UNSUPERVISED MACHINE LEARNING:

CLUSTERING

COURSE PROJECT:

Creating Customer Segments using.

Wholesale Customer Data Set

IBM / COURSERA

IDDI ABDUL AZIZ

INTRODUCTION

Whenever one need to find the best customer, customer segmentation is the ideal methodology. In this project, I will analyse a dataset containing data on various customers' annual spending amounts (reported in *monetary units*) of diverse product categories for internal structure. One goal of this project is to best describe the variation in the different types of customers that a wholesale distributor interacts with. Doing so would equip the distributor with insight into how to best structure their delivery service to meet the needs of each customer.

In this Machine learning project, I will provide you the background of customer segmentation. Then I will explore the data upon which I will be building the segmentation model. Also, in this machine learning project, we will see the descriptive analysis of our data and then implement several versions of the K-Means algorithm.

Customer Segmentation is one of the most important applications of unsupervised learning. Using clustering techniques, companies can identify the several segments of customers, allowing them to target the potential user base. In this project, I will make use of the **K-Means Clustering** which is the essential algorithm for clustering unlabelled dataset.

The dataset for this project can be found on the UCI Machine Learning Repository. For the purposes of this project, the features 'Channel' and 'Region' will be excluded in the analysis — with focus instead on the six product categories recorded for customers.

BRIEF DESCRIPTION OF THE DATA SET AND SUMMARY OF ITS ATTRIBUTES

In this section, I will begin exploring the data through visualizations and code to understand how each feature is related to the others. I will observe a statistical description of the dataset, consider the relevance of each feature, and select a few sample data points from the dataset which I will track through the course of this project, with that I will gain necessary insights about the data.

[1]:
```
# Import libraries necessary for this project
import numpy as np
import pandas as pd
from IPython.display import display # Allows the use of display() for DataFrames

# Import supplementary visualizations code visuals.py
import visuals as vs

# Pretty display for notebooks
%matplotlib inline

# Load the wholesale customers dataset
try:
    data = pd.read_csv("customers.csv")
    data.drop(['Region', 'Channel'], axis = 1, inplace = True)
    print("Wholesale customers dataset has {} samples with {} features each.".format(*data.shape))
except:
    print("Dataset could not be loaded. Is the dataset missing?")
```

```
Wholesale customers dataset has 440 samples with 6 features each.
```

[3]:
```
data.head()
```

[3]:

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
0	12669	9656	7561	214	2674	1338
1	7057	9810	9568	1762	3293	1776
2	6353	8808	7684	2405	3516	7844
3	13265	1196	4221	6404	507	1788
4	22615	5410	7198	3915	1777	5185

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
count	440.000000	440.000000	440.000000	440.000000	440.000000	440.000000
mean	12000.297727	5796.265909	7951.277273	3071.931818	2881.493182	1524.870455
std	12647.328865	7380.377175	9503.162829	4854.673333	4767.854448	2820.105937
min	3.000000	55.000000	3.000000	25.000000	3.000000	3.000000
25%	3127.750000	1533.000000	2153.000000	742.250000	256.750000	408.250000
50%	8504.000000	3627.000000	4755.500000	1526.000000	816.500000	965.500000
75%	16933.750000	7190.250000	10655.750000	3554.250000	3922.000000	1820.250000
max	112151.000000	73498.000000	92780.000000	60869.000000	40827.000000	47943.000000

Implementation: Selecting Samples

In other to get a better understanding of the customers and how their data will transform through the analysis, I selected a few sample data points and explore them in more detail.

Chosen samples of wholesale customers dataset:

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
0	16117	46197	92780	1026	40827	2944
1	112151	29627	18148	16745	4948	8550
2	3	333	7021	15601	15	550

Considering the total purchase cost of each product category and the statistical description of the dataset above for our sample customers. I made a prediction on what kind of customer could each of the three samples we've chosen represent.

The mean values are as follows:

- Fresh: 12000.2977
- Milk: 5796.2
- Grocery: 3071.9
- Detergents_paper: 2881.4
- Delicatessen: 1524.8

Knowing this, how do our samples compare?

1) Index 85: Retailer:
   - Largest spending on detergents and paper and groceries of the entire dataset, which usually are products for houses.
   - Higher than average spending on milk.
   - Lower than average spending on frozen products.

2) Index 181: Large market
   - High spending on almost every product category.
   - Highest spending on fresh products of the entire dataset. Likely to be a large market.
   - Low spending on detergents.

3) Index 338: Restaurant
   - The amount of every product is significantly lower than the previous two customers considered.
   - The spending on fresh products is the lowest of the entire dataset.
   - The spending on milk and detergent and papers is in the bottom quartile.
   - It may be a small and cheap restaurant which needs groceries and frozen food to serve the meals.

Implementation: Feature Relevance

One interesting thought to consider is if one (or more) of the six product categories is actually relevant for understanding customer purchasing. That is to say, is it possible to determine whether customers purchasing some amount of one category of products will necessarily purchase some proportional amount of another category of products? We can make this determination quite easily by training a supervised regression learner on a subset of the data with one feature removed, and then score how well that model can predict the removed feature. Below is the findings on the diagram.

- We tried to predict the Grocery feature.
- The reported prediction score was 67.25%.

- As we obtained high score, it as indicator of a very good fit. So this feature is easy to predict considering the rest of spending habits and, therefore, not very necessary for identifying customers' spending habits.

[5]:
```
data.head(1)
```

[5]:

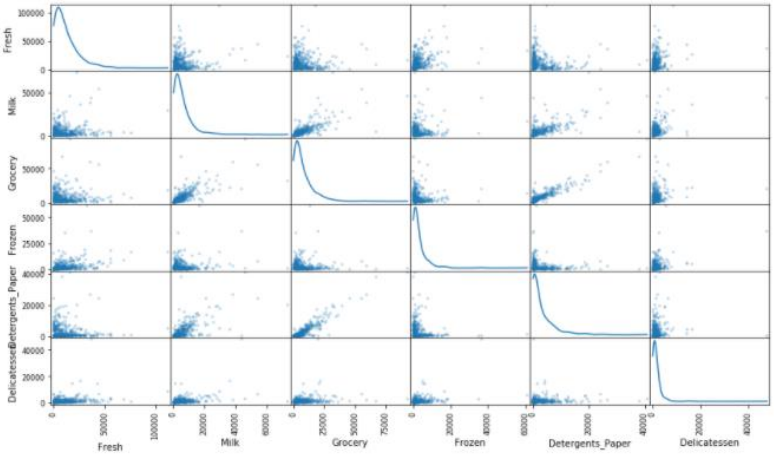
	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
0	12669	9656	7561	214	2674	1338

[6]:
```
# Make a copy of the DataFrame, using the 'drop' function to drop the given feature
new_data = data.drop('Grocery', axis=1)

# Split the data into training and testing sets(0.25) using the given feature as the target
# Set a random state.
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(new_data, data.Grocery, test_size=0.25, random_state=42)

# Create a decision tree regressor and fit it to the training set
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor = regressor.fit(X_train, y_train)
prediction = regressor.predict(X_test)

# Report the score of the prediction using the testing set
from sklearn.metrics import r2_score
score = r2_score(y_test, prediction)
print("Prediction score is: {}".format(score))
```

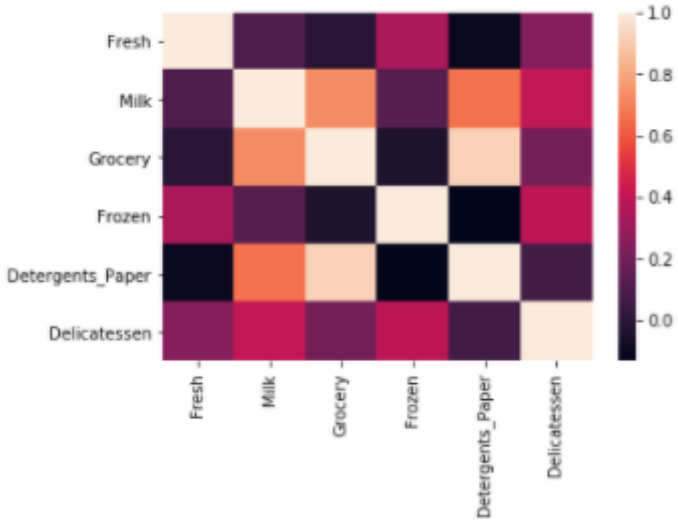
```
Prediction score is: 0.6725491826461042
```

Visualize Feature Distributions

To get a better understanding of the dataset, we can construct a scatter matrix of each of the six product features present in the data. Features relevant to predict the above, won't show a correlation in the scatter matrix and the inverse is also true.




```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6c61500c18>
```



Using the scatter matrix as a reference, we can discuss the following:

- Data is not normally distributed, it is positively skewed and they resemble the log-normal distribution.
- In most plots, most data points lie near the origin which shows little correlation between them.
- From the scatter plots and the heatmap of correlation, we can see that there is a strong correlation between the *'Grocery'* and *'Detergent_paper'* features. The features *'Grocery'* and *'Milk'* also show a good degree of correlation.

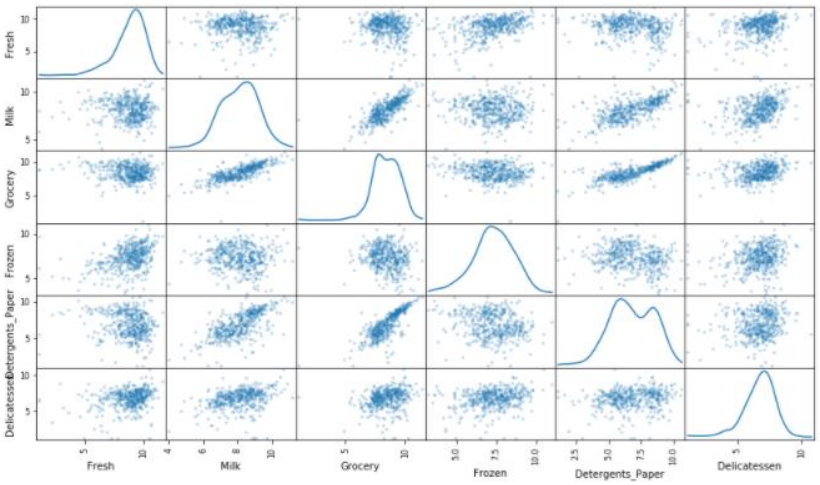
- This correlation confirms my guess about the relevance of the *'Grocery'* feature, which can be accurately predicted with the *'Detergent_paper'* feature. And, therefore, is not an absolutely necessary feature in the dataset.

Data Pre-processing

In this section, we will pre-process the data to create a better representation of customers by performing a scaling on the data and detecting (and optionally removing) outliers. Pre-processing data is often times a critical step in assuring that results we obtain from our analysis are significant and meaningful.

Implementation: Feature Scaling

If data is not normally distributed, especially if the mean and median vary significantly (indicating a large skew), it is most often appropriate to apply a non-linear scaling — particularly for financial data. One way to achieve this scaling is by using a Box-Cox test, which calculates the best power transformation of the data that reduces skewness. A simpler approach which can work in most cases would be applying the natural logarithm.



Observation

After applying a natural logarithm scaling to the data, the distribution of each feature appears much more normal. For any pairs of features we have identified earlier as being correlated, we observe here that correlation is still present (and whether it is now stronger or weaker than before).

Implementation: Outlier Detection

Detecting outliers in the data is extremely important in the data pre-processing step of any analysis. The presence of outliers can often skew results which take into consideration these data points. There are many *"rules of thumb"* for what constitutes an outlier in a dataset. Here, we will use Tukey's Method for identifying outliers: An *outlier step* is calculated as 1.5 times the interquartile range (IQR). A data point with a feature that is beyond an outlier step outside of the IQR for that feature is considered abnormal.

Observations

- Datapoints considered outliers that are present in more than one feature are: 65, 66, 75, 128, 154.

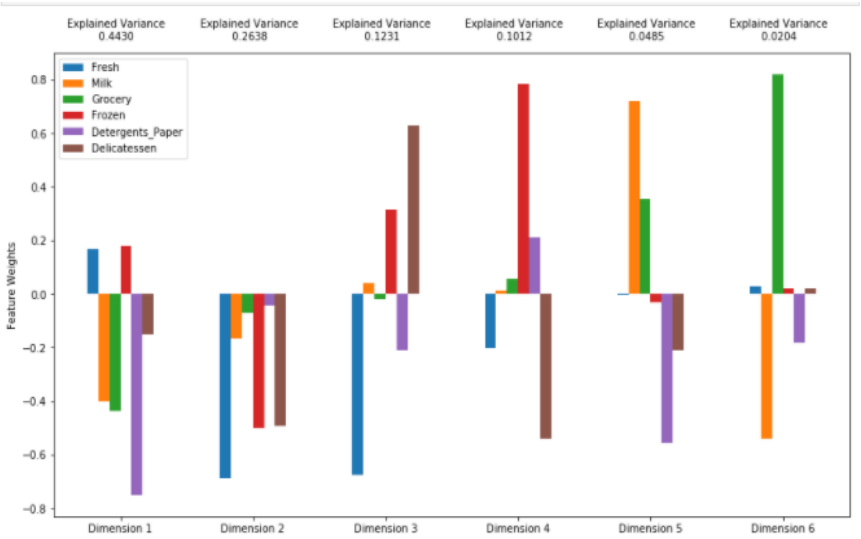
- K-Means is heavily influenced by the presence of outliers as they increase significantly the loss function that the algorithm tries to minimize. This loss function is the squared sum of the distances of each datapoint to the centroid, so, if the outlier is far enough, the centroid will be incorrectly situated. Because of this, the outliers should be removed.

Feature Transformation

In this section I used principal component analysis (PCA) to draw conclusions about the underlying structure of the wholesale customer data. Since using PCA on a dataset calculates the dimensions which best maximize variance, with that I will find which compound combinations of features best describe customers.

Implementing: PCA

Now that the data has been scaled to a more normal distribution and has had any necessary outliers removed, I can then applied PCA to the good_data to discover which dimensions about the data best maximize the variance of features involved. In addition to finding these dimensions, PCA also report the *explained variance ratio* of each dimension — how much variance within the data is explained by that dimension alone. Note that a component (dimension) from PCA can be considered a new "feature" of the space, however it is a composition of the original features present in the data.



Observations

- The variance explained by the first two Principal Components is the 70.68% of the total.
- The variance explained by the first three Principal Components is the 93.11% of the total.

- Dimensions discussion:
  - Dimension 1: This dimension represents well, in terms of negative variance, the following features: *Detergent_Paper*, *Milk* and *groceries*. Mostly utilities for everyday consuming.
  - Dimension 2: This dimension represents well, in terms of negative variance, the following features: *Fresh, Frozen* and *Delicatessen*. Mostly food consuming.
  - Dimension 3: This dimension represents well, in terms of positive variance, the *Delicatessen* features, and in terms of negative variance the *Fresh* feature. Food to be consumed on the day.
  - Dimension 4: This dimension represents well, in terms of positive variance, the Frozen feature, and in terms of negative variance, the *Delicatessen* Feature. Food that can be storage.

Observation

The code below shows how the log-transformed sample data has changed after having a PCA transformation applied to it in six dimensions. Observe the numerical value for the first four dimensions of the sample points.

```
# Display sample log-data after having a PCA transformation applied
display(pd.DataFrame(np.round(pca_samples, 4), columns = pca_results.index.values))
```

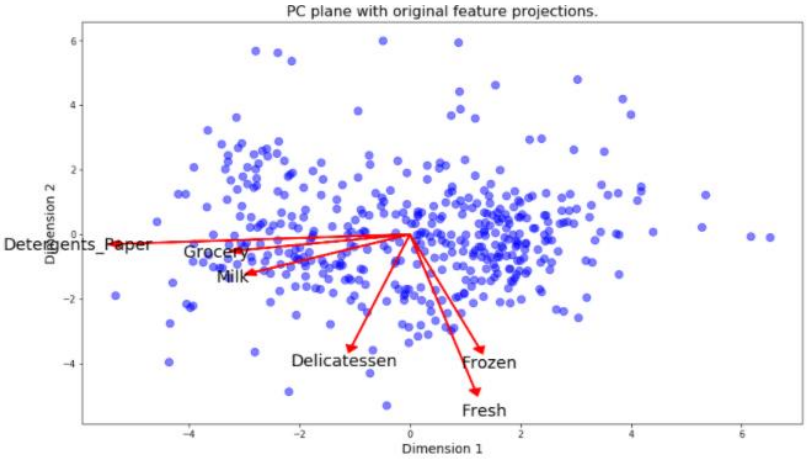
	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6
0	-5.3316	-1.8845	-0.6957	-0.1972	0.5461	0.3802
1	-2.1899	-4.8605	0.0008	0.4827	0.5041	-0.1988
2	3.0206	4.8169	6.4519	2.7403	0.7788	2.1415

Implementation: Dimensionality Reduction

I used principal component analysis, to reduce the dimensionality of the data its as one of the main goals, in effect, reducing the complexity of the problem. Dimensionality reduction comes at a cost: Fewer dimensions used implies less of the total variance in the data is being explained. Because of this, the *cumulative explained variance ratio* is extremely important for knowing how many dimensions are necessary for the problem. Additionally, if a significant amount of variance is explained by only two or three dimensions, the reduced data can be visualized afterwards.

Visualizing a Biplot

A biplot is a scatterplot where each data point is represented by its scores along the principal components. The axes are the principal components (in this case Dimension 1 and Dimension 2). In addition, the biplot shows the projection of the original features along the components. A biplot can help us interpret the reduced dimensions of the data, and discover relationships between the principal components and original features.



Observation

Once we have the original feature projections (in red), it is easier to interpret the relative position of each data point in the scatterplot. For instance, a point the lower right corner of the figure will likely correspond to a customer that spends a lot on 'Milk', 'Grocery' and 'Detergents_Paper', but not so much on the other product categories.

Clustering

In this section, I will either use a K-Means clustering algorithm or a Gaussian Mixture Model clustering algorithm to identify the various customer segments hidden in the data. I will then recover specific data points from the clusters to understand their significance by transforming them back into their original dimension and scale.

1) The main advantages of using K-Means as a cluster algorithm are:

- It is easy to implement.
- With large number of variables, if (K is small) it may be computationally faster than hierarchical clustering.
- Consistent and scale-invariant.
- It is guaranteed to converge.

2) The main advantages of using Gaussian Mixture Models as a cluster algorithm are:

- It is much more flexible in terms of cluster covariance. Which means that each cluster have unconstrained covariance structure. In other words, whereas K-means assumes that every cluster have spherical structure, GMM allows elliptical.
- Points can belong to different clusters, with different level of membership. This level of membership is the probability of each point to belong to each cluster.

3) Chosen algorithm:

- The chosen algorithm is Gaussian mixture model. Because data is not splitted in clear and different clusters, so I do not know how many clusters are there.

Implementation: Creating Clusters

I quantify the "goodness" of a clustering by calculating each data point's *silhouette coefficient*. The silhouette coefficient for a data point measures how similar it is to its assigned cluster from -1 (dissimilar) to 1 (similar). Calculating the *mean* silhouette coefficient provides for a simple scoring method of a given clustering.

```
Number of clusters: 2
Cluster Center: [[ 1.2512378  -0.18013806]
 [-2.22116886  0.31977698]]
Sample predictions: [1 1 0]
Silhouette score is: 0.421916846463

Number of clusters: 3
Cluster Center: [[ 1.3874837  -0.24378903]
 [-2.04781298 -0.05820189]
 [-1.08932369  1.063399  ]]
Sample predictions: [2 2 2]
Silhouette score is: 0.404248738241

Number of clusters: 4
Cluster Center: [[ 1.36580355  0.2583133 ]
 [-2.3145402   0.01022103]
 [-0.32043976  1.79401921]
 [ 1.17387101 -0.79626779]]
Sample predictions: [2 1 2]
Silhouette score is: 0.293269564847

Number of clusters: 5
Cluster Center: [[ 1.71130552 -0.22201663]
 [-2.08799545 -0.5201485 ]
 [ 2.20617662  2.92837026]
 [ 0.03785843 -0.70356503]
 [-2.79826984  1.54857783]]
Sample predictions: [1 3 2]
Silhouette score is: 0.300456388725

Number of clusters: 6
Cluster Center: [[ 0.62535432  0.56453323]
 [-1.99042622 -0.67842086]
 [ 2.11689604  3.15226455]
 [ 0.50197961 -1.80803617]
 [-2.79898205  1.72544922]
 [ 1.93033498 -0.34721167]]
Sample predictions: [1 3 2]
Silhouette score is: 0.326139450471

Scores: {2: 0.42191684646261496, 3: 0.40424873824078805, 4: 0.29326956484658406, 5: 0.30045638872525937, 6: 0.32613
945047115767}
```

Answer:

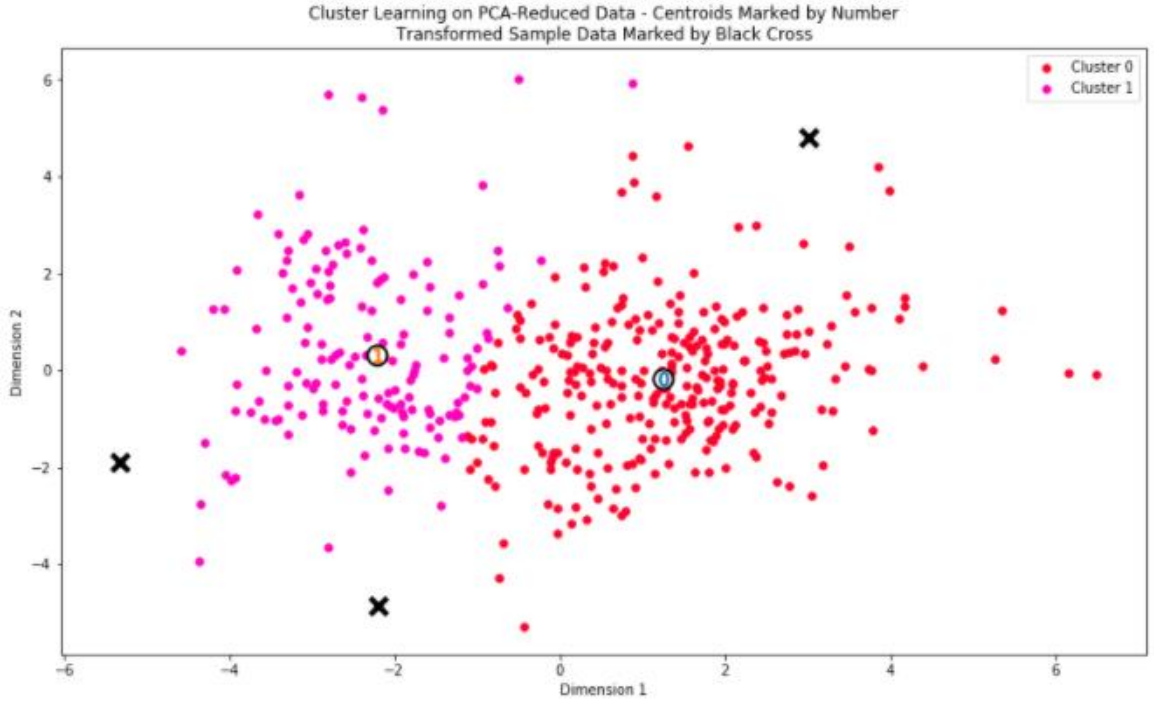
- The Silhouette score for each number of clusters (from 2 to 6) are:

Scores: {2: 0.42191684646261496, 3: 0.40424873824078805, 4: 0.29326956484658406, 5: 0.30045638872525937, 6: 0.32613945047115767}

- The number of clusters with the best silhouette score is 2, with a 0.42 score.

Cluster Visualization

Once I've chosen the optimal number of clusters for clustering algorithm using the scoring metric above, I can now visualise the results below.



Implementation: Data Recovery

From the above visualisation each cluster present in the visualisation above has a central point. These centres (or means) are not specifically data points from the data, but rather the *averages* of all the data points predicted in the respective clusters. For the problem of creating customer segments, a cluster's centre point corresponds to *the average customer of that segment*. Since the data is currently reduced in dimension and scaled by a logarithm, I can recover the representative customer spending from these data points by applying the inverse transformations.

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
Segment 0	8953.0	2114.0	2765.0	2075.0	353.0	732.0
Segment 1	3552.0	7837.0	12219.0	870.0	4696.0	962.0

Observations

- Segment 0 may represent a fresh food market as every feature except Frozen and Fresh are below the median.
- Segment 1 may represent a supermarket as every feature except fresh and frozen are above the median.

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicatessen
count	440.000000	440.000000	440.000000	440.000000	440.000000	440.000000
mean	12000.297727	5796.265909	7951.277273	3071.931818	2881.493182	1524.870455
std	12647.328865	7380.377175	9503.162829	4854.673333	4767.854448	2820.105937
min	3.000000	55.000000	3.000000	25.000000	3.000000	3.000000
25%	3127.750000	1533.000000	2153.000000	742.250000	256.750000	408.250000
50%	8504.000000	3627.000000	4755.500000	1526.000000	816.500000	965.500000
75%	16933.750000	7190.250000	10655.750000	3554.250000	3922.000000	1820.250000
max	112151.000000	73498.000000	92780.000000	60869.000000	40827.000000	47943.000000

The code block below shows to which each sample point is predicted to be.

```
# Display the predictions
for i, pred in enumerate(sample_preds):
    print("Sample point", i, "predicted to be in Cluster", pred)
```

```
Sample point 0 predicted to be in Cluster 1
Sample point 1 predicted to be in Cluster 1
Sample point 2 predicted to be in Cluster 0
```

Observations

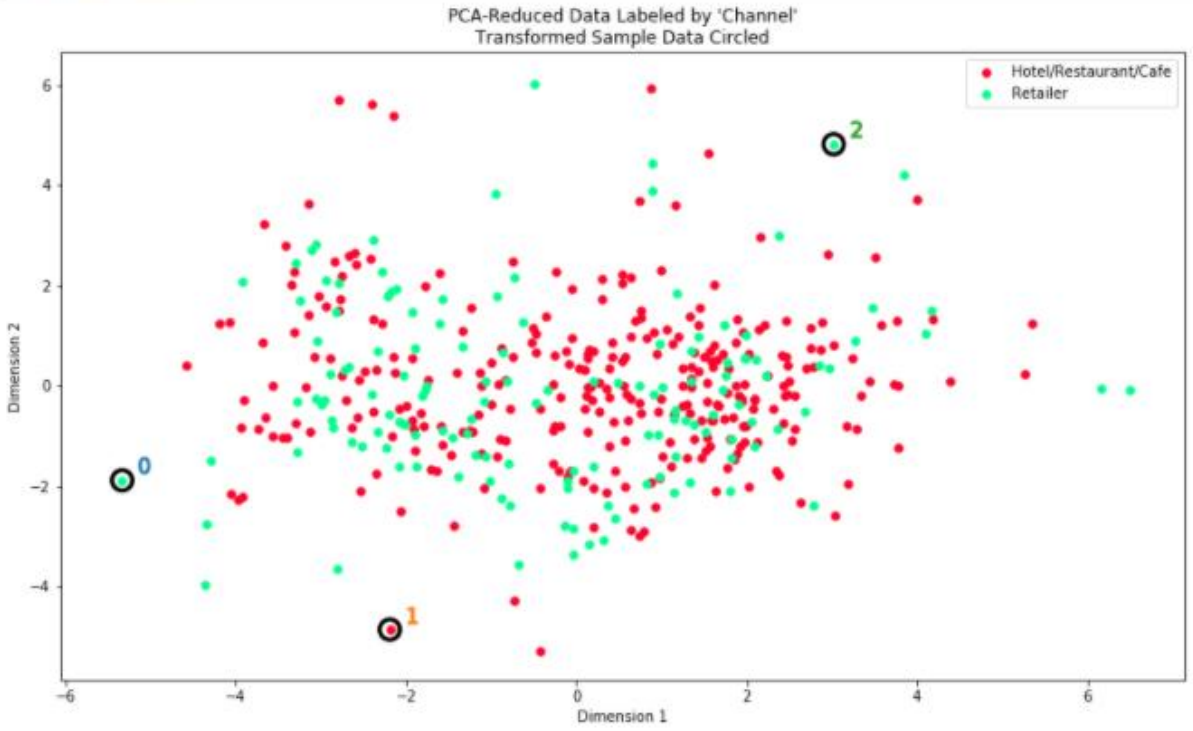
- Sample point 0 --> Supermarket and the original guess was a retailer. This difference may be explained because of the size of the cluster (which is pretty big)
- Sample point 1 --> Supermarket and the original guess was the same.
- Sample point 2 --> Fresh food market and the original guess was a restaurant which is reasonable considering the amount of the spending of the features.

Visualizing Underlying Distributions

At the beginning of this project, it was discussed that the 'Channel' and 'Region' features would be excluded from the dataset so that the customer product categories were emphasized in the analysis. By reintroducing the 'Channel' feature to the dataset, an interesting structure emerges when considering the same PCA dimensionality reduction applied earlier to the original dataset.

The code block below shows how each data point is labeled either 'HoReCa' (Hotel/Restaurant/Cafe) or 'Retail' the reduced space.

```
# Display the clustering results based on 'Channel' data
vs.channel_results(reduced_data, preds, pca_samples)
```



Observations

- The cluster algorithm does a pretty good job of clustering the data to the underlying distribution as the cluster 0 can be associated perfectly with a retailer and the cluster 1 to the Ho/Re/Ca.
- Yes, these definitions are consistent with the previous definitions of customer segments.