

Table 5 Computation time of Hierarchical cluster algorithm

data size	computation time (seconds)
20000	3503
30000	11407

With three clusters, the silhouette score computes very high of 0.9871, but the element numbers in each cluster seem very trivial. (Table 8)

Table 6 Hierarchical clustering results

cluster number	number of sellers
0	29998
1	1
2	1

Justification

Here are the sample data per each cluster which shows distinct differences between clusters.

The data that were not included in the analysis is non-active and they are labeled as Cluster 3 (New seller). The cluster labels were appended to the original dataset and csv files for each group were independently generated.

time_on_site	positive _rating	neutral _rating	negative _rating	listing	listing_g mv	sale	buyers	gmv	listing_per_ day	clusters
5	0	0	0	40	357	0	0	0	8	1
69	79	6	0	145	20190	85	76	12650	2.101449	2
20	3	1	0	12	205	4	4	18	0.6	0

Average listings per each labeled group were computed, which can explain the characteristics of each cluster.

Cluster 0: Casual seller (avg 8.7 listing)
Cluster 1: Business seller (avg 104.1 listing)
Cluster 2: Top seller (avg 691.6 listing)
Cluster 3: New seller (avg 0 listing)

5. Conclusion

Reflection

In this project, a clustering analysis was conducted on a dataset that contains seller activities from mobile shopping mall app. The features of the data were processed with feature selection and dimensional reduction (PCA) and then various clustering analyses were applied.
The purpose of the clustering analysis is to group the sellers into four so that the company could provide appropriate customer supports for each group.

K-Means, DBSCAN and hierarchical clustering analyses were applied to the dataset, but the computational time for DBSCAN and hierarchical clustering analyses were greatly longer than K-Means and not all of 97000 dataset could be used due to the computer memory issue.

Improvement

For a future task, the active user set which were used for K-Means clustering analysis could be used as a ground truth classification labels and a new supervised classifier model could be constructed. Therefore, any future or unseen dataset can be analyzed with the classifier and a reasonable group could be predicted.