‹ Return to "Machine Learning Engineer Nanodegree" in the classroom

Creating Customer Segments

REVIEW | HISTORY

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

Dear student,

This is an excellent first submission! 👏
There are 3 minor things I'd like you to revise, but given your general understanding of the topic, they should be pretty easy for you. I can tell you're on a right path to becoming a great machine learning engineer!

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.

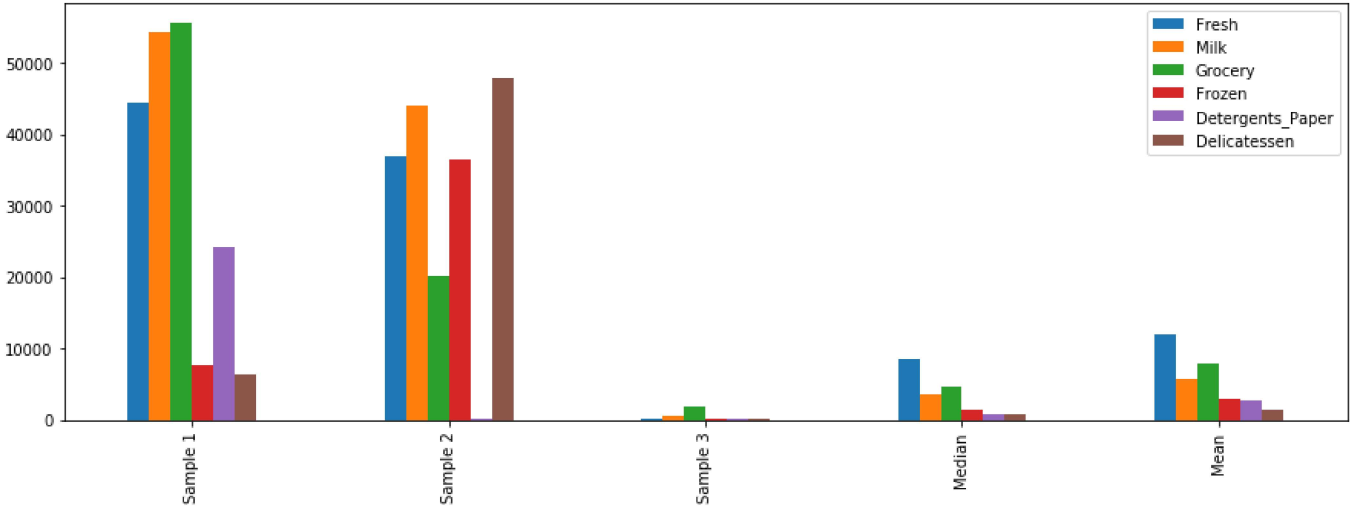
You can also plot your samples' feature values next to average or median sample to compare the establishments visually. Try running the following code in your project notebook.

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

samples_for_plot = samples.copy()
samples_for_plot.loc[3] = data.median()
samples_for_plot.loc[4] = data.mean()

labels = ['Sample 1', 'Sample 2', 'Sample 3', 'Median', 'Mean']
samples_for_plot.plot(kind='bar', figsize=(15, 5))
plt.xticks(range(5), labels)
plt.show()
```

Your plot will look something like this (with different values for samples of your choice)



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.

What about `Grocery` feature? What's the score for that one? Would it be necessary for our clustering algorithm?

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

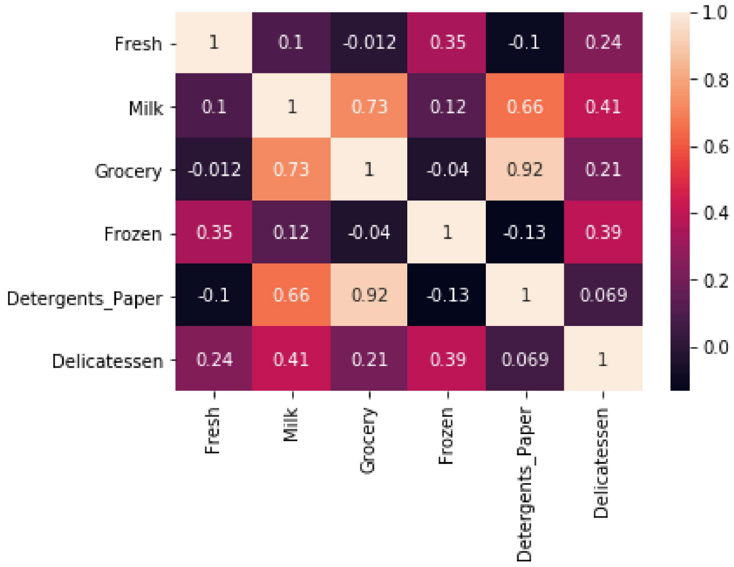
Well done!

`Grocery` and `Detergents_Paper` really are strongly correlated, which (as you mentioned) does confirm your answer in previous section about `Grocery` not bringing a lot of unique information.

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

Awesome use of `data.corr()` to show feature correlation matrix. It can be also beautifully visualised in a heat map.

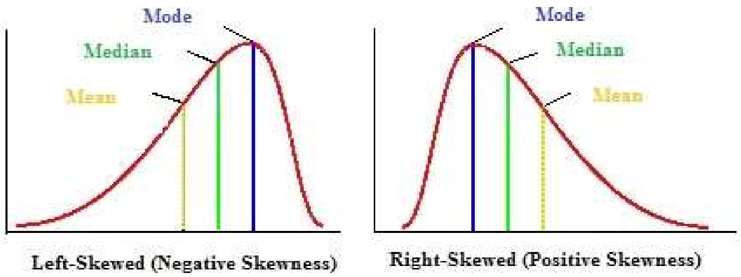
```
import seaborn as sns
sns.heatmap(data.corr(), annot=True);
```



However, I'd still like you to answer the following question:

> How is the data for those features distributed?

If you need, you can also read this article on data distribution.



Left-Skewed (Negative Skewness)

Right-Skewed (Positive Skewness)

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!

Note that you could also simply do:

```
log_data = np.log(data)
log_samples = np.log(samples)
```

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

Good!

However you're mentioning that data points that are outliers in multiple features should be removed, but your `outliers` list is empty, meaning you're not actually removing any outliers in code. Please do one of the following:

- add the double-featured outliers to `outliers` list in code to really remove them; or
- leave the `outliers` list empty but update the answer by discussing why you chose not to remove any outliers

Either way, I'd like you to also justify your decision to (not) remove outliers in a bit more detail. 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

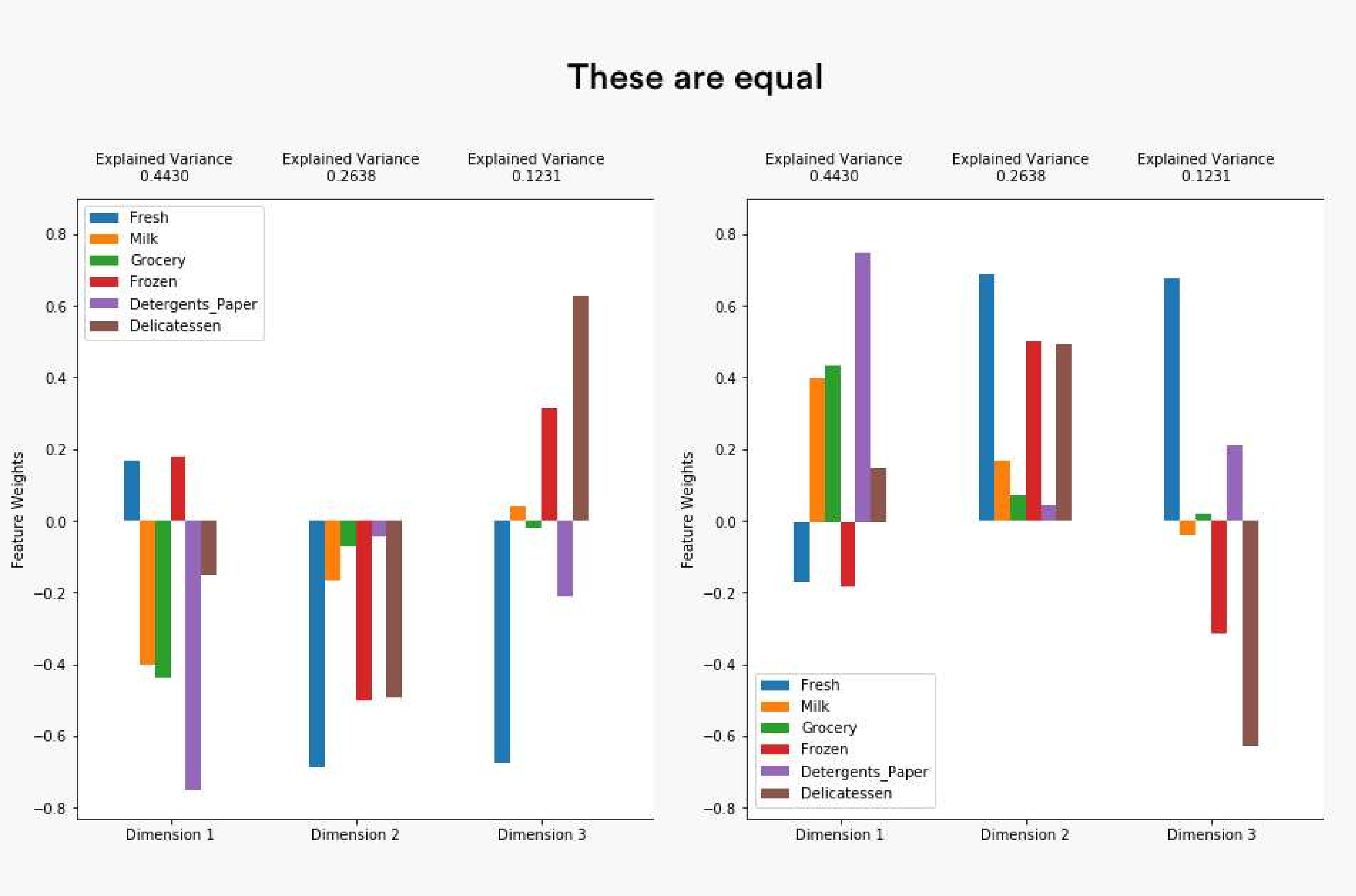
The total variance explained for two and four dimensions of the data from PCA is accurately reported. The first four dimensions are interpreted as a representation of customer spending with justification.

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.

> (Dimension 3): This seems to represent delicatessens that do not also sell fresh food.

Right, or vice versa! 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.



These are equal

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

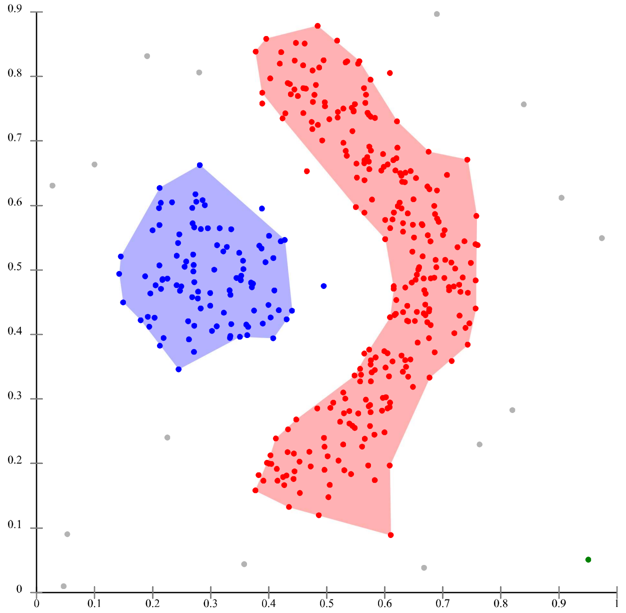
The Gaussian Mixture Model and K-Means algorithms have been compared in detail. Student's choice of algorithm is justified based on the characteristics of the algorithm and data.

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=3` 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 some outliers actually removed 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!

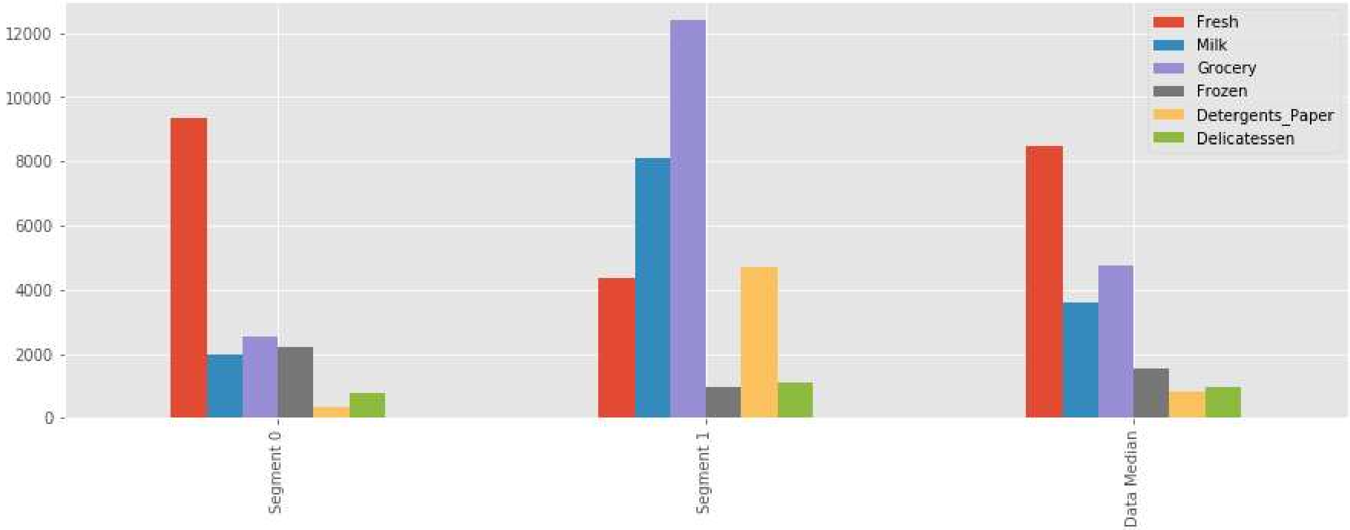
Your application of inverse transformation and scaling on the cluster centers to “recover” the representative customers' spending is flawless. I like your discussion, too.

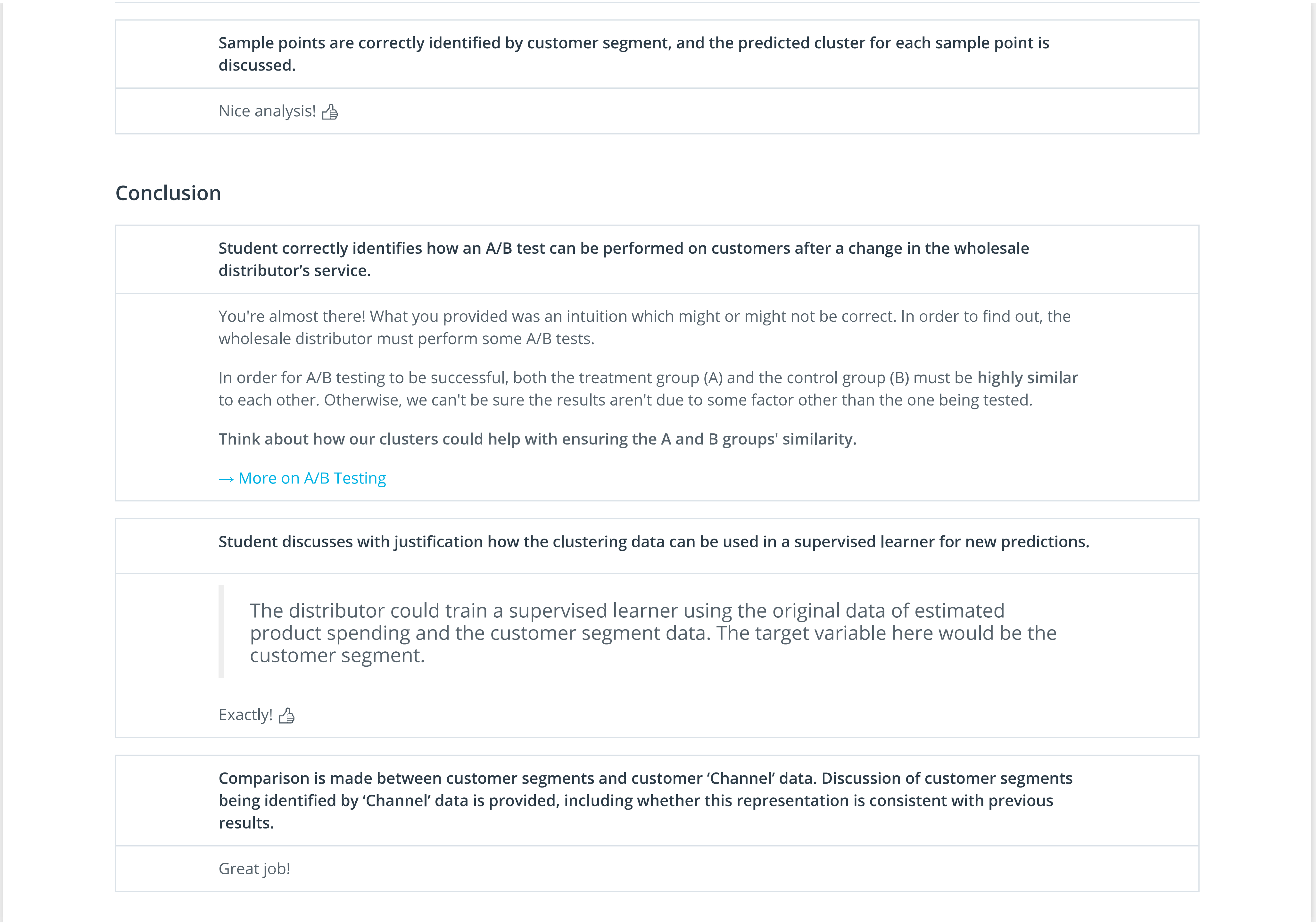
Similarly as I suggested in the first answer, you can plot the representative establishments next to data median to compare the representations visually. Try running the following code in your project notebook.

```
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 analysis! 👍

Conclusion

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

You're almost there! What you provided was an intuition which might or might not be correct. In order to find out, the wholesale distributor must perform some A/B tests.

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.

Think about how our clusters could help with ensuring the A and B groups' similarity.

→ More on A/B Testing

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

> The distributor could train a supervised learner using the original data of estimated product spending and the customer segment data. The target variable here would be the customer segment.

Exactly! 👍

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!

☑ RESUBMIT PROJECT

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)

RETURN TO PATH