‹ Back to Machine Learning Engineer Nanodegree

Creating Customer Segments

REVIEW

HISTORY

Meets Specifications

Data Exploration

Three separate samples of the data are chosen and their establishment representations are proposed based on the statistical description of the dataset.

Nice job on inferring the customer establishment. Great that you compared each sample against the data statistics to make the inference.

Suggestion: As we can see from the description of the dataset, the data is skewed with large deviation of the mean from median, and thus it makes very good sense to compare the spendings against percentile rather than mean of the data, which can be distorted by outliers.

A prediction score for the removed feature is accurately reported. Justification is made for whether the removed feature is relevant.

You are right. A low R^2 means the feature Fresh cannot be predicted by other features, and it may provide additional information. Hence it may be necessary to identify the customer's spending habit.

Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

Good job to identify the correlated pairs, and nice description on data distributions.

Skewness is a good measure of whether data is normal distributed. We can measure the skewness by using the `scipy.stats.skew()`, which results in 0 for normally distributed data. A skewness value > 0 means positive / right skew, i.e. more weight in the left tail of the distribution, like the data we have here.

Ref: http://docs.scipy.org/doc/scipy-0.13.0/reference/generated/scipy.stats.skew.html

Data Preprocessing

Feature scaling for both the data and the sample data has been properly implemented in code.

Well done on the use of `np.log()` to scale the data. Many algorithms rely on the assumption that data is unskewed like normal distribution. Scaling is one technique to make skewed data more symmetric. Besides using log transformation, other techniques include taking square root of the data, and more advanced ones like Box-Cox transformation:

```
from scipy.stats import boxcox
x_boxcox, _ = boxcox(x)
```

Ref: http://scipy.github.io/devdocs/generated/scipy.stats.boxcox.html

Note: You can also read more about when we should / should not perform feature scaling

here:
http://stats.stackexchange.com/questions/121886/when-should-i-apply-feature-scaling-for-my-data

Student identifies extreme outliers and discusses whether the outliers should be removed. Justification is made for any data points removed.

Awesome job to implement outlier detection and removal with nice justification.

Outlier detection and removal is quite subjective. There are different metrics for outlier detection: Peirce's criterion, Tukey’s test, kurtosis-based, etc. The existence of outliers may affect some clustering algorithms like k-means, and the outliers should be removed. We may also remove outliers to reduce the skewness of data. On the other hand, we should not remove too many outliers, as this reduces our data size. To balance the above two (somewhat conflicting) objectives, it makes very good sense to only remove the data points that you have identified.

Suggestion:

> •Yes, points with 2 outliers - 66, 75, 128; 3 outliers - 154.

It would be good to list all the data points that are considered as outliers for more than one features. We can identify such points with Python’s `Counter` to look for points with counts greater than 1.

Hint: There are altogether five such data points and you have found four of them correctly.

Feature Transformation

The total variance explained for two and four dimensions of the data from PCA is accurately reported. The first four dimensions are interpreted as a representation of customer spending with justification.

Awesome explanation of the principal components. It is noteworthy that here we are more interested in the significant weights. It is perfectly acceptable to flip the positive and negative weights, and the result should not affect the discussion. For example, for dimension 1, the predominant spending is on milk, grocery and detergents_paper. Although they all have negative weights, it is the absolute value that matters, as long as the signs of the weights are identical. Similarly, dimension 2 spends a lot on fresh, frozen, and deli, as they all have strong negative weights. It is interesting to observe the inverse correlation in dimension 3, which shows that spending on fresh is inversely correlated with spending on deli. Here the sign of the weight does matter to indicate the inverse correlation.

Here is just another link on interpretation of PCA result for your reference:
https://onlinecourses.science.psu.edu/stat505/node/54

PCA has been properly implemented and applied to both the scaled data and scaled sample data for the two-dimensional case in code.

Well done on dimension reduction with PCA. If we visualize the data distribution after dimension reduction using:

```
pd.scatter_matrix(reduced_data, alpha = 0.3, figsize = (10,6), diagonal = 'kde')
```

we can see that dimension 1 has a bi-modal distribution that looks very similar to the distribution of Milk, Grocery, and Detergents_Paper after log transformation in Question 3. This is not a coincidence. Milk, Grocery, and Detergents_Paper are the features with large emphasis in the first principal component in Question 5.

Clustering

The Gaussian Mixture Model and K-Means algorithms have been compared in detail. Student’s choice of algorithm is justified based on the characteristics of the algorithm and data.

Nice job on selecting GMM with justifications. The soft assignment property of GMM makes it a good fit for this problem, where we don't see very clear boundaries between the clusters.

Several silhouette scores are accurately reported, and the optimal number of clusters is chosen based on the best reported score. The cluster visualization provided produces the optimal number of clusters based on the clustering algorithm chosen.

Indeed two clusters give the best score. Intuitively, given this small data size, we usually do not want to have too many clusters. So dividing data into two clusters is quite reasonable here.

Note: The Silhouette score is calculated based on Euclidean distance, which is also the proximity measure used in k-means clustering. For GMM whose assignment is probabilistic, Silhouette score may not be the best metric, and alternative metrics could be Bayesian Information Criterion or Akaike information criterion.

Ref: http://scikit-learn.org/stable/auto_examples/mixture/plot_gmm_selection.html#example-mixture-plot-gmm-selection-py

The establishments represented by each customer segment are proposed based on the statistical description of the dataset. The inverse transformation and inverse scaling has been properly implemented and applied to the cluster centers in code.

Sample points are correctly identified by customer segment, and the predicted cluster for each sample point is discussed.

Conclusion

Student correctly identifies how an A/B test can be performed on customers after a change in the wholesale distributor's service.

Nice discussion. Intuitively the segment buying more perishable product may be affected more by the change in schedule due to their reliance on fresh product, as you have pointed out.

Suggestion:
Besides the intuitive argument, we would like to confirm our hypothesis through A/B test, which also helps the distributor make decision more effectively. In particular, there are some important questions that need to be addressed on the design of A/B test:

- How are the group of customers defined?
- Should we perform A/B test on each group individually?
- In A/B test, we normally have a control set and a variation set. How can we determine the control and variation set?

Student discusses with justification how the clustering data can be used in a supervised learner for new predictions.

Comparison is made between customer segments and customer 'Channel' data. Discussion of customer segments being identified by 'Channel' data is provided, including whether this representation is consistent with previous results.

Very good discussion. We can see that GMM has done a good job to cluster points far away from the center. It may be hard to cluster the data points in the central region correctly, as we can see that there are a lot of crossing-over points. Intuitively, this could be due to the the different types / scales of operation. For example, a grocery store may have a spending pattern more similar to 'Hotels/Restaurants/Cafes', and thus it borderlines between the two segments. In such cases, the probabilistic / soft assignment of GMM can be quite useful: it gives us a confidence on how well we can trust the clustering result. To view the probability that a point belongs to a cluster, we can use the `predict_proba` of GMM instead of `predict` when clustering the data.

Ref: http://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMixture.predict_proba

DOWNLOAD PROJECT

Student FAQ