UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

This was a great submission! There are just a couple minor points of revision, but they shouldn't take too long to correct.

Keep up the great work! I look forward to the revision.

Data Exploration

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

Good job here!
Your intuitions are backed up with statistical descriptions of the data 👍

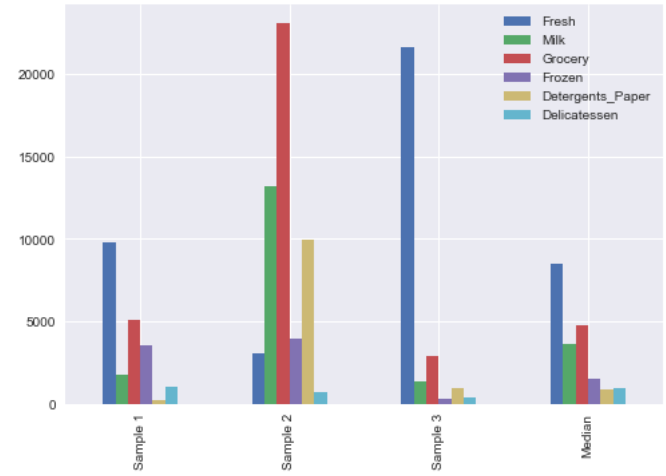
TIP

In general, I find it really helpful to visualize sample points when I'm trying to figure out what they represent. You can do this quite simply with the following code 😄

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

samples_for_plot = samples.copy()
samples_for_plot.loc[3] = data.median()

labels = ['Sample 1','Sample 2','Sample 3','Median']
samples_for_plot.plot(kind='bar')
plt.xticks(range(4),labels)
plt.show()
```

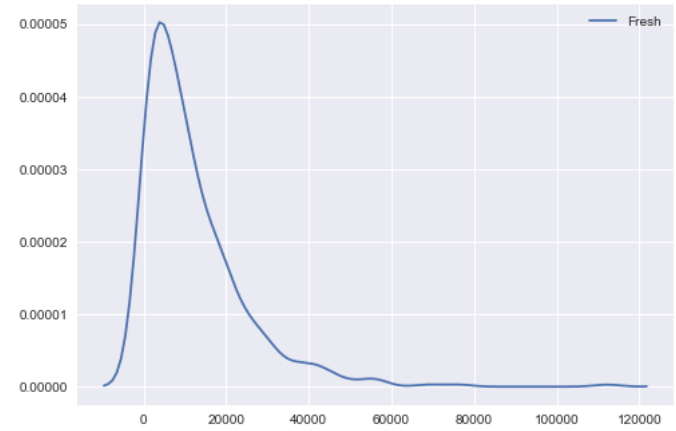


A prediction score for the removed feature is accurately reported. Justification is made for whether the removed feature is relevant.

Great!
You nailed the key point here - if we can reliably reconstruct a feature from other features, it probably doesn't contain a whole lot of unique information. 👍

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

Great discussion overall! Although I think you may have misinterpreted the distribution point. When we are examining the distribution for individual features, make sure you are looking at the plots on the diagonal (KDE plots). The fresh one, for example, looks like this:



What sort of distributions are we looking at for our features here? Normal? Skewed?

Data Preprocessing

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

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

Great code implementation! However, it looks like you might have accidentally deleted the "Question 4" header here. I've posted a picture below:

Question 4

Are there any data points considered outliers for more than one feature based on the definition above? Should these data points be removed from the dataset? If any data points were added to the `outliers` list to be removed, explain why.

In short, simply justify your decision to remove the outliers here. I recommend discussing how our clustering algorithms may be impacted.

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.

Your explanation of the components is decent at a high-level, but we'd like you to go quite a bit deeper into what they represent. Let's take the first component as an example:

If you take a look at the first dimension, as you mentioned you'll find a high negative weight on grocery, milk and detergent with a moderate weight on fresh and frozen. What this really means is that customers who buy lots of detergents, grocery and milk but don't buy much of fresh and frozen goods will have a high positive value for this component (you should think of these components as features). Customers who, on the other hand, buy very little detergent, grocery and milk with relatively more purchases of fresh and frozen will have a high negative value here.

So what you should be doing here is trying to find out what pattern is being represented by each component, and what customer segments we might be able to separate out using them. What sort of establishment would have a high positive vs. high negative value in this component?

Try to answer that question for each of the first 4 components here 😄

RESOURCES

PCA is a pretty difficult topic (at least it was for me). As such, I highly recommend taking a look at the resources below. They do a great job of explaining PCA in a meaningful way.
https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues
http://setosa.io/ev/principal-component-analysis/
https://www.quora.com/What-is-an-intuitive-explanation-for-PCA

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

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.

Great work!

In general, K-Means offers better performance if we care about

- Speed
- Scaleability
- Simplicity

Whereas GMM provides more

- Flexibility
- Robustness

The fact that K-Means assumes that all clusters is globular is a pretty enormous assumption, and is always something we have to take into consideration. GMM is far less rigid in this - it allows these spheres to be stretched and compressed.

There are a ton of other models you can use that weren't discussed in lectures as well. One of my personal favourites is DBScan, which uses a *density* measure rather than a distance measure to determine clusters. This can allow for far more unrestricted cluster shapes, which makes this algorithm quite powerful!

If you're interested, check out the links below for more on this subject 😄
https://algorithmicthoughts.wordpress.com/2013/05/29/machine-learning-dbscan/
https://www.quora.com/What-is-an-intuitive-explanation-of-DBSCAN
http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

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.

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.

Your intuition here is great! However, you should make some statistical arguments to back up your claims. I recommend comparing the feature information to the median of the data or to what quantile they exist in.

TIP

It can be helful to visualize our segments when we're trying to gain an intuition about what they represent. We can do this like so:

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

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

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

Conclusion

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

> Second option - Dividing Segment0 and Segment1 each into two subgroups and performing separate A/B testing on both groups. As both segments are quite different as per my understanding of what they represent, I think delivery service change may affect them independently.

Nice work!

The key here is that we perform a separate A/B test on each segment. This ensures we aren't generalizing our results to customers where they don't apply.

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

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.

Really fantastic work! It's clear that your clusterer did an awesome job on the data.

RESUBMIT

DOWNLOAD PROJECT