UDACITY

‹ Back to Machine Learning Engineer Nanodegree

Creating Customer Segments

REVIEW

HISTORY

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

Great work! There's just one coding error, but it shouldn't be too hard to fix.
Keep up the great work - we look forward to the next one!

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 job!

It looks like you found the feature-pairs which are most closely correlated in our dataset.

TIP

It's often worth visualizing our correlations in a *heatmap*, as it can be much easier to comprehend than a scatterplot matrix.

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

Good job addressing the distributions! Often, when we have right-skewed data it approximates a log-normal distribution. It's for this reason that we normalize it in the next section with a log-transformation. However, there are cases where the log-transform doesn't get us as close to normal as we'd like. Another option in that case is the BoxCox transformation.
We can apply it here like so

```
from scipy.stats import boxcox
boxcox_data = data.apply(lambda x: boxcox(x)[0])
```

```
pd.scatter_matrix(boxcox_data, alpha = 0.3, figsize = (14,10), diagonal = 'kd
e');
```

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.

You've got an error in your code here:

```
for feature in log_data.keys():

    # Calculate Q1 (25th percentile of the data) for the given feature
    Q1 = np.percentile(log_data, 25)

    # Calculate Q3 (75th percentile of the data) for the given feature
    Q3 = np.percentile(log_data, 75)
```

You should be calculating the outliers separately for each feature. Your code should look something like this:

```
for feature in log_data.keys():

    # Calculate Q1 (25th percentile of the data) for the given feature
    Q1 = np.percentile(log_data[<<INSERT VARIABLE>>], 25)

    # Calculate Q3 (75th percentile of the data) for the given feature
    Q3 = np.percentile(log_data[<<INSERT VARIABLE>>], 75)
```

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.

Really fantastic work!

Good job finding the cumulative explained variance. As a tip, we can do this programmatically like so

```
print np.cumsum(pca_results['Explained Variance'])
```

Really good description of the PCA components. It looks like you identified what they mean in terms of correlation as well as how they might represent the degree to which a customer is like a certain kind of customer segment.
It seems like you have a solid handle on PCA, but if you're interested in reading more I encourage you to check out the post below. It really helped me wrap my head around it when I first started 😄
https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

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.

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.

Good work!
It looks like you've made some solid statistical arguments to support your choices 👍

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.

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.

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH

Rate this review

Student FAQ