UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

Excellent job with your analysis and really like all of the extra analysis in this project! Very impressed with your coding ability and wish you the best of luck in your future!!

If you would like to dive in deeper into Machine Learning material, here might be some cool books to check out

- An Introduction to Statistical Learning Code is in R, but great for understanding
- elements of statistical learning More mathy
- Python Machine Learning I have this one, great intuitive ideas and goes through everything in code.

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 additions here. I would recommend explicitly comparing the these samples purchasing behaviors to the descriptive stats of the dataset. As stating "*Sample 0 has the highest Delicatessen spending of all customers and moderate to high spending on all other categories except Detergents & Paper,* " wouldn't necessarily give a good representation of how this customer compares to the entire dataset as a whole. Thus a good idea here would be to compare each product to the mean / median / quartiles.

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

Nice work here, as you have addressed the previous reviewers concerns. Thus if we have a high r^2 score(high correlation with other features), this would actually not be good for identifying customers' spending habits(since the customer would purchase other products along with the one we are prediction). Therefore a negative / low r^2 value would represent the opposite as we could identify the customers specific behavior just from the one feature.

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

Good with the correlation between features and nice with the data distribution and the skewed right, thus we typically see this type of distribution when working with sales / income data.

Pro Tip: We can also easily see correlation with a seaborn correlation plot

```
import seaborn as sns
sns.corrplot(data)
```

Data Preprocessing

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

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

Excellent coding here and great job discovering the indices of the five data points which are outliers for more than one feature of `[65, 66, 75, 128, 154]`. And good idea to remove these, since these data points can highly affect our clustering or PCA algorithm.

A good learning experience here would be to run clustering with and without these outliers and see how the silhouette scores change.

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.

Nice job with the percent of variance explained.

- As with two dimension we can easily visualize the data(as we do later)
- And with four components we retain much more information(great for new features)

I would hesitant to state each component as one single customer since PCA deals with the variance of the data and the correlation between features. Therefore in terms of customers, the first component would represent that we have some customers who purchase a lot of Milk, Grocery and Detergents_Paper products while other customers purchase very few amounts of Milk, Grocery and Detergents_Paper, hence spread in the data.

Pro Tip: You can also visualize the percent of variance explained to get a very clear understanding of the drop off between dimension. Here is a some starter code, as np.cumsum acts like `+=` in python.

```
import matplotlib.pyplot as plt
x = np.arange(1, 7)
plt.plot(x, np.cumsum(pca.explained_variance_ratio_), '-o')
```

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

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.

Good justification for your choice in K-Means. You could actually try both and see the difference. As the main two differences in these two algorithms are the speed and structural information of each:

Speed:

- K-Mean much faster and much more scalable
- GMM slower since it has to incorporate information about the distributions of the data, thus it has to deal with the co-variance, mean, variance, and prior probabilities of the data, and also has to assign probabilities to belonging to each clusters.

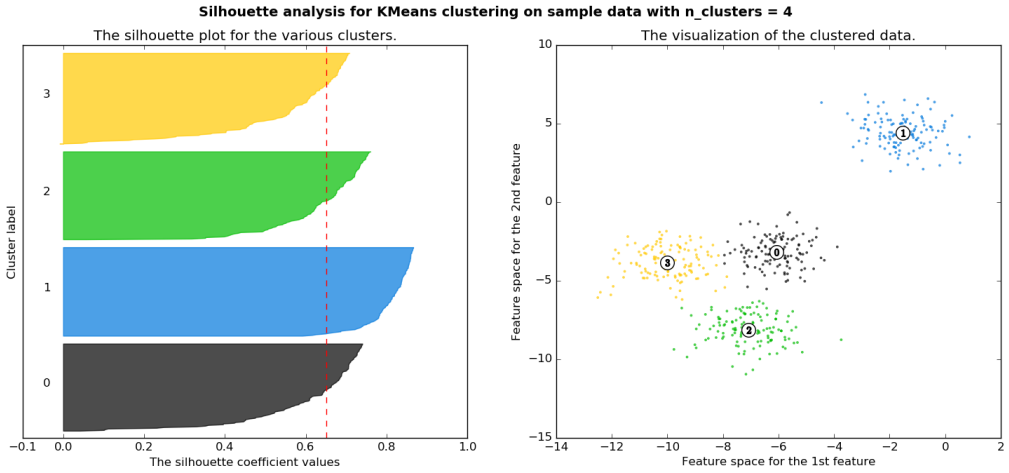
Structure:

- K-Means straight boundaries (hard clustering)
- GMM you get much more structural information, thus you can measure how wide each cluster is, since it works on probabilities (soft clustering)

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.

Great coding here with your for loop and function. As we can clearly see that K = 2 gives the highest silhouette score. Another cool interpretation technique for silhouette scores is like this

(http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html)



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.

Love the box and whisker plot included below, as this is excellent justification for the type of establishment for these cluster centroids.

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

Solid analysis for your predictions!! If we were to use GMM, we could also check out the probabilities for belonging to each cluster with the use of `predict_proba()`

Conclusion

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

"*test the new delivery schedule on another slice of one or both of the segments*"

We definitely need to run separate A/B tests on each segment independently. As if we were to use all of our customers we would essentially have multiple variables(different delivery methods and different purchasing behaviors).

✓

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

"*New data would need to be logarithmically scaled and run through the already-fit clusterer_6d instance to generate a segment prediction (our target variable).*"

We could also use a subset of the newly engineered features as new features(great for curing the curse of dimensionality). As PCA is really cool and seem almost like magic at time. Just wait till you work with hundreds of features and you can reduce them down into just a handful. This technique becomes very handy especially with images. There is actually a handwritten digits dataset, using the "famous MNIST data" where you do just this and can get around a 98% classification accuracy after doing so. This is a kaggle competition and if you want to learn more check it out here KAGGLE

✓

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.

Your sample data points are very interesting, probably because you have chosen the argmax.

DOWNLOAD PROJECT

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Rate this review

Student FAQ