UDACITY

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

DISCUSS ON STUDENT HUB

Creating Customer Segments

REVIEW

HISTORY

Meets Specifications

Dear student,

Thanks for updating your answers based on the previous reviewer's feedback and congratulations on completing the unsupervised section of MLND! 🎉

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.

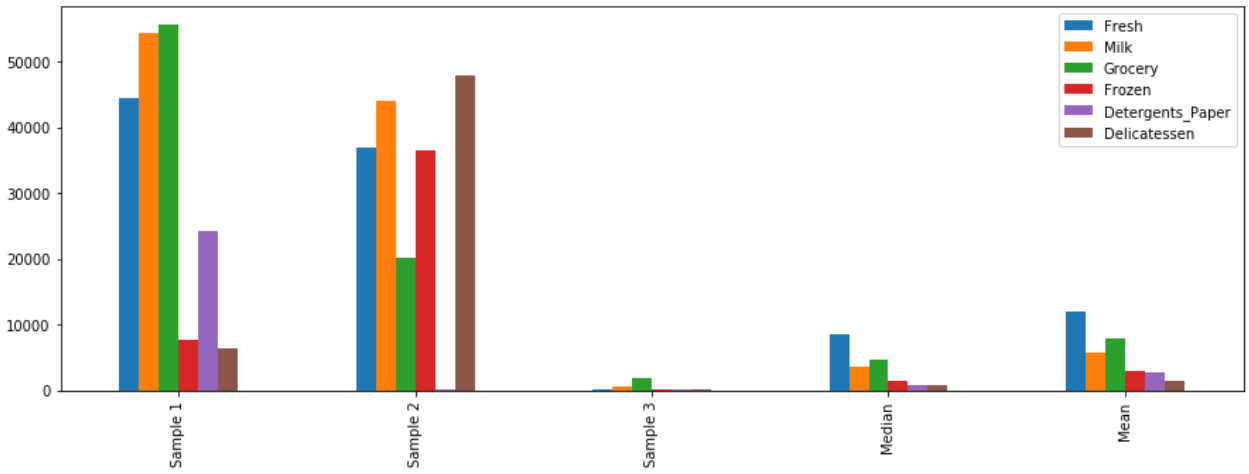
You can also plot your samples' feature values next to average or median sample to compare the establishments visually. Try running the following code in your project notebook.

```
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.

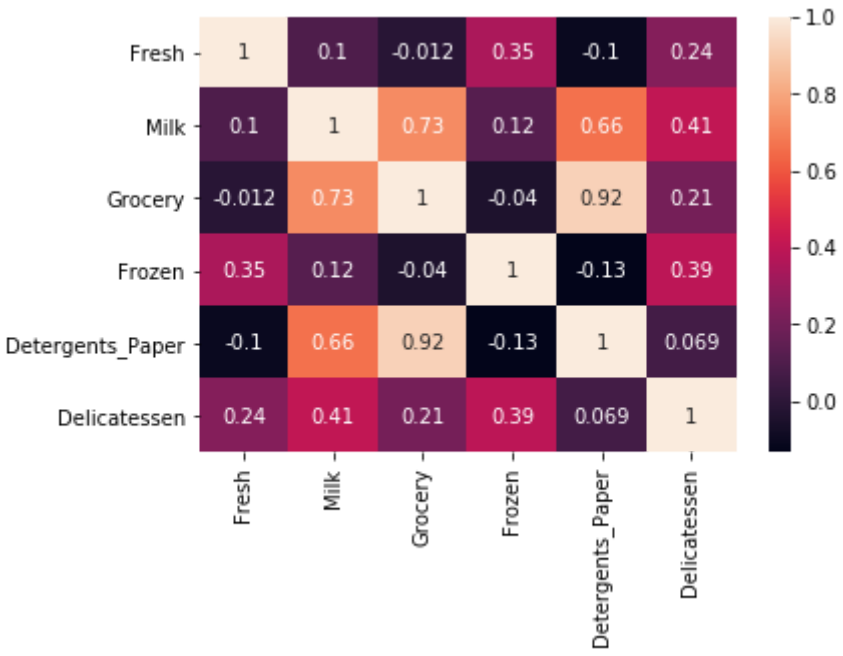
Well done!

The strongest correlation is `Grocery` and `Detergents_Paper`.

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

Feature correlations can also be beautifully visualised using heat map. Unlike scatter plots that rely heavily on the person's subjective representation of the plot, heat maps generate **easily readable scale** on which you can evaluate the correlations.

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



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!

Please note that `np.log()` always returns a new object, you don't have to pass `data.copy()` to it. Although not big deal here, you might waste memory if you're working with bigger data sets.

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

Very good! It's also worth noting 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.

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

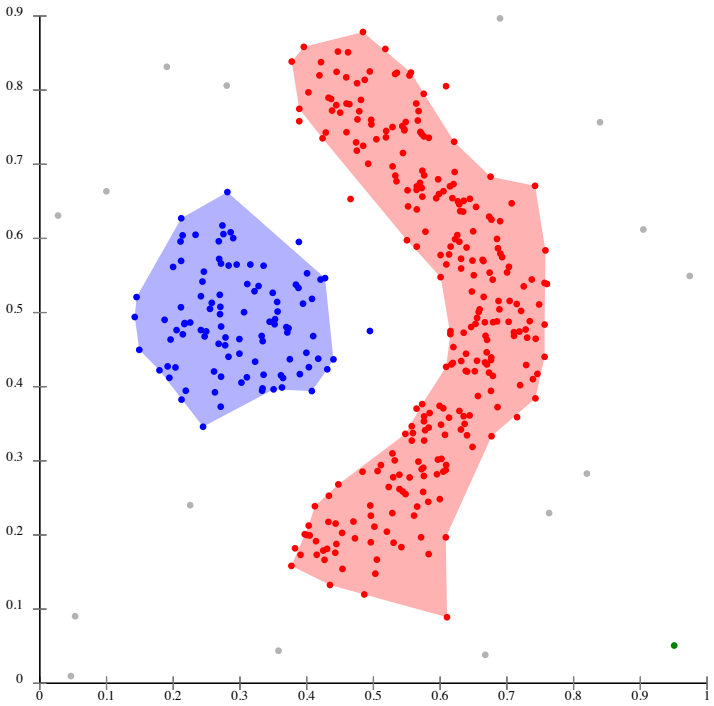
The Gaussian Mixture Model and K-Means algorithms have been compared in detail. Student's choice of algorithm is justified based on the characteristics of the algorithm and data.

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 and choice of algorithm impacts the most optimal number of clusters (or at least the achieved silhouette scores).

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!

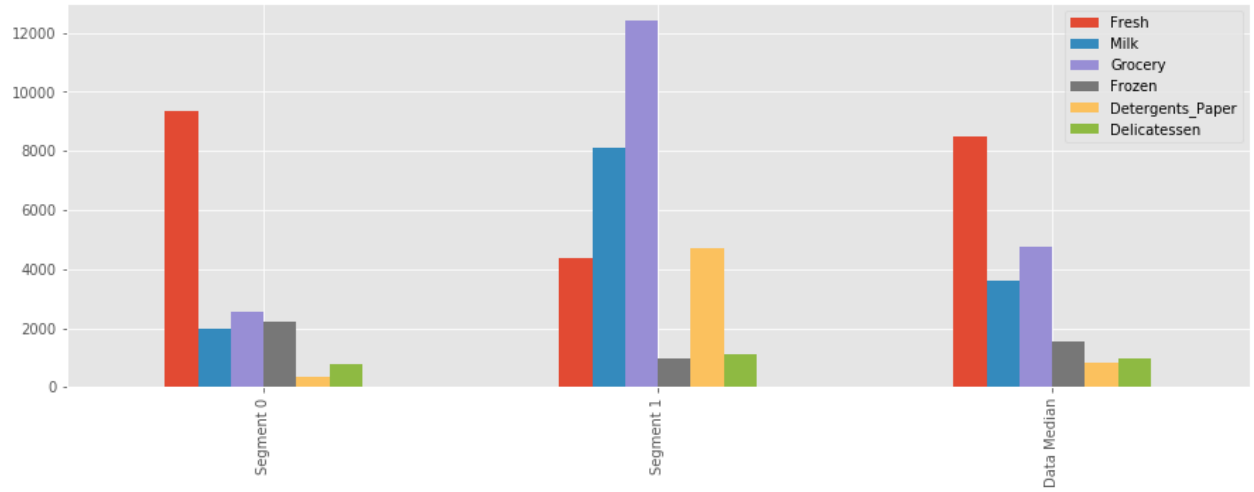
Your application of inverse transformation and scaling on the cluster centers to “recover” the representative customers' spending is flawless. I like your discussion, too.

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.

That's right. The key is that **the distributor has to perform the tests on each segment independently** (or at least evaluate results from the segments separately).

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.

> Using supervised learn we try to predict "customer segment" as our target variable.

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! 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