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

Overall this is a very good submission, with solid code implementations and answers in most sections. You do have some adjustments to make in order to meet all the specs, but you're very close to completion. 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 in the discussion, and mention anything that stands out.

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())
```

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

Fantastic job predicting all the features, and determining that Grocery might not be relevant! Detecting redundant features is a common step during feature selection.

It's not part of the spec, but if you repeated the regression 100x and averaged the scores to smooth out variation in decision tree formation and train/test splits, you might see values similar to this...

```
Fresh: -0.6662

Milk: 0.1087

Grocery: 0.6561

Frozen: -1.447

Detergents_Paper: 0.6452

Delicatessen: -2.7253
```

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

Great work spotting the correlations and describing the distributions! The feature distributions appear to be skewed right/positive, or lognormal, with a mean greater than the median.

And for a closer look at the correlations we can also use `data.corr()` with a heatmap...

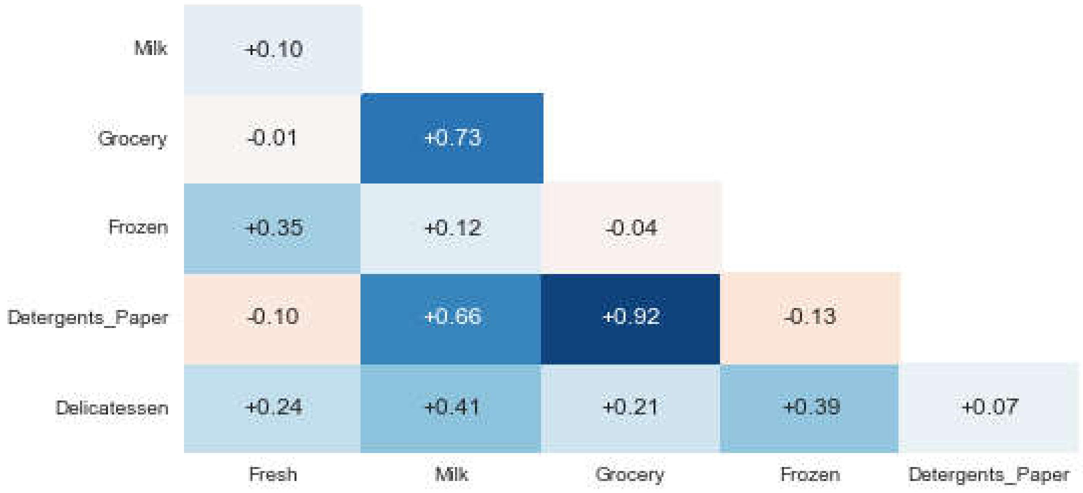
```
import seaborn as sns
import matplotlib.pyplot as plt

# get the feature correlations
corr = data.corr()

# remove first row and last column for a cleaner look
corr.drop(['Fresh'], axis=0, inplace=True)
corr.drop(['Delicatessen'], axis=1, inplace=True)
```

```
# create a mask so we only see the correlation values once
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask, 1)] = True

# plot the heatmap
with sns.axes_style("white"):
    sns.heatmap(corr, mask=mask, annot=True, cmap='RdBu', fmt='+.2f', cbar=False)
```



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.

Good work getting the outliers, and justifying their removal from the dataset!

Outlier detection and removal can be very subjective, and while we definitely benefit by reducing the skewness of the data and avoiding effects on k-means clustering, there are also costs.

We don't want to reduce our dataset size too much, given that it's already quite small. Also, outliers can contain valuable information that we may want to analyze. In this case, those outliers may be very important customers, and we should be careful not to neglect them.

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 *high* values in the first component purchase a lot of these 3 categories, while those with *low* values in the component purchase very little.

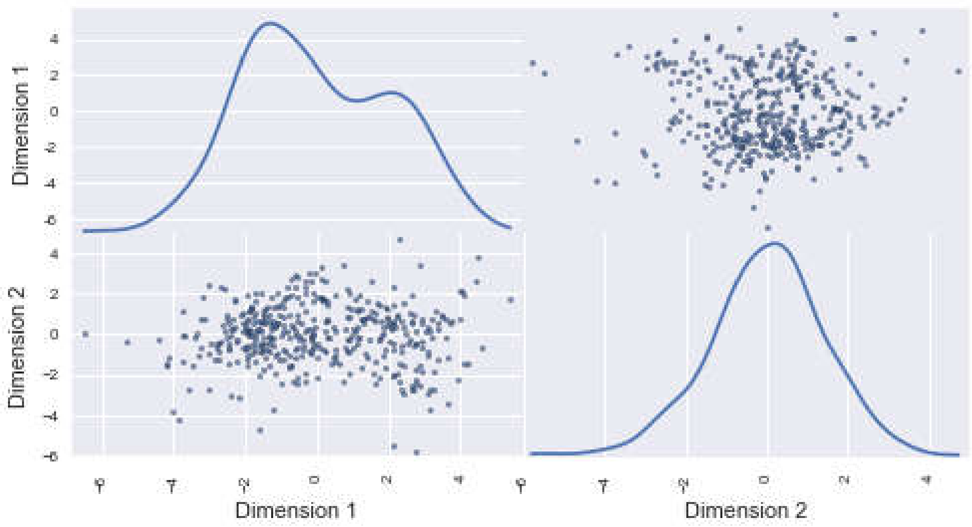
For more on PCA, you can also check out this nice visualization.

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');
```



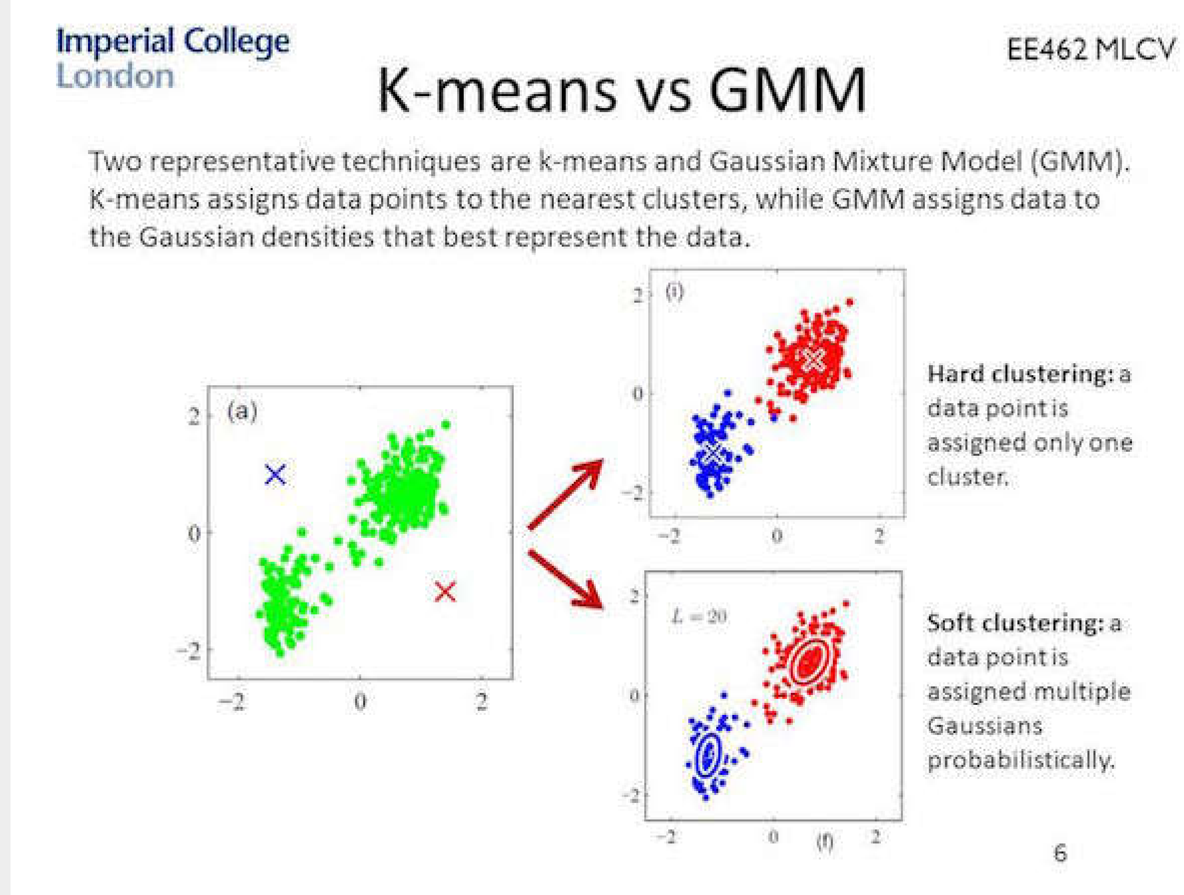
(remember, **Dimension 1** generally correlates with **Milk, Grocery, Detergents_Paper*

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 work discussing the approaches, including how they differ in speed (scalability) and in using soft vs hard assignment of points to clusters!

Note that 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 here 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.

re: Question 8
Nice discussion here of the segment centers with reference to the category spending stats, but it looks like you just have a typo in the answer — the answer refers to both segments as "Segment 0"...

> Segment 0 has above-average spendings on Fresh and Frozen, below-average spendings on Milk, Grocery and Detergents & Paper and somewhat average spendings on Delicatessen. This could be a profile of restaurants selling mostly fresh food but also a bit of frozen food as well as Delicatessen.
> Segment 0 has above-average spendings on Milk, Grocery and Detergents & Paper, low spendings on Fresh food and average spendings on Delicatessen. This could be the profile of a retail store.

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

re: Question 9
Try to quickly mention how the category spending of each of the 3 sample points seem to be consistent (or not) with their predicted clusters' category spending.

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.

re: Question 10
Great thoughts 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 an 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? (or, multiple A/B tests)

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

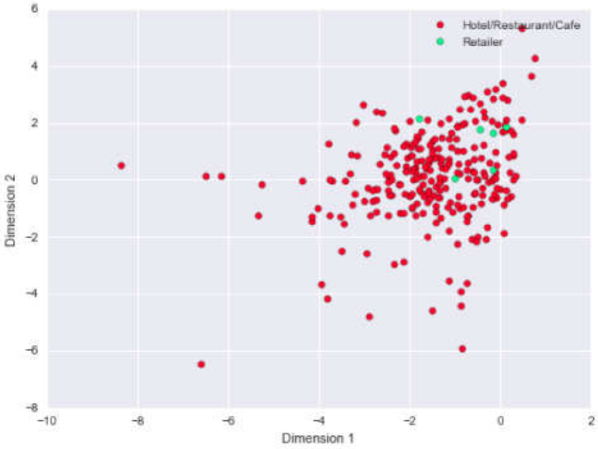
Student discusses with justification how the clustering data can be used in a supervised learner for new predictions.

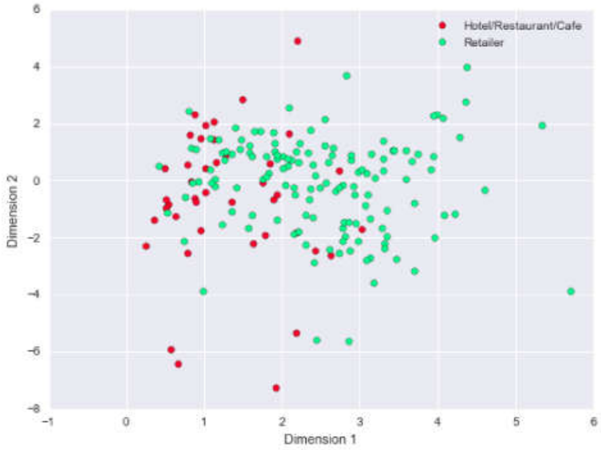
Great job identifying how we can use the cluster labels! The basic idea is that we can perform feature engineering and use the output of an unsupervised learning analysis as an input to a new supervised learning analysis.

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

Learn the best practices for revising and resubmitting your project.

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

RETURN TO PATH

Student FAQ