

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!  

Requires Changes

5 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.

Terrific discussion of the samples in relation to the overall category spending stats! Getting a closeup view of customers should help as we [apply machine learning to optimize food delivery](#).

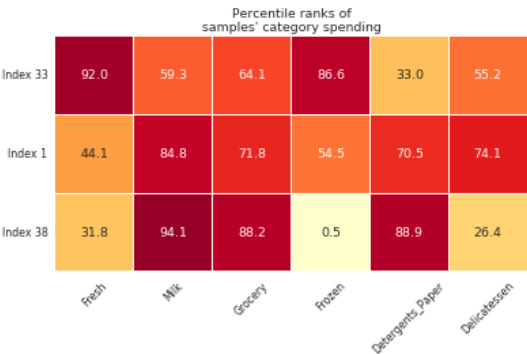
Suggestion: look at percentile ranks

As we'll see later, the distribution of our customers' spending data has some large skew, so another thing you could try is looking at the category spending percentile ranks with a [heatmap](#):

```
import matplotlib.pyplot as plt
import seaborn as sns

# look at percentile ranks
pcts = 100. * data.rank(axis=0, pct=True).iloc[indices].round(decimals=3)
print pcts

# visualize percentiles with heatmap
sns.heatmap(pcts, annot=True, linewidth=.1, vmax=99, fmt='.1f', cmap='YlOrRd', square=True, cbar=False)
plt.yticks([2.5, 1.5, .5], ['Index '+str(x) for x in indices], rotation='horizontal')
plt.title('Percentile ranks of \nsamples\' category spending')
```



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](#).

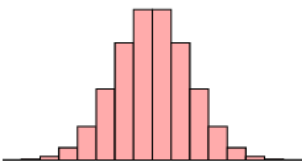
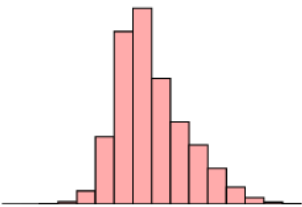
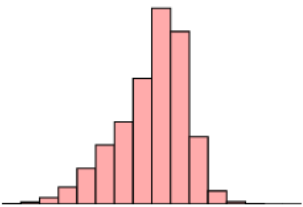
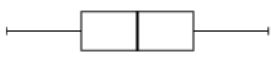
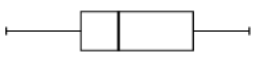
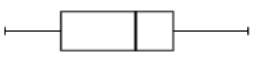
Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

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](#)?

Symmetric	Skewed right (positive)	Skewed left (negative)
		
		

(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

Your code here to detect Tukey outliers is solid, but be sure to identify all 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.

re: Question 5

1) Nice work transforming the data, but make sure to provide the correct values for the cumulative variance of both the first 2 & first 4 dimensions — for example, you can output the values programmatically with `cumsum()` ...

```
print pca_results['Explained Variance'].cumsum()
```

2) The discussion should also describe each of the first 4 dimensions in terms of category spending. When describing each of the components, mention both the *direction* as well as the *magnitudes* of any significant weights.

For example, something like...

*"The first principal component is made up of large positive weights in Detergents\_Paper, Grocery & Milk. It also correlates with a decrease in Fresh and Frozen. This pattern might represent spending in household staples products that are purchased together."*

PCA deals with the variance of the data and the correlation between features. 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 (e.g., Retailers) purchase a lot of these 3 categories, while those with *LOWER* values in the component (e.g., Restaurants) purchase very little.

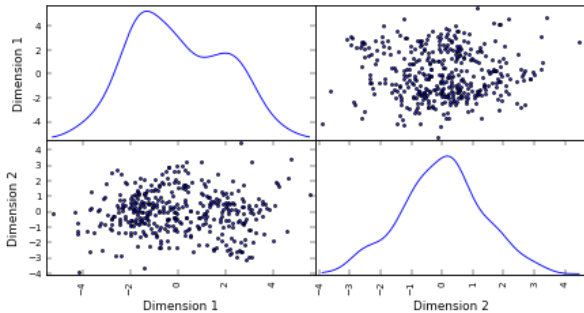
For more on PCA, you can also check out this [nice visualization](#), as well as this [PCA tutorial](#).

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 touching on 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)

You can also read more on the [differences of the methods](#), and [how they are related](#).

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 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 with reference to the category spending stats — 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 median 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

Try to evaluate the cluster predictions for each of the 3 sample points by looking at their category spending in comparison to the category spending of their predicted clusters.

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

"For Sample point <<INSERT>>, the values for Grocery, Milk, & Detergents\_Paper are above average.

This mirrors the category spending for the Segment <<INSERT>> 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.

re: Question 10

Great instincts here in speculating that the segments of customers might be affected differently by a delivery schedule change, but how would we actually prove that?

Focus on [describing a segmented A/B test](#) that changes a single variable (eg, delivery schedule) for one group of customers (the "A" or "control" group) and compares them to a second group that looks *exactly the same* except that it gets no change (the "B" or "treatment" group).

Knowing that we have clusters of customers that seem to be different, how could we use that knowledge to help structure the A/B test?

Further reading on A/B testing here if you're interested:  
<http://multithreaded.stitchfix.com/blog/2015/05/26/significant-sample/>

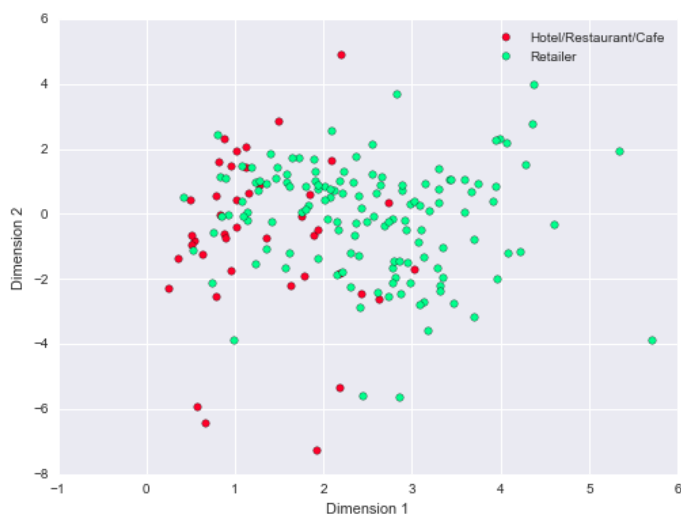
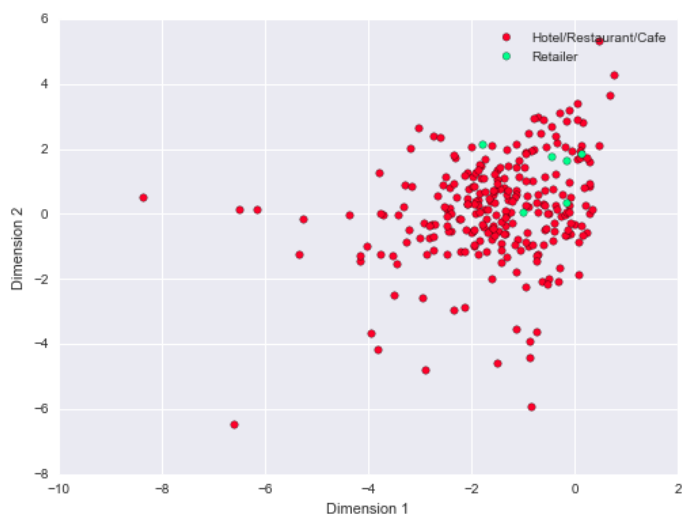
Student discusses with justification how the clustering data can be used in a supervised learner for new predictions.

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)...



 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](mailto:review-support@udacity.com) and include the link to this review.

RETURN TO PATH

---

[Student FAQ](#)