UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Congratulations! Your revised submission is perfect, and you have done a great job to successfully completed this project on clustering. Keep up your excellent work!

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.

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

You are right. A high R^2 means features such as Detergents_Paper and Grocery can be predicted by other features and does not provide much additional information. Thus they are not necessary to identify customers' spending habits.

On the other hand, features having low R^2 (e.g. Deli and Milk) means they cannot be predicted well by other features, and 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.

> The data is not normally distributed for any of the features, and are right-skewed (data points are close to the left axis), as seen in the graphs above.

It is great that you have noted the skewness in data. 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 five data points that you have identified.

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. 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.

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, regardless of whether we are using the K-means or GMM method.

> Also the best scores are close to each other with Kmeans slightly higher than GMM (0.426 vs. 0.412)

The Silhouette score is calculated based on Euclidean distance, which is also the proximity measure used in K-means clustering. So K-means may have some unfair advantage here. 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.

> Intuitively, it seems that Cluster 0 customers will be more affected by this change, and Cluster 1 may be able to tolerate the delivery schedule change.

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.

> Given Cluster 0 (larger) customers generates more revenue for the distributor, the distributor should do separate A/B tests for the two clusters. First do A/B tests for high revenue/high volume customers of Cluster 0. Then do the same separately for corresponding customers of Cluster 1.

Your proposed implementation of A/B test is great. The key is to conduct the A/B test on each segment independently to identify the segment that react significantly to the change.

As a suggestion, if we want our test result to be more accurate, we can pick customers that are more representative of each segment. For example, we can select the customers that are close to the centers of each cluster.

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

> The Labels for each customer would be the Customer Segment

Exactly. A common technique in feature engineering is to use the result of unsupervised learning for next stage supervised learning, and here we can use the clustering result as target label for our training data.

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.

Nice discussion. We can see from Cluster Visualization that the K-means defines the boundary as a vertical line around 0.5 along Dimension 1. This generally agrees with the channel labels. For points at the two edges far away from the decision boundary, they are generally clustered correctly. For points near the boundary, as their spending patterns may be very close, it is hard to define an exact cluster assignment. 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 might be useful: it gives us a confidence on how well we can trust the clustering result.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ