UDACITY

PROJECT

Creating Customer Segments

A part of the Machine Learning Engineer Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

Awesome work! This is a very solid analysis and it definitely seems you have spent a lot of time with your answers here. Just have one last section to perfect, but I bet you will never switch these around again(sorry for the previous reviewer). We look forward to seeing your perfected submission!!!

Data Exploration

Three separate samples of the data are chosen and their establishment representations are proposed based on the statistical description of the dataset.

Very nice justification for your samples here by comparing the purchasing behavior to the mean of the dataset, impressive. Just note that it may be more appropriate to use the median/quartiles rather than the mean, since the median/quartiles are more robust to outliers, which we have here. But great job!

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

"*The reported prediction score was -0.429. This implies that this feature is necessary to predict customers' spending habits, because its value cannot be predicted from the rest of the features, therefore it contains non-redundant information.*"

Correct here, as Deli is independent, so necessary to predict customers' spending habits.

Code Note: Instead of having to drop all other features with

```
Y_data.drop(['Fresh', 'Milk', 'Grocery', 'Frozen', 'Detergents_Paper'], axis = 1, inplace = True)
```

Could simply do

```
Y_data = Y_data['Delicatessen']
```

Could also just do

```
print regressor.score(X_test, y_test)
```

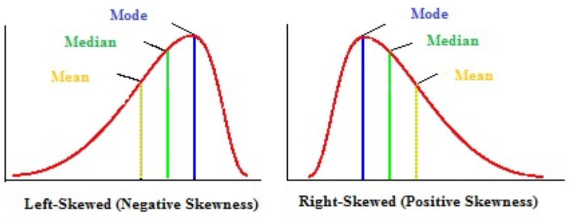
Student identifies features that are correlated and compares these features to the predicted feature. Student further discusses the data distribution for those features.

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, I apologize for marking this as *Requires Changes*, but it seems as the previous reviewer must have missed this(not sure how). But relook at your comment of "*The curve is not symmetrical, and the mean seems skewed to the left, with a lot more points to the right of the mean than the left.*" As I believe that you have the skewness and where most of the data point lie switched. Therefore check out this visual and link

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


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.

Nice coding to discover the indices of the five data points which are outliers for more than one feature of `[65, 66, 75, 128, 154]`.

Would suggest expanding on why we should remove outliers in general as well. As how can these affect the analysis? Why remove any data points? How could these outliers affect distributions? other algorithms? etc...

(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[[o for o in c.keys() if c[o]>1]]
```

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 also use `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)

I won't mark this one as *Requires Changes*, but your interpretation is a bit off here, as the sign of the component really wouldn't matter, since PCA deals with the variance of the data and the correlation between features. As you say "*In terms of customer spending, this dimension generally represents very low spending in detergents, and low spending in groceries and milk, and lower spending in fresh and frozen.*" Thus in terms of customers, 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.

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.

Would recommend also discussing the other main advantages of the two. K-Means speed and GMM's soft clustering. 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 as we can clearly see that K = 2 gives the highest silhouette score. Would recommend doing this programmatically with the use of a for loop

```
for k in range(2, 7):
    clusterer = GaussianMixture(n_components=k, random_state=1).fit(reduced_data)
    ....
```

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 the means 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.

Great! 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). Impressive!

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 estabelishment to mantain 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 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.

Real world data is really never perfectly linearly separable but it seems as our GMM algorithm did a decent job. Might be interesting to check out the probability for belonging to each cluster for your sample point 2.

☑ RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

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

RETURN TO PATH

Rate this review

Student FAQ