UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

This is a very solid analysis here and very impressed with the thoroughness of your answers. The big issue here is with your `pca_samples`, but once this is fixed everything else should fall into place. You have an excellent grasp on these unsupervised learning techniques. You just need to prefect a few more sections here and you will be good to go. Keep up the great 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 ideas for potential establishments, however for this section please also compare the purchasing behavior of each sample to the descriptive stats of the dataset. As stating "*higher end which sell in large numbers*" wouldn't necessarily give a good representation of how this customer compares to the entire dataset as a whole. Thus a good idea here would be to compare each product to the mean / median / quartiles.

This may help

```
display(samples - np.round(data.mean()))
display(samples - np.round(data.median()))
```

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

Love the for loop and great with your comment of "*As Fresh is the most difficult feature to predict, it has to be relevant to customer spending habits.*" Thus if we have a high r^2 score(high correlation with other features), this would not be good for identifying customers' spending habits(since the customer would purchase other products along with the one we are prediction). Therefore a negative / low r^2 value would represent the opposite as we could identify the customers specific behavior just from the one feature.

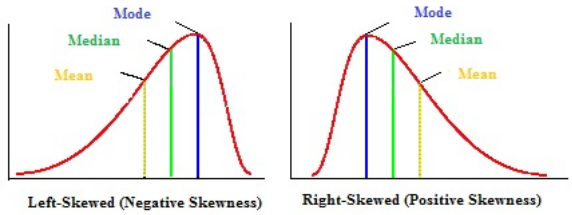
Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

Great job capturing the correlation between features. We could actually get some more insight by looking at numerical correlation by adding it to the plot with

```
axes = pd.scatter_matrix(data, alpha = 0.3, figsize = (14,8), diagonal = 'kde')
corr = data.corr().as_matrix()
for i, j in zip(*np.triu_indices_from(axes, k=1)):
    axes[i, j].annotate("%.3f" %corr[i,j], (0.8, 0.8), xycoords='axes fraction', ha='center', va='center')
```

However relook at your comments such as "*All the feature data is left skewed*", as I believe that you have the skewness switched. The side of the skew refers to the direction of the tail(where the outliers are). Check out this link and visual

(http://www.everythingmaths.co.za/maths/grade-11/11-statistics/11-statistics-05.cnxmlplus)



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.

Nice work discovering the indices of the five data points which are outliers for more than one feature of `[65, 66, 75, 128, 154]`. Would recommend also listing these out.

And great analysis for the need to remove these data points. As these can greatly influence future algorithms, such as a distance based clustering algorithm or PCA. Awesome!

(http://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/)
(http://graphpad.com/guides/prism/6/statistics/index.htm?stat_checklist_identifying_outliers.htm)

Anything particular about these data points

```
data.ix[outliers]
```

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 with the cumulative explained variance for two and four dimensions.

- As with two dimension we can easily visualize the data(as we do later)
- And with four components we retain much more information(great for new features)

And good simple justification for these PCA components. To go even further here with the interpretation of the PCA components:

- In terms of customers, since PCA deals with the variance of the data and the correlation between features, the first component would represent that we have some customers who purchase a lot of Milk, Grocery and Detergents_Paper products while other customers purchase very few amounts of Milk, Grocery and Detergents_Paper, hence spread in the data.

Pro Tip: You can also visualize the percent of variance explained to get a very clear understanding of the drop off between dimension. Here is a some starter code, as np.cumsum acts like `+=` in python.

```
import matplotlib.pyplot as plt
x = np.arange(1, 7)
plt.plot(x, np.cumsum(pca.explained_variance_ratio_), '-o')
```

PCA has been properly implemented and applied to both the scaled data and scaled sample data for the two-dimensional case in code.

You have a code issue here that is giving you incorrect results later on. In your

```
# TODO: Transform the sample log-data using the PCA fit above
pca_samples = pca.transform(log_data)
```

You should be transforming the `log_samples` and not the `log_data`

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.

Good choice in GMM, as we can actually measure the level of uncertainty of our predictions. I would choose the same! As the main two differences in these two algorithms are the speed and structural information of each:

Speed:

- K-Mean much faster and much more scalable
- GMM slower since it has to incorporate information about the distributions of the data, thus it has to deal with the co-variance, mean, variance, and prior probabilities of the data, and also has to assign probabilities to belonging to each clusters.

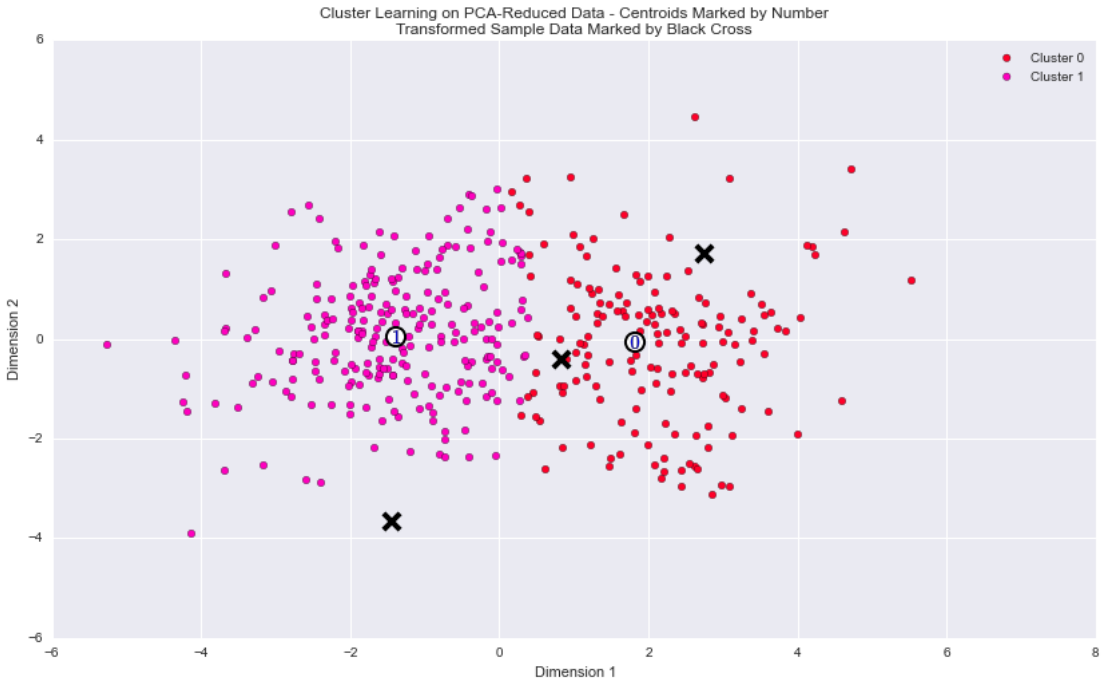
Structure:

- K-Means straight boundaries (hard clustering)
- GMM you get much more structural information, thus you can measure how wide each cluster is, since it works on probabilities (soft clustering)

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.

Love the for loop! Could also try removing only outliers for two+ features and run your code, as you might observe how the distance between 2 and 3 clusters grows.

The reason this is marked as *Requires Changes* is that your plot here should only have three `x`, not 400. Therefore when you fix your code issue above this will be fixed. It should look like this(maybe different colors)



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 justification for your cluster centroid by comparison of cluster centers with descriptive stats of the dataset. Great work!! You could also examine the reduce PCA plot. Anything interesting about dimension 1 and how the clusters are split?

Pro Tip: We can also add the median values from the data and very easily visualize the cluster centroids with a pandas bar plot

```
true_centers = true_centers.append(data.describe().ix['50%'])
true_centers.plot(kind = 'bar', figsize = (16, 4))
```

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

Great justification for your predictions by comparing the purchasing behavior of the sample to the purchasing behavior of the cluster centroid!

Another cool thing we can do, since you are using GMM, it check out the probabilities for belonging to each cluster

```
for i,j in enumerate(pca_samples):
    print "Probability of Sample {}: {}".format(i,clusterer.predict_proba([j])[0])
```

Conclusion

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

"*Considering the data is generally divided into two segments, a small test can be run on a small subset of customers from both segments. If the customers in these small subsets are satisfied with the delivery they will stay with the same wholesale, or even increase the returns from the same customer.* "

Spot on! We should run separate A/B tests for each cluster independently. As if we were to use all of our customers we would essentially have multiple variables(different delivery methods and different purchasing behaviors).

The wholesale distributor can look at the p values for the tests that the null hypothesis that the difference between the chosen metric between the control group and the experiment is zero. If the p value for segment 0 A/B test is smaller, it means segment 0 customers are affected more by the change.

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

Nice idea to use the cluster assignment as new labels. Another cool idea would be to use a subset of the newly engineered PCA components as new features(great for curing the curse of dimensionality). PCA is really cool and seem almost like magic at time. Just wait till you work with hundreds of features and you can reduce them down into just a handful. This technique becomes very handy especially with images. There is actually a handwritten digits dataset, using the "famous MNIST data" where you do just this and can get around a 98% classification accuracy after doing so. This is a kaggle competition and if you want to learn more check it out here KAGGLE

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.

You have good analysis here, this will pass when your visual is fixed and every data point is not a sample data point.

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)

Have a question about your review? Email us at review-support@udacity.com and include the link to this review.

RETURN TO PATH

Student FAQ