UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

This is a very solid analysis here and very impressed with your answers. You have an excellent grasp on these unsupervised learning techniques. Hopefully you have learned a bunch throughout these submissions. 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.

Good ideas for potential establishments, I would recommend comparing the purchasing behavior of **each** sample to the descriptive stats of the dataset(as you have done with customer : 1). As stating "*this customer spent less amount of money on Detergents_Paper and decent amount of money on other stuffs*" 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.

```
display(samples - np.round(data.mean()))
display(samples - np.round(data.median()))
```

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

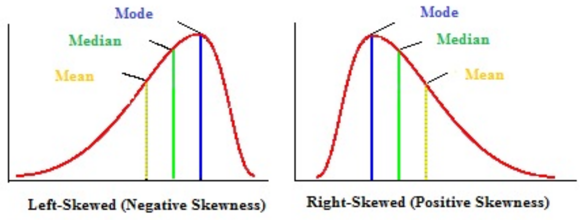
Great! Thus if we have a high r^2 score(high correlation with other features), this would not be good for identifying customers' spending habits(since the customer would purchase other products along with the one we are predicting, as we could actually derive this feature from the rest of the features). Therefore a negative / low r^2 value would represent the opposite as we could identify the customer's 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.

Great job capturing the correlation between features. We could actually get some more insight by looking at numerical correlation by adding it to the plot with

```
axes = pd.scatter_matrix(data, alpha = 0.3, figsize = (14,8), diagonal = 'kde')
corr = data.corr().as_matrix()
for i, j in zip(*np.triu_indices_from(axes, k=1)):
    axes[i, j].annotate("%.3f" %corr[i,j], (0.8, 0.8), xycoords='axes fraction', ha='center', va='center')
```

And good ideas regarding the data distributions. A skewed right distribution is a good idea. As we can actually get an idea of this from the basic stats of the dataset, since the mean is above the median for all features. We typically see this type of distribution when working with sales or income data.


Mode
Median
Mean
Left-Skewed (Negative Skewness)
Mode
Median
Mean
Right-Skewed (Positive Skewness)

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.

Nice work discovering the indices of the five data points which are outliers for more than one feature of `[65, 66, 75, 128, 154]`.

I would recommend also providing some reasoning in why you don't remove these outliers. Another idea would be to actually run our future analysis with these data points removed and with these included and see how the results change.

(http://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/)
(http://graphpad.com/guides/prism/6/statistics/index.htm?stat_checklist_identifying_outliers.htm)

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 work with the cumulative explained variance for two and four dimensions.

- 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 recommend also mentioning the Deli in the forth component here as well for a more complete answer. But you have the right ideas. To go even further here with the interpretation of the PCA components:

- In terms of customers, since PCA deals with the variance of the data and the correlation between features, 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 comparison here. 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.

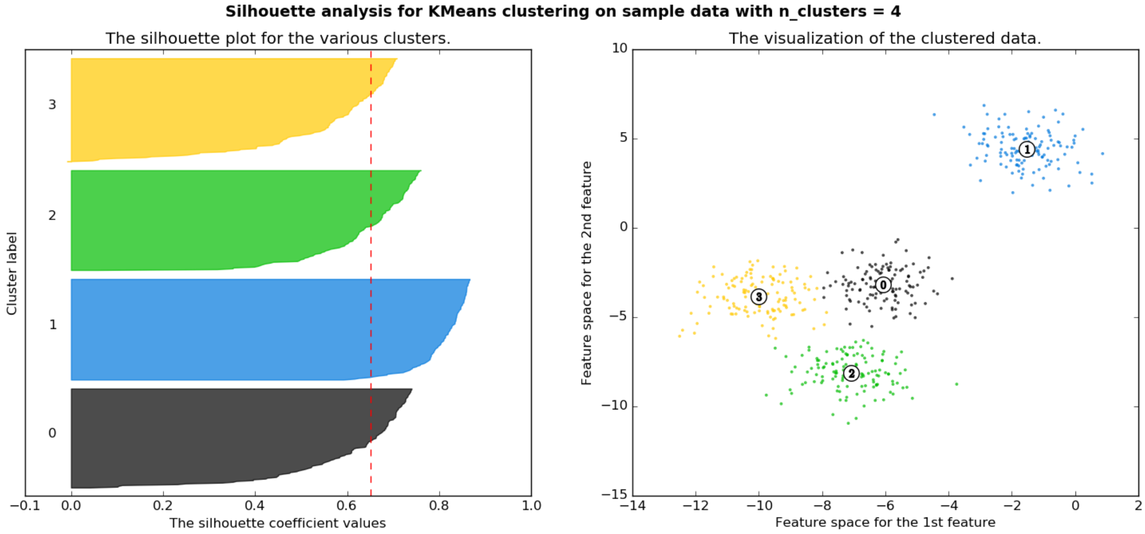
Structure:

- 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.

Impressive coding and nice work as we can clearly see that K = 2 gives the highest silhouette score. Another cool interpretation method for Silhouette score 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.

I won't mark this as *Requires Changes* to be consistent with the previous reviewers, but you really haven't answered the question

WHAT SET OF ESTABLISHMENTS COULD EACH OF THE CUSTOMER SEGMENTS REPRESENT?

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

Great justification for your predictions by comparing the purchasing behavior of the sample to the purchasing behavior of the cluster centroid! Very impressive!

Maybe check out the distance between the sample point and the cluster center

```
for i, pred in enumerate(sample_preds):
    print "Sample point", i, "predicted to be in Cluster", pred
    print 'The distance between sample point {} and center of cluster {}:'.format(i, pred)
    print (samples.iloc[i] - true_centers.iloc[pred])
```

Conclusion

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

Your comment of "*So we should further divide those segments into a control group which would have an unchanged delivery schedule and a test group with the new delivery schedule*" is key here. We should run separate A/B tests for each cluster independently(or maybe multiple in each cluster, as you have mentioned). As if we were to use all of our customers we would essentially have multiple variables(different delivery methods and different purchasing behaviors).

After this the wholesale distributor can look at the p values for the tests that the null hypothesis that the difference between the chosen metric between the control group and the experiment is zero. If the p value for segment 0 A/B test is smaller, it means segment 0 customers are affected more by the change.

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

Nice idea to use the cluster assignment as new labels. Another cool idea would be to use a subset of the newly engineered PCA components as new features(great for curing the curse of dimensionality). 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.

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