UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

Overall, this is an excellent first submission, and you only have some minor adjustments to make in order to meet all the specs. You're very close, 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.

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.

For example, Customer 1 looks like it has high spending in relation to the category mean & median for Milk, Grocery, Detergents_Paper. It could be a medium to large-sized 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())
```

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

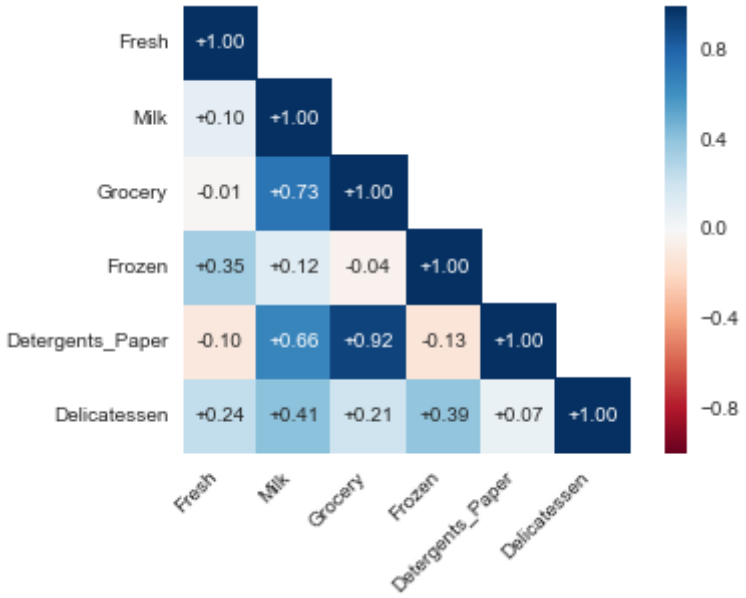
```
Fresh: -0.6662

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.

Excellent 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.

It's also great you point out that your observed correleations — *"confirmed my suspicions about the relevance of the Groceries feature"*. 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
corr = data.corr()
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask, 1)] = True
with sns.axes_style("white"):
    ax = sns.heatmap(corr, mask=mask, square=True, annot=True,
cmap='RdBu', fmt='+.2f')
    plt.xticks(rotation=45, ha='right');
```



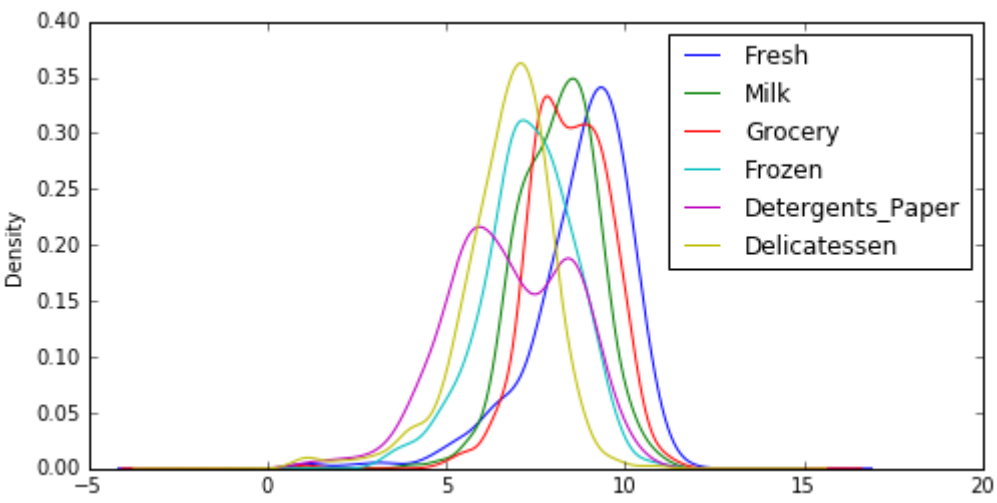
Data Preprocessing

✓

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

Nice job here scaling the data with a very concise code implementation. To get another view of how the log-transformed distributions compare, you can plot them on top of each other:

```
import matplotlib.pyplot as plt
# plot densities of log-transformed data
plt.figure(figsize=(8,4))
for col in data.columns:
    log_data[col].plot.kde()
plt.legend();
```



✓

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

Great stuff here identifying all of the multiple-category outliers! If you wanted, you could also justify removing outlier data points because of the effect they might have on clustering analysis... http://stackoverflow.com/questions/13989419/removing-outliers-from-a-k-mean-cluster

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.

You're generally on point with the discussion, but it could be adjusted slightly to reflect that the dimensions represent patterns of spending that follow the direction of the category weights *as well as* spending that's in the opposite direction of the weights. For example, something like...

> *"The first principal component is made up of large positive weights in Detergents_Paper, and lesser but still sizeable positive weights on Grocery and Milk. It also correlates with a decrease in Fresh and Frozen. This might represent spending in household staples products that are purchased together — some customers buy a lot of household staples while others don't."*

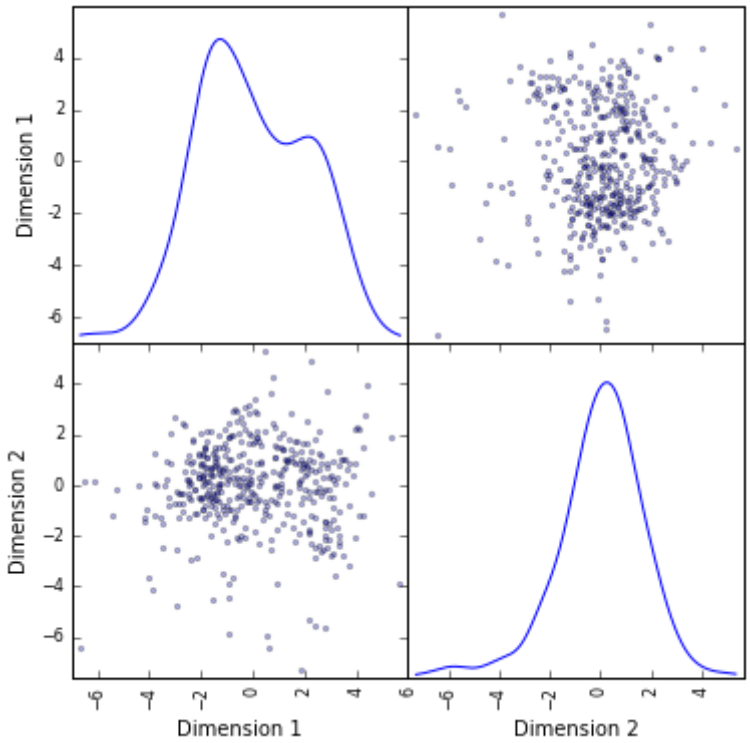
A principal component with feature weights that have **opposite** directions can reveal how customers *buy more* in one category while they *buy less* in the other category (and, vice versa) — for example, a customer that has a large *positive* value for the 3rd component buys a lot of the positive-weighted feature(s) but not much of the negative-weighted feature(s). A customer that has a large *negative* value buys in the opposite proportions.

For a nice visualization of PCA, take a look a this explanation of PCA.

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 look at 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. Remember, the 1st pca component generally correlates with spending in Milk, Grocery, Detergents_Paper.

```
# Produce a scatter matrix for pca reduced data
pd.scatter_matrix(reduced_data, alpha = 0.3, figsize = (6,6), diagonal = 'kd
```



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 scalability (speed) of the two approaches, but it would also be useful to discuss how they differ in soft vs hard assignment of points to clusters...

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 *<< soft or hard? >>* assignment of points to cluster (assumes symmetrical spherical shapes)
- GMM *<< soft or hard? >>* assignment gives more information such as probabilities (assumes elliptical shape)

You can read more on the differences of the methods, and how they are related
(KMeans can be seen as a special case of GMM with equal covariance per cluster).

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.

Great discussion here of the segment center spending 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 average spending in Milk, Grocery, Detergents_Paper.

✓

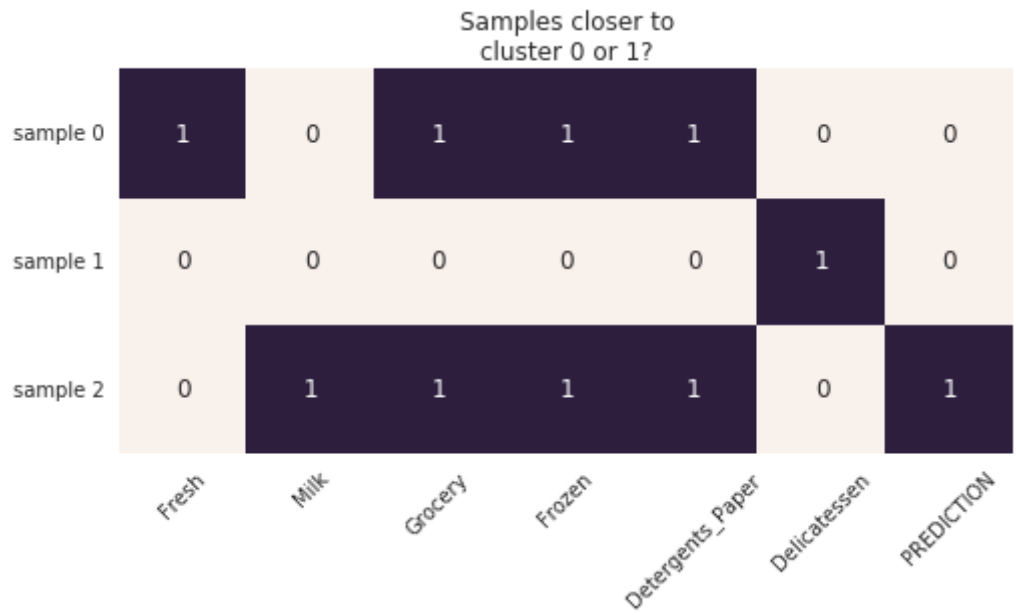
Sample points are correctly identified by customer segment, and the predicted cluster for each sample point is discussed.

Excellent work commenting on the spending of the samples compared to the cluster centers! With the clusters generally split on spending in the first dimension (heavy weights on Milk, Grocery, Detergents_Paper spending), it's interesting to see how the samples have differing combinations of spending in these categories to determine which cluster they're assigned to.

Suggestion:
To examine whether a sample is closer to segment 0 or 1 in a given category, you can also try the below code:

```
import seaborn as sns
import matplotlib.pyplot as plt
# check if samples' spending closer to segment 0 or 1
df_diffs = (np.abs(samples-true_centers.iloc[0]) < np.abs(samples-true_cen
ters.iloc[1])).applymap(lambda x: 0 if x else 1)

# see how cluster predictions align with similariy of spending in each cat
egory
df_preds = pd.concat([df_diffs, pd.Series(sample_preds,
name='PREDICTION')], axis=1)
sns.heatmap(df_preds, annot=True, cbar=False, yticklabels=['sample 0', 'sa
mple 1', 'sample 2'], square=True)
plt.title('Samples closer to\ncluster 0 or 1?')
plt.xticks(rotation=45, ha='center')
plt.yticks(rotation=0);
```

Not all the categories match up with the predicted clusters, but there does seem to be general agreement regarding Milk, Grocery, Detergents_Paper (except perhaps sample 0).



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 here pointing out that we can test the clusters separately — the customers in each cluster might possibly be affected differently by a delivery change, so we're essentially running 2 A/B tests.

```
1. Split a subset of Segment 0 customers into "A/control" and "B/treatmen
t" groups to detect a difference
2. Split a subset of Segment 1 customers into "A/control" and "B/treatmen
t" groups to detect a difference
```

If you're interested, here's some further info on A/B tests in the real world...

- When A/B testing shouldn't be trusted
- Pitfalls of A/B testing

✓

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

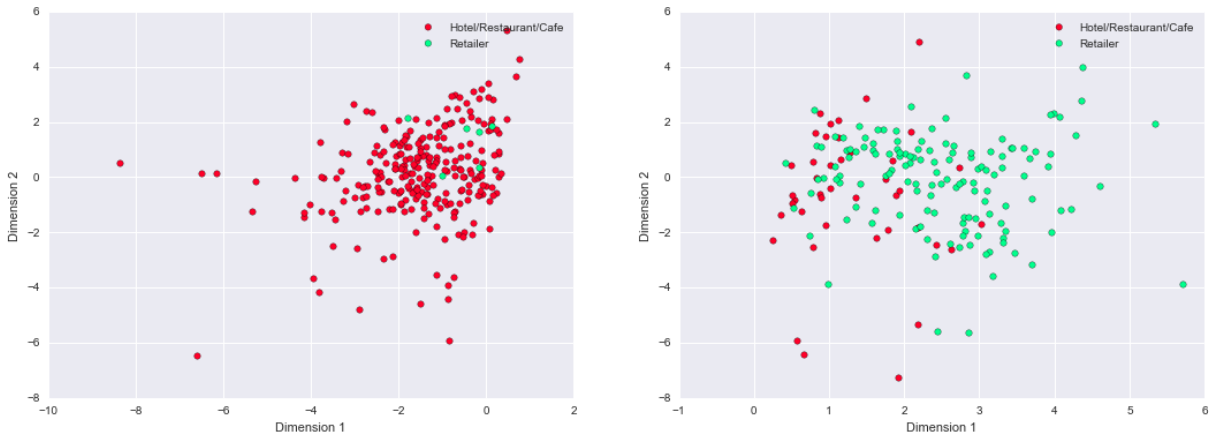
Nice job identifying how we can use the cluster labels! The basic idea is that we can perform feature engineering and use the output of an unsupervised learning analysis as an input to a new supervised learning analysis.

Note: If you download the latest versions of the udacity projects, you can see some updated question language that makes the question intent clearer. The nanodegree project templates get updated periodically, so it's a good idea to check before diving into them.

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.

Great job examining the 'Channel' data in relation to our learned clustering and observing that — *"There is some overlap between the segments, but it's not as critical and happens all the time in the real life"*. Although there's 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 from a 2-cluster analysis are aligned, below you can see the 2 clusters from a K-Means analysis plotted separately (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 and include the link to this review.

RETURN TO PATH

Rate this review

Student FAQ