UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

You raised some good points with respect to the previous review, and it was fun to think and discuss about them! Hopefully, you find my arguments relevant 😄

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 work predicting the establishments represented by the sample points based on the comparison of their features to the dataset quartiles.

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

Your interpretation of the relevance of `Delicatessen`, based on the average prediction score obtained, is absolutely correct! The low/negative prediction score for a feature means that the values of that feature cannot be predicted well by the other features in the dataset and therefore, the feature is not redundant and may contain useful information not contained in other features.

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

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.

Good work, correctly identifying the Tukey outliers for more than one features!

Code tip:

You can use `Counter` for this task in the following way:

```
from collections import Counter
outliers_index = []
for feature in log_data.keys():
    ....
    # Display the outliers
    print "Data points considered outliers for the feature '{}':".format(feature)
    outliers_temp = log_data[~((log_data[feature] >= Q1 - step) & (log_data[feature] <= Q3 + step))]
    display(outliers_temp)
    outliers_index.extend(list(outliers_temp.index.values))
```

```
# OPTIONAL: Select the indices for data points you wish to remove
outliers = [item for item, count in Counter(outliers_index).iteritems() if count > 1]
```

Required:

You need to give a reasonable justification for not removing any outlier. In particular, how could the outliers influence the clustering analysis, in general, and why do you think that this would not be the case in the context of this project?

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.

" I indeed interpreted only the absolute values and not the sign of the numbers"

It is important to remark that `Fresh` and `Delicatessen` have opposing signs in Dimension 3. Such a dimension would characterise very different set of customers compared to a dimension in which `Fresh` and `Delicatessen` have large absolute weights of the same sign.
It doesn't matter, as you seem to be well aware, whether `Fresh` has a positive sign and `Delicatessen` negative, or vice versa, but it matters that they have opposing signs.

It is also important to provide examples of the customers which would have spending pattern characteristic of each dimension. For instance, supermarkets would exhibit a spending pattern similar to the first PCA dimension.

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.

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 absolutely right that Silhouette score should not be the only criterion to decide the optimal number of clusters. But you do need some complementary criterion if you want to choose a number giving sub-optimal score.
For example, in the link that you provided, 2 is not considered optimal, despite having a better Silhouette score, because it doesn't result in ***balanced*** clusters, while 4 does.

To conclude, you can certainly choose a number giving sub-optimal Silhouette score, but you must justify this choice using some alternate, and objective, criterion, not by saying that a higher number would lead to a more fine-grained clustering.

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 analysis!

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

You need to explicitly compare the sample points to the `cluster_centers` to conclude whether the `predictions` from the algorithm agree with your intuition or not.

Conclusion

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

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

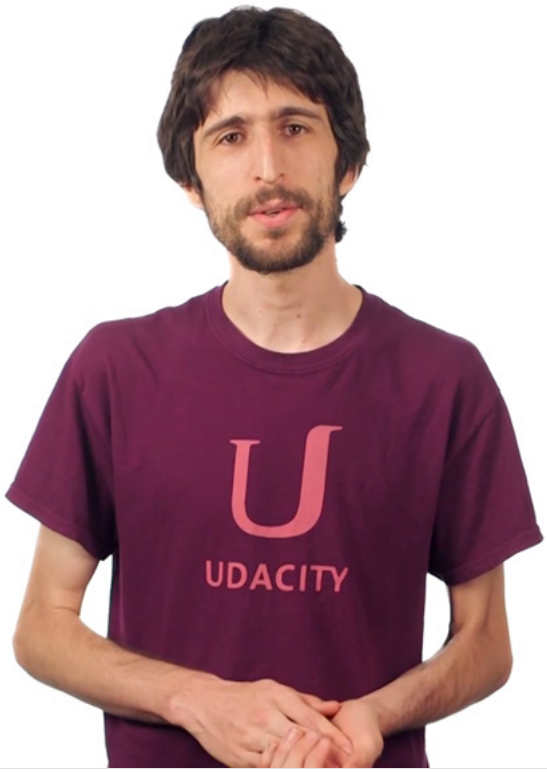
Comparison is made between customer segments and customer 'Channel' data. Discussion of customer segments being identified by 'Channel' data is provided, including whether this representation is consistent with previous results.

KMeans seems to have done a pretty decent job here! Even when the data is not linearly separable, as is generally the case in real world, KMeans can still give surprisingly good results, and is therefore, a good first algorithm to use for a lot of clustering problems.

GMM could also have been a good choice here as the scalability is not an issue and the clusters do have a fair amount of overlap in reality. Although a perfect classification is not possible to achieve even with GMM, soft clustering gives us confidence levels in our predictions, which would understandably be low at the boundary between two clusters.

RESUBMIT

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

Rate this review

Student FAQ