UDACITY

Back to Machine Learning Engineer Nanodegree

Creating Customer Segments

REVIEW

HISTORY

Meets Specifications

Dear student,

Congratulations on completing the unsupervised section of MLND!
This was an outstanding submission—it doesn't happen every day that someone passes this project on their first attempt

I wish you the best of luck and keep up the hard work!

Data Exploration

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

What a great start!

I like how you made use of the descriptive statistics to guide your inference.

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

You're right!

If we cannot explain most of the feature's variance from other features, it most probably holds a lot of unique information and thus is important for our model—and vice versa!

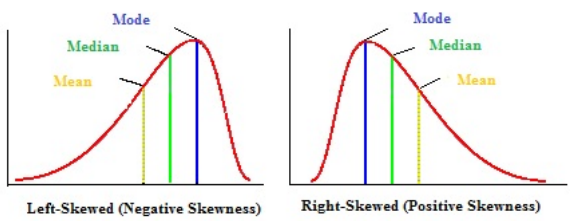
Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

Well done!

`Grocery` and `Detergents_Paper` really are strongly correlated.

You are also right about `Detergents_Paper <-> Milk` and `Grocery <-> Milk`. It might be worth noting that even *scatter plots* for these two pairs form less defined line; telling us that the correlation is relatively mild.

Data is definitely not normally distributed, you're right about that—in fact, it is *positively skewed*.



Data Preprocessing

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

Your code implementation of data scaling is correct, good job!

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

Very good! I like how you're aware that removing all the outliers (altogether 42) would result in a loss of ~10% of our data set which isn't that huge itself.

Deciding whether the outliers should be removed or not is always a difficult task. Some algorithms are better off with the outliers left in the data set, while other might suffer from their presence. Maybe this article on outliers might aid you in making the decision in your future machine learning ventures.

I definitely recommend reading this Quora thread on impact of outliers on clustering. I think it succinctly and comprehensively describes how too many outlying points skew are clustering results.

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.

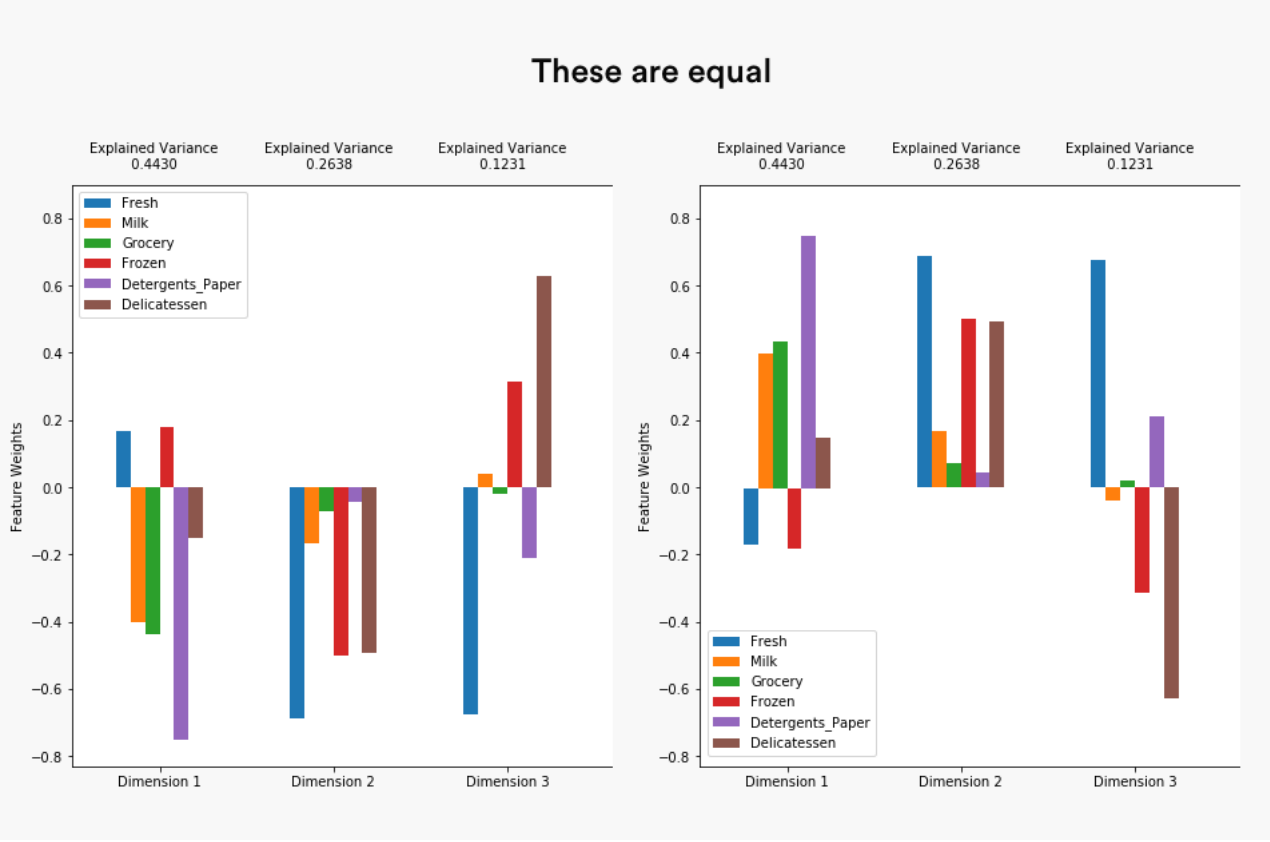
Nice work!

It's always important to analyze what each dimension axis represents in terms of customer behavior and how it separates individual customers. I'm glad you included this analysis in your answer.

While the absolute signs of features don't matter, **the relative signs between the individual features do matter**.

Take *Dimension 3* as an example. You can say this dimension explains customers that spend a lot on `Delicatessen` *and at the same time* have very low spending on `Fresh` products—or vice versa (huge spending on `Fresh`, close to none on `Delicatessen`).

Why is that? Because *the absolute signs of the feature weights don't matter*. During multiple runs of your program, you might get the PCA components same, but with the opposite signs. Note that this is perfectly fine and very common. **Because of that, it doesn't matter if (in Dimension 3) `Fresh` is `-0.7` or `0.7` (it might be both during different runs of the program). What only matters is its relationship with other features.** That means if Fresh is -0.7, Delicatessen will be 0.6—but they could be the opposite: Fresh 0.7 and Delicatessen -0.6.



Quick tip for you: next time you can calculate the cumulative explained variance programmatically:

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

Although you seem to have pretty good understanding of what PCA represents, I also recommend reading this StackExchange thread. It's pretty funny.

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

Nice and clean, well done!

Clustering

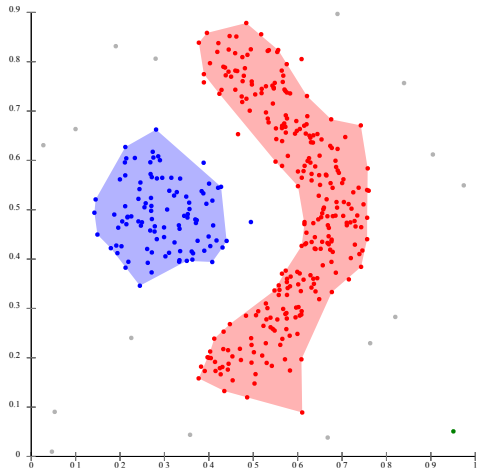
The Gaussian Mixture Model and K-Means algorithms have been compared in detail. Student's choice of algorithm is justified based on the characteristics of the algorithm and data.

Awesome job comparing K-Means and Gaussian Mixture Model! I think you pointed out the most significant differences there.

Since there doesn't seem to be any hard line that could hard-separate our data, your decision to use GMM is perfectly reasonable, I would choose the same.

In your future ventures as a machine learning engineer, you might run into a situation when none of these two algorithms will sufficiently cluster your data. No worries, there are ton of other clustering algorithms you can use.

I would recommend reading up on **DBSCAN**. It can find arbitrarily shaped clusters, is robust to outliers and does not require you to specify the number of clusters in advance. (source: Wikipedia)



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.

You are right, `n_components=2` results in the best Silhouette score of all!

If you're curious, you can also try re-running your project and computing the Silhouette score for model trained on dataset with all outliers in place to see how your data impacts the most optimal number of clusters.

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 job!

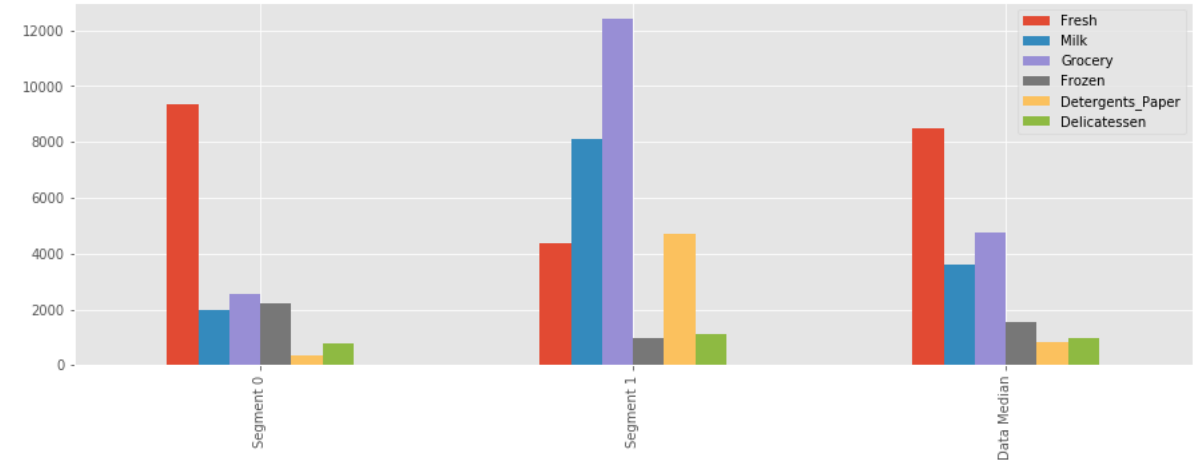
Your application of inverse transformation and scaling on the cluster centers to "recover" the representative customers' spending is flawless. I like your discussion, too.

Similarly as I suggested in the first answer, you can plot the representative establishments next to data median to compare the representations visually. Try running the following code in your project notebook.

```
compare = true_centers.copy()
compare.loc[true_centers.shape[0]] = data.median()

plt.style.use('ggplot')
compare.plot(kind='bar', figsize=(15, 5))
labels = true_centers.index.values.tolist()
labels.append('Data Median')
plt.xticks(range(compare.shape[0]), labels)
plt.show()
```

Again, your plot will look something like this



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

Nice

Conclusion

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

That's right. The key is that **the distributor has to perform the tests on each segment independently.**

In order for A/B testing to be successful, both the treatment group (A) and the control group (B) must be **highly similar** to each other. Otherwise, we can't be sure the results aren't due to some factor other than the one being tested. That's where our new customer segments will help (in identifying group of similar test subjects).

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

Exactly!

Simply put, the new *customer_segment* value you have just discovered using unsupervised methods could be used as a label for a supervised classifier.

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! Really.

I would agree with you that your previous clustering results definitely match the underlying Channel data!
There is a fair amount of overlap, but as you said, the algorithm did a decent job splitting most of the customers.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ