UDACITY

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

DISCUSS ON STUDENT HUB

Creating Customer Segments

REVIEW

CODE REVIEW

HISTORY

Requires Changes

7 SPECIFICATIONS REQUIRE CHANGES

Dear student,

Even though you need to revise answers to a few questions, this is an excellent first attempt!
I hope that the hints and the reading material given in this review will help you meet all the specifications in your next submission.

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.

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

Suggestion:

As we see later, the features' distribution is highly *right-skewed*, therefore, the median would probably serve as a better reference than mean. In fact, I would recommend comparing to the quartiles to get a better idea of the nature of the establishments represented.

Code tip:

You can use the following code to plot the percentile heatmap for sample points:

```
import seaborn as sns

percentiles_data = 100*data.rank(pct=True)
percentiles_samples = percentiles_data.iloc[indices]
sns.heatmap(percentiles_samples, annot=True)
```

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

Required:

You need to revisit the interpretation of the relevance of `Delicatessen` for identifying customers' spending habits, based on the R^2-score obtained.
Hint: Think of this question in terms of *feature selection*. Which features would you like to keep in your dataset: those which cannot be predicted by other features, and hence, provide unique information, or the inverse?

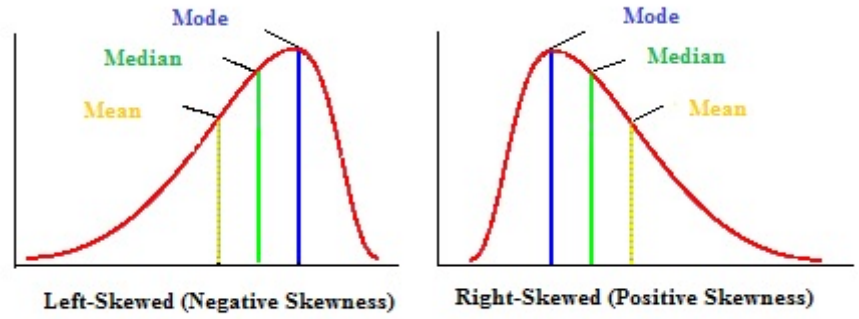
Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

The most significant correlation is definitely between `Grocery` and `Detergents_Paper`. `Milk` is also correlated with both these features, but the correlation is relatively mild.

Required:

- You must re-interpret the relevance of `Delicatessen` in view of these correlations after doing the same in the previous question, and say a quick word about how the observations in the two questions align.
- Although your observation that the data is *not normally distributed* is correct, you seem to be looking only at the off-diagonal joint scatter plots. It can be misleading to infer the marginal distribution from a joint distribution plot, therefore, you need to look at the *marginal distribution* plots along the diagonal of the `scatter_matrix` for describing the features' distribution.
  This would help you provide more details on the features' distribution, such as whether the distribution is

skewed, and in what direction. You can also refer to the following illustration for comparison:



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.

Code issue:

Your code,

```
Q1 = np.percentile(log_data, 25)
Q3 = np.percentile(log_data, 75)
```

calculates percentiles with respect to the all the features, while to find the outliers for a specific feature, we would need these percentiles for that particular feature. For example, to find the outliers for 'Grocery', the corresponding code should look like:

```
# This calculates the percentiles for just `Grocery`
Q1 = np.percentile(log_data['Grocery'], 25)
Q3 = np.percentile(log_data['Grocery'], 75)
```

How would this piece of code look within a `for` loop?

Also, could you please be a little more specific in your justification for outlier removal? In particular, what do you expect to gain from outlier removal in the context of clustering? And why not remove all the Tukey outliers, instead of a smaller subset?

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.

Remarks:

Nice work elaborating on the PCA dimensions and interpreting them as a representation of customer spending! Two main takeaways:

- A high/low (absolute) value along the PCA dimension can help differentiate between different types of customers. For example, a dimension giving relatively high (positive or negative) weights to `Fresh`, `Milk`, `Frozen` and `Delicatessen` would likely separate out the restaurants from the other types of customers.
- A corollary of the above remark is that the sign of a PCA dimension itself is not important, only the relative signs of features forming the PCA dimension are important. In fact, on running the PCA code again, one might get the PCA dimensions with the signs inversed. For an intuition about this, it is helpful to think about a vector and its negative in 3-D space - both are essentially representing the same direction in space. You might find this exchange informative in this context.

The following links might be of interest in the context of this question:
https://onlinecourses.science.psu.edu/stat505/node/54
http://setosa.io/ev/principal-component-analysis/

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.

Required:

> Since PCA resulted in the first components explaining most of the variance in the data set, a K-means approach using our final two components seems to be a good approach.

The number of significant PCA dimensions has no link with the number of expected clusters in a dataset. And even if there was a link, it won't be an advantage for KMeans, since you have to specify and tune the number of

clusters for GMM as well.

I provide below some citations which might help you make a more informed decision here:
http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/mixture.html
http://www.nickgillian.com/wiki/pmwiki.php/GRT/GMMClassifier
http://playwidtech.blogspot.hk/2013/02/k-means-clustering-advantages-and.html
http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/K-Means_Clustering_Overview.htm
http://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means
http://www.r-bloggers.com/k-means-clustering-is-not-a-free-lunch/
http://www.r-bloggers.com/pca-and-k-means-clustering-of-delta-aircraft/
https://shapeofdata.wordpress.com/2013/07/30/k-means/
http://mlg.eng.cam.ac.uk/tutorials/06/cb.pdf

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.

Required:

As specified in the statement of Q8, please justify your prediction of the establishments represented by the two clusters by comparing *explicitly* to the statistical measures of the dataset.

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

Required:

Q9 needs a little more analysis. In particular, you should explicitly compare the features of the sample points to those of 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.

Required:

You are on the right track with your intuition that the proposed change will impact the customers in different segments differently, but could you also briefly describe the implementation of A/B testing to verify your hypothesis?
In particular, please state precisely how many A/B tests that you would need to run, and identify the experimental and control groups for each test.

To recall, the principle behind A/B testing can be stated as follows:

A/B testing is an experiment performed on small samples from the population, just large enough to get statistically significant results. In A/B testing, everything besides the testing parameter should remain as similar as possible for both the experiment (A) and the control (B) groups, so that we can study the change in behavior caused by the testing parameter.

Following links might be of interest here. In particular, the last link discusses A/B testing in the context of clustering:
https://www.quora.com/When-should-A-B-testing-not-be-trusted-to-make-decisions/answer/Edwin-Chen-1
http://techblog.netflix.com/2016/04/its-all-about-testing-netflix.html
https://vwo.com/ab-testing/
http://stats.stackexchange.com/questions/192752/clustering-and-a-b-testing

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.

Indeed, KMeans does 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.

Code tip:

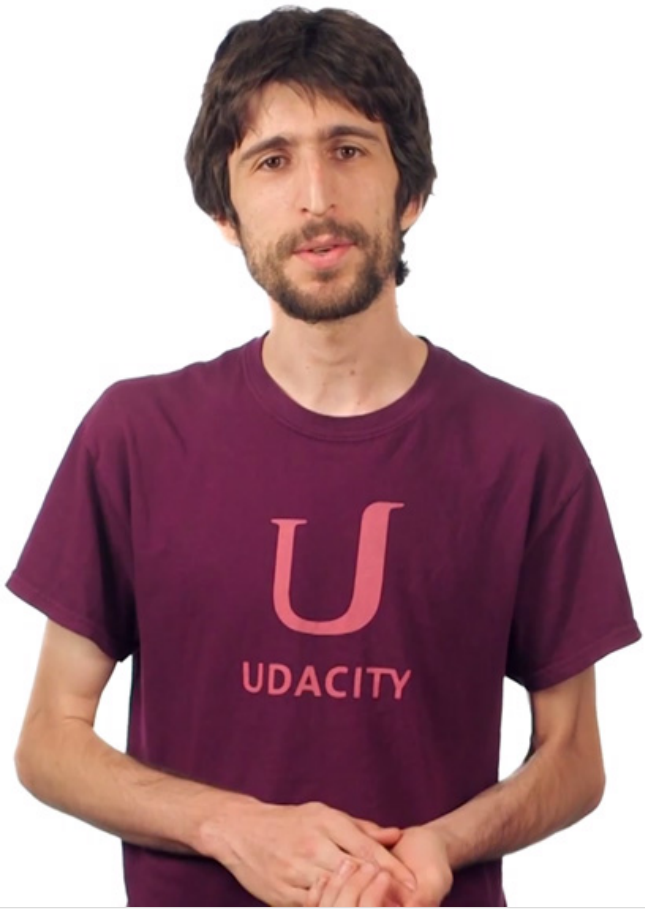
You can calculate the accuracy score for clustering using the following code:

```
channel_labels = pd.read_csv("customers.csv")["Channel"]
channel_labels = channel_labels.drop(channel_labels.index[outliers]).reset_in
dex(drop = True) - 1
# channel_labels = abs(channel_labels -1)
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(channel_labels,preds)
```

Note that I've subtracted 1 from `channel_labels`, because the given `channel_labels` are `1` and `2`, while our cluster-labels are `0` and `1`.
Also, note that the assignment of labels - `0` and `1` - in the clustering algorithm is completely arbitrary. Therefore, you might have to keep or remove `channel_labels = abs(channel_labels -1)` in the above code, to ensure that the cluster and channel labels are "compatible".

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