UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

All in all this is a very good first submission and you only have some minor adjustments to make in order to meet all the specs. You're very close to completion, so keep at it! 😃

Data Exploration

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

re: Question 1
You're on the right track here, but make sure to refer explicitly to the overall category spending stats (mean, median) in the discussion, and mention anything that stands out.

For example, Customer 0 has high spending in relation to the category mean & median for all categories except Fresh & Frozen. It could be a big chain retailer.

You can use the below code to help come up with points to address in your answer...

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

Suggestion: pandas plotting
To visualize the comparison of sample spending to the category means and/or medians, you could also try pandas plotting...

```
((samples-data.median()) / data.median()).plot.bar(figsize=(10,4), title='Samples compared to MEDIAN', grid=True)
((samples-data.mean()) / data.mean()).plot.bar(figsize=(10,4), title='Samples compared to MEAN', grid=True);
```

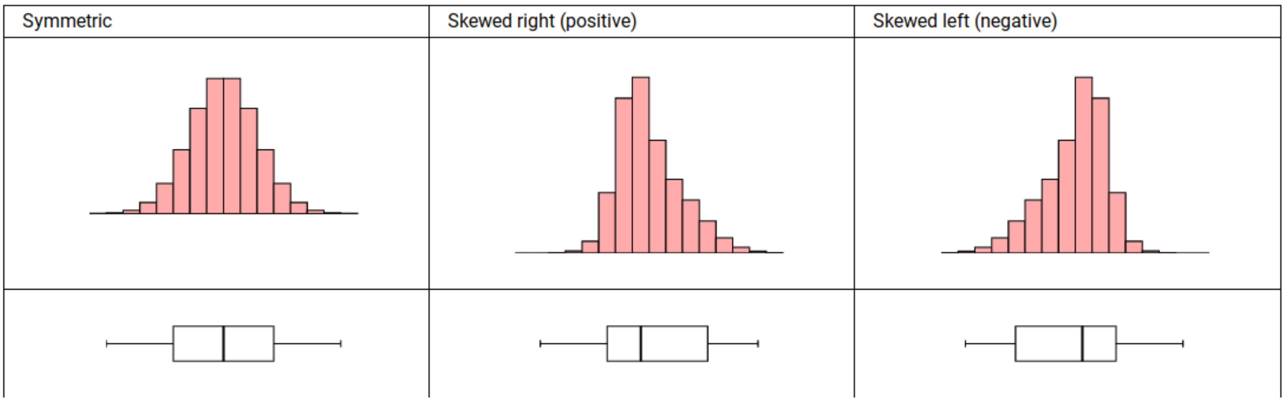
A prediction score for the removed feature is accurately reported. Justification is made for whether the removed feature is relevant.

Fantastic job predicting the features, and determining which ones might or might not be relevant. Detecting redundant features is a common step during feature selection.

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

re: Question 3
Excellent work spotting the correlated features, but be sure to also describe the distribution of the features themselves (see plots on the diagonal of visualization rather than the scatterplots):

- Does the data appear normally distributed?
- Is there any skewness?



(note that a normal distribution would be symmetric)

Data Preprocessing

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

Nice job scaling the data with a very concise code implementation — this will help our data appear more normally distributed and more appropriate to use with a variety of machine learning techniques.

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

re: Question 4
The code here to detect Tukey outliers is solid, but it looks like the answer is missing one of the multiple-category outliers (there are 5 of them in total).

Instead of manually looking for them, you could try to identify them programmatically using a Counter.

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.

Excellent work reporting the cumulative variance and identifying what the category weights in each dimension represent!

PCA deals with the variance of the data and the correlation between features. For example, the first component shows that we have a lot of variance in customers who purchase **Milk, Grocery & Detergents_Paper** — customers with *HIGHER* values in the first component purchase a lot of these 3 categories, while those with *LOWER* values in the component purchase very little.

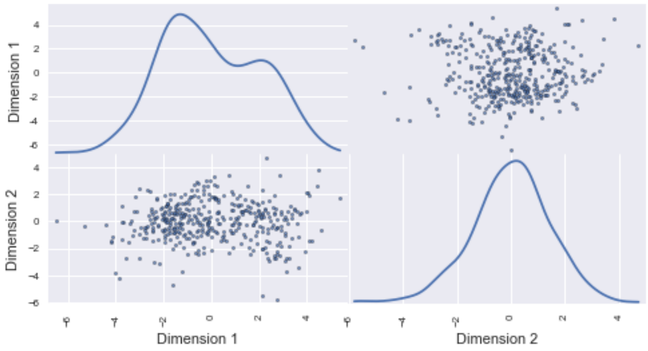
For more on PCA, you can also check out this nice visualization, as well as this PCA tutorial.

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

Great job implementing the dimensionality reduction!

If we view a scatter matrix of the reduced data, we can see 2 humps in the 1st Dimension that seem to indicate the presence of 2 distinct groups within the distribution...

```
# Produce a scatter matrix for pca reduced data
pd.scatter_matrix(reduced_data, alpha = 0.8, figsize = (8,4), diagonal = 'kde');
```



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 here discussing the clustering methods. There really is no *right* or *wrong* approach to use with our current dataset, as they will be produce somewhat similar clusters. Choosing K-Means for its simplicity is a perfectly good idea. Here are some of the key points to consider...

Speed/Scalability:

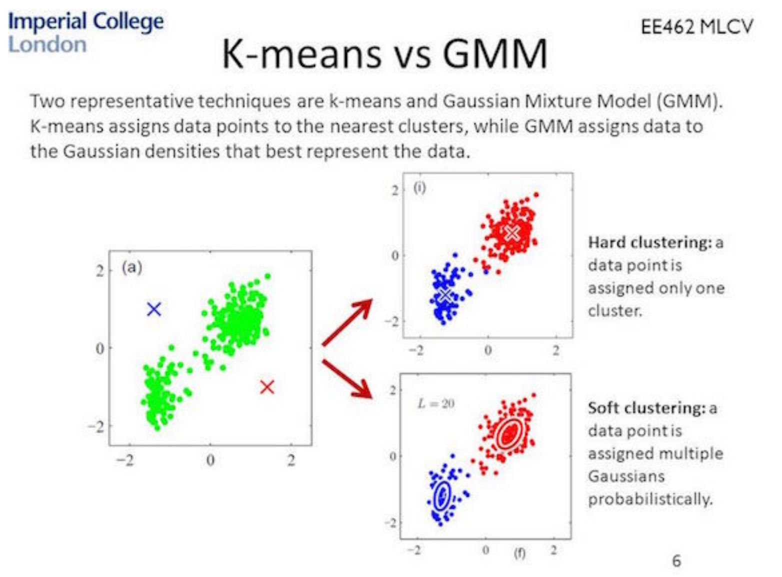
- K-Means faster and more scalable
- GMM slower due to using information about the data distribution — e.g., probabilities of points belonging to clusters.

Cluster assignment:

- K-Means hard assignment of points to cluster (assumes symmetrical spherical shapes)
- GMM soft assignment gives more information such as probabilities (assumes elliptical shape)

(read more on the differences of the methods, and how they are related)

Note: K-Means vs GMM shape
A big drawback with KMeans is that it assumes the groups are spherical (globular) shapes that are symmetrical, which don't always occur with real data. GMM assumes the groups to be elliptical...



Imperial College London

EE462 MLCV

K-means vs GMM

Two representative techniques are k-means and Gaussian Mixture Model (GMM). K-means assigns data points to the nearest clusters, while GMM assigns data to the Gaussian densities that best represent the data.

Hard clustering: a data point is assigned only one cluster.

Soft clustering: a data point is assigned multiple Gaussians probabilistically.

6

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.

Fantastic work looping through the cluster sizes to determine the best score, and also setting a random state on the `clusterer` to make your results reproducible. Very few students do this. Kudos! 😎

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.

Great discussion of the segment centers, although it would help here to make explicit references to the category spending stats (mean, median) — the clusters appear to be generally split on spending in the 1st pca dimension, with the cluster centers largely characterized by having above or below average spending in **Milk, Grocery, Detergents_Paper**.

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

re: Question 9
You're on the right track here, but try to evaluate the cluster predictions for each of the 3 sample points by looking at their spending in the 6 categories in comparison to the category spending of their predicted clusters.

Here's an example of how the discussion could be structured...

> "For Sample *<< INSERT NUMBER >>*, the values for **Grocery, Milk, & Detergents_Paper** are above average.
>
> This mirrors the category spending for the Segment *<< INSERT NUMBER >>* center, so the predicted cluster seems to be consistent with the sample."

Conclusion

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

Terrific job pointing out that we can test the segments separately — the customers in each cluster might be affected differently by a delivery change, so to test all the customers properly we'd essentially need to be running multiple A/B tests.

If interested, here's more reading on A/B testing in the real world...

- When A/B testing shouldn't be trusted
- Pitfalls of A/B testing
- Great Data Skeptic podcast on "p-hacking" and other shady practices to watch out for in A/B tests

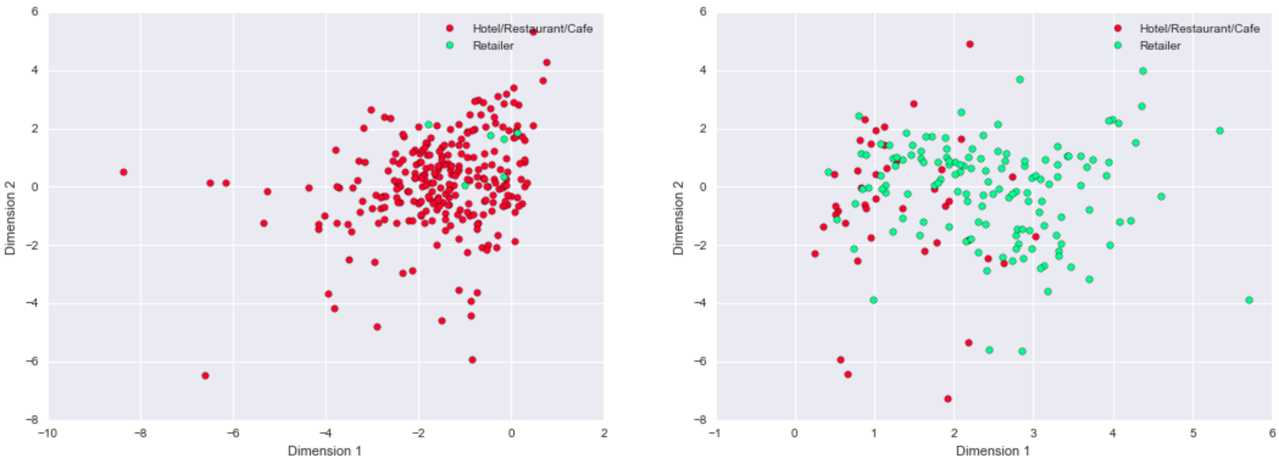
Student discusses with justification how the clustering data can be used in a supervised learner for new predictions.

Great job identifying how we can use the cluster labels! The basic idea is that we can perform feature engineering and use the output of an unsupervised learning analysis as an input to a new supervised learning analysis.

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 examination of the 'Channel' data in relation to our learned clustering — although there is disagreement with some of the data points, the overall alignment is actually pretty good.

To give another look at how well the 'Channel' data and segments are aligned, you can see the 2 clusters from a K-Means implementation plotted separately below (no outliers removed from data)...



NOTE: It looks like you actually removed one of your 3 sample points from the analysis by including it in `outliers`. Ideally you'd either keep the data point in the analysis or choose a different sample point.

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

⊙ Watch Video (3:01)

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

RETURN TO PATH