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

This is a very solid analysis here and impressed with your answers. You have an excellent grasp on these unsupervised learning techniques. You just need to fine tune a few of these sections and you will be good to go, but should be simple fixes and great for learning the material even better. 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 "*Large quantity of fresh items, milk, and grocery sold*" 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()))
```

Using descriptive stats is a bit part of unsupervised learning and understanding distributions.

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

> "The features with low R^2 scores ('Delicatessen' and Milk in this sentivity study) cannot be predicted well from the remaining features, and are necessary for identifying customer's spending habits. Detergents_Paper and Grocery features have medium-high R^2 values and could be considered for removal if time and cost of collecting such data and analysis were constraints. "

These comments are spot on! 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 predicting, as we could actually derive this feature from the rest of the features). Therefore a negative / low r^2 value would represent the opposite as we could identify the customer's specific behavior just from the one feature.

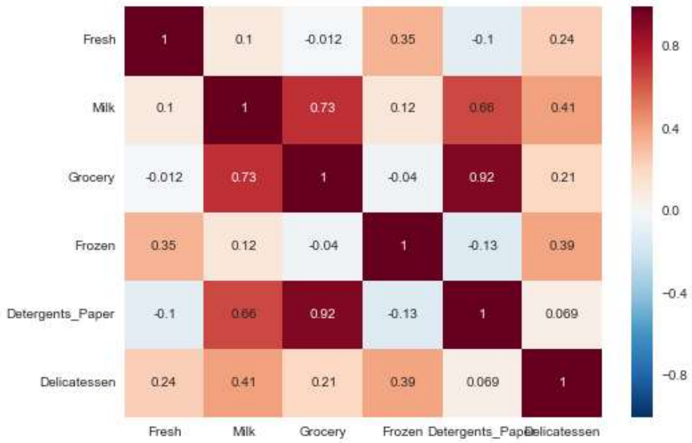
Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

SUGGESTION

> " Milk and Delicatessen indicate correlation that is good at low quantities, but becomes weaker at higher quantities."

I would be very hesitant to say Milk and Delicatessen have correlation, as Milk and Detergents_Paper should be more relevant.

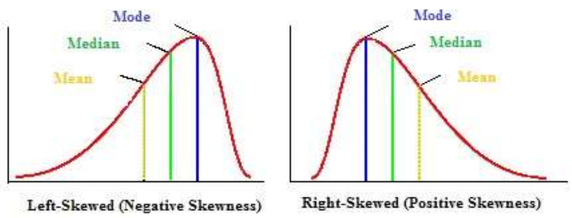
```
import seaborn as sns
sns.heatmap(data.corr(), annot = 'True')
```



AWESOME

> "The data is not normally distributed for any of the features, and are right-skewed (data points are close to the left axis), as seen in the graphs above."

Skewed right is correct. Could also mention log normal. And correct that we can actually get an idea of this from the basic stats of the dataset, since the mean is above the median for all features. We typically see this type of distribution when working with sales or income data.


Mode
Median
Mean
Left-Skewed (Negative Skewness)
Mode
Median
Mean
Right-Skewed (Positive Skewness)

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.

Great job discovering the indices of the five data points which are outliers for more than one feature of `[65, 66, 75, 128, 154]`.

As outlier removal is a tender subject, as we definitely don't want to remove too many with this small dataset. But we definitely need to remove some, since outliers can greatly affect distributions, influence a distance based algorithm like clustering and/or PCA! One cool thing about unsupervised learning is that we could actually run our future analysis with these data points removed and with these data points included and see how the results change.

(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)

Maybe also examine these duplicate data points further with a heatmap in the original data.

```
# Heatmap using percentiles to display outlier data
import matplotlib.pyplot as plt
import seaborn as sns
percentiles = data.rank(pct=True)
percentiles = percentiles.iloc[outliers_frequent]
plt.title('Multiple Outliers Heatmap', fontsize=14)
```

```
heat = sns.heatmap(percentiles, annot=True)
display(heat)
```

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. Could look into using `np.cumsum(pca.explained_variance_ratio_)`

- 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)

REQUIRED

> "Dimension 3: Delis like The Sandwich Spot SF or Miller's East Coast Deli (higher positive weight on Delicatessen).
> Dimension 4: Frozen meat/seafood stores(some online) like Stop and Shop (higher positive weights on Frozen items)."

In PCA 3 and PCA 4 just make sure you also mention the large feature weights in the opposite direction as well. As the most prevalent features in each component refers to the highest absolute magnitude. Remember here that PCA deals with the variance(spread) of the data and the correlation between features. And a principal component with feature weights that have opposite directions can reveal how customers buy more in one category while they buy less in the other category.

Can check out these links

- https://onlinecourses.science.psu.edu/stat505/node/54
- http://webspace.ship.edu/pgmarr/Geo441/Lectures/Lec%2017%20-%20Principal%20Component%20Analysis.pdf

SUGGESTION

> "Dimension 1: Convenience stores like 7-Eleven or Gas Station Mini-Market (negative weights for Fresh and Frozen items)."

Should also actually mention that PCA refers to the spending of Deter, Milk, Grocery.

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.

Love the thorough analysis here, great job! 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.

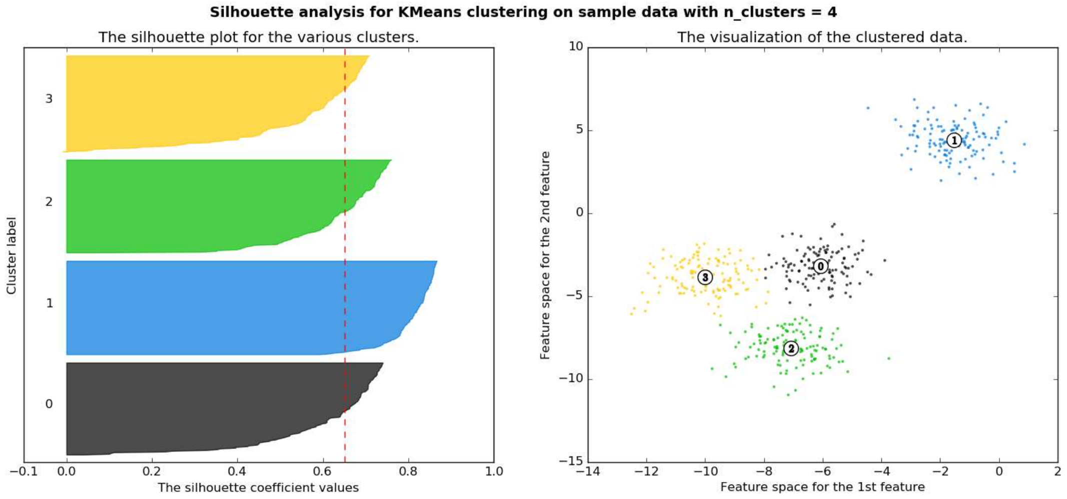
Structure:

- 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.

Good work with the for loop and glad that you played around with both! As we can clearly see that K = 2 gives the highest silhouette score. Another cool interpretation method for Silhouette score is like this

(http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html)



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 idea for potential establishments, however also for this section, you should explicitly reference the descriptive stats of the features for justification for each cluster centroid. Thus a good idea here would be to compare each product to the mean / median / quartiles.

This may help, as 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!

Could also check out the distance to each cluster centroid for justification

```
for i, pred in enumerate(sample_preds):
    print "Sample point", i, "predicted to be in Cluster", pred
    print 'The distance between sample point {} and center of cluster {}:'.format(i, pred)
    print (samples.iloc[i] - true_centers.iloc[pred])
```

Conclusion

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

> "the distributor should do separate A/B tests for the two clusters. First do A/B tests for high revenue/high volume customers of Cluster 0. Then do the same separately for corresponding customers of Cluster 1."

This comment is key! 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).
https://en.wikipedia.org/wiki/A/B_testing#Segmentation_and_targeting
https://stats.stackexchange.com/questions/192752/clustering-and-a-b-testing

The two clusters that we have in our model reveal two different consumer profiles that can be tested via A/B test. To better assess the impact of the changes on the delivery service, we would have to split the segment 0 and segment 1 into subgroups measuring its consequences within a delta time. Hypothetically we can raise a scenario where the segment 0 is A/B tested. For this we divide the segment 0 (can also be implemented in segment 1) into two sub-groups of establishments where only one of them would suffer the implementation of the new delivery period of three days a week, and the another would remain as a control with five days a week as usual. After a certain period of time, we could, through the consumption levels of the establishments, come to some conclusions, such as: whether the new frequency of deliveries is sufficient or not for a buyer. Where a sensible increase in overall consumption of all products may indicate the need for the establishment to maintain a storage because of the decreasing delivery frequency; or if it negatively affects the consumption profile of certain products, like groups of costumers who have greater buying fresh produce that can be negatively impacted, precisely because of the demand for fresh products with a higher delivery frequency. We can not say that the change in frequency will affect equally all customers because of the different consumption profiles that are part of the two segments. There will therefore consumers that will be affected, and possibly groups of buyers who will not undergo any change.

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

Nice step by step process and good 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.

> "The clusters shown in this plot with channel data, compare well with the clusters obtained earlier, see notebook cell In [25]: (two clusters, Cluster 0, and Cluster 1)."

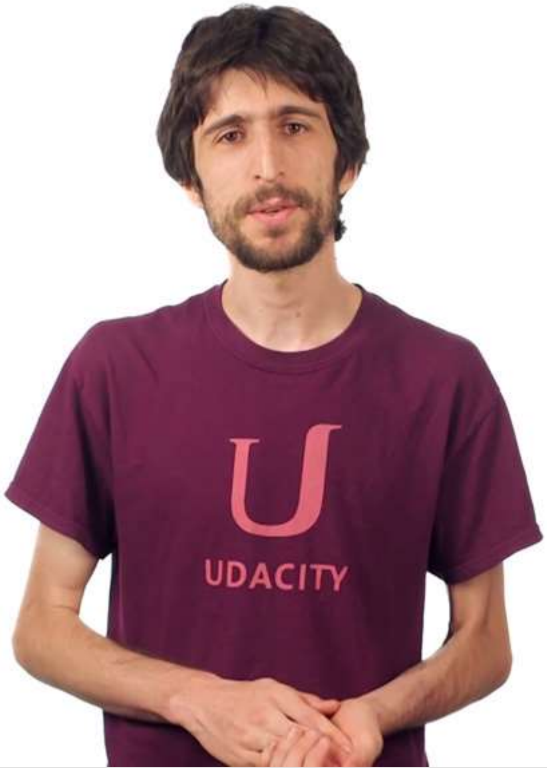
Would agree. Real world data is really never perfectly linearly separable but it seems as your clustering algorithm did a decent job.

Maybe also fully examine how well the clustering algorithm did!

```
#find percentage of correctly classified customers
data = pd.read_csv("customers.csv")
data = data.drop(data.index[outliers_frequent]).reset_index(drop = True)
# might need to switch around the 0 and 1, based on your cluster seed
df = np.where(data['Channel'] == 2, 1, 0)
print "Percentage of correctly classified customers: {:.2%}".format(sum(df == preds)/float(len(preds)))
```

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

Student FAQ