Creating Customer Segments

Table of Contents

Project Description

To help a wholesale grocery distributor determine which changes will benefit their business. They recently tested out a change to their delivery method, from a regular morning delivery to a cheaper, bulk evening delivery. Initial tests didn't discover any significant effect, so they implemented the cheaper option. Almost immediately, they began getting complaints about the change and losing customers. As it turns out, the highest volume customers had an easy time adapting to the change, whereas smaller family run shops had serious issues with it. However, these issues were washed out statistically by noise from the larger customers.
For the future, they want to have a sense of what sorts of different customers they have. Then, when implementing changes, they can look at the effects on these different groups independently.

Project Aim

The task is to use unsupervised techniques to see what sort of patterns exist among existing customers, and what exactly makes them different.

Data Source

https://github.com/udacity/machine-learning/tree/master/projects/creating_customer_segments\customers.csv

Software and Libraries

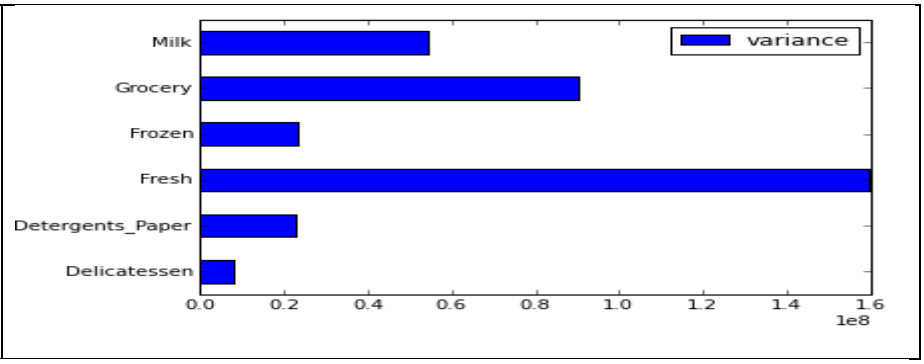
- Python 2.7
- NumPy 1.10
- Pandas
- scikit-learn 0.17
- iPython Notebook (with iPython 4.0)

Component analysis

Reflection on PCA/ICA

PCA seeks to reduce the dimension of the data by finding a few orthogonal linear combinations of the original variables with the largest variance [1]. Therefore, when performing the variance computation, we clearly obtain two main principal components: 'Fresh' (with a variance of 1.6+E08) and 'Grocery' (with a variance of 9+E07).

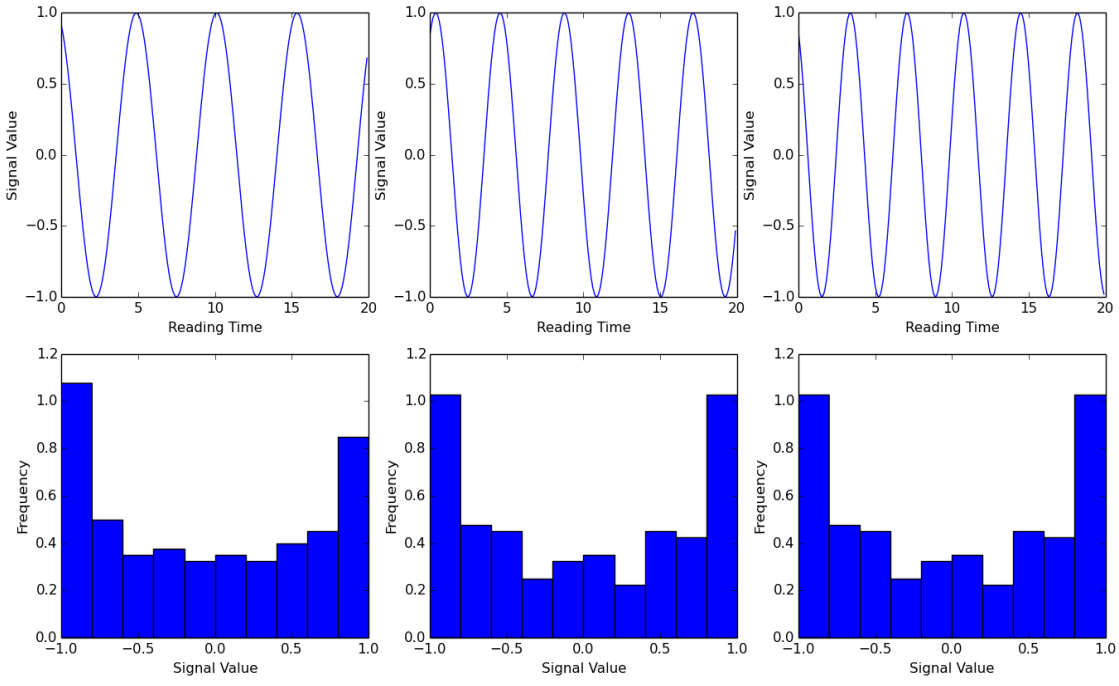
	variance
Delicatessen	7.934923e+06
Detergents_Paper	2.268077e+07
Fresh	1.595914e+08
Frozen	2.351429e+07
Grocery	9.010485e+07
Milk	5.434617e+07



ICA on the other hands focuses on seeking linear projections, not necessarily orthogonal to each other, that are as nearly statistically independent as possible [1].

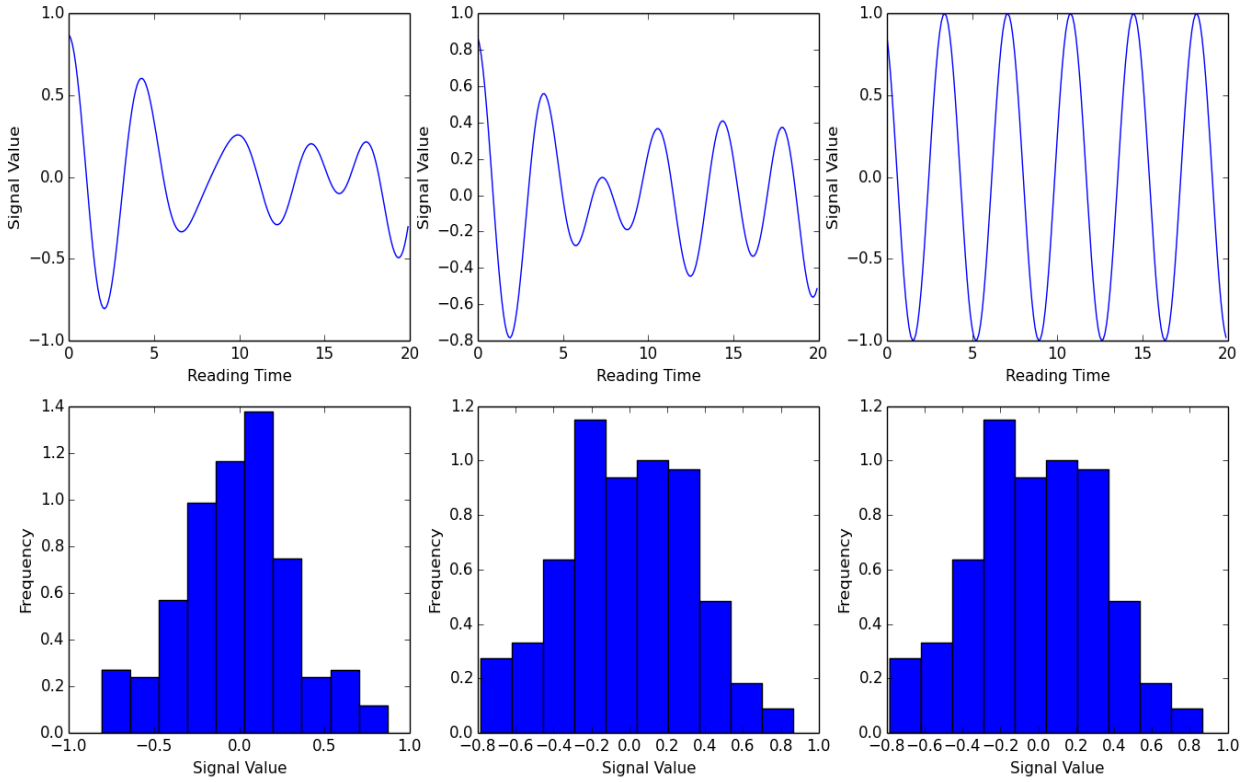
ICA looks for a linear combinations (weighted sums) of the original features such that recovered features have a distribution that is as far from a normal distribution as possible.
You can see why by minimizing the degree to which the recovered features fit a normal distribution we can recover independent features if you assume that the original features are independent and not normally distributed such as:

s**1** = np.sin(1.2*t+2) || **s2** = np.sin(1.5*t+1) || **s3** = np.sin(1.7*t+2.1)



Then by the central limit theorem, any mixing (linear combination) of the original features will have a distribution closer to a normal distribution such as:

m1 = 0.3*s1 + 0.4*s2 + 0.3*s3 || **m2** = 0.2*s1 + 0.3*s2 + 0.5*s3 ||**m3** = 0.4*s1 + 0.2*s2 + 0.4*s3



So ICA recovers the original signals by minimizing the degree to which each of the recovered features has a normal distribution. Note however that if the original signals are in fact normally distributed, even if they are correlated, ICA will not be able to recover them.

[1] Imola K. Fodor (June 2002), *A survey of dimension reduction techniques*, Lawrence Livermore National Laboratory, UCRL-ID148494, p2 and p6.

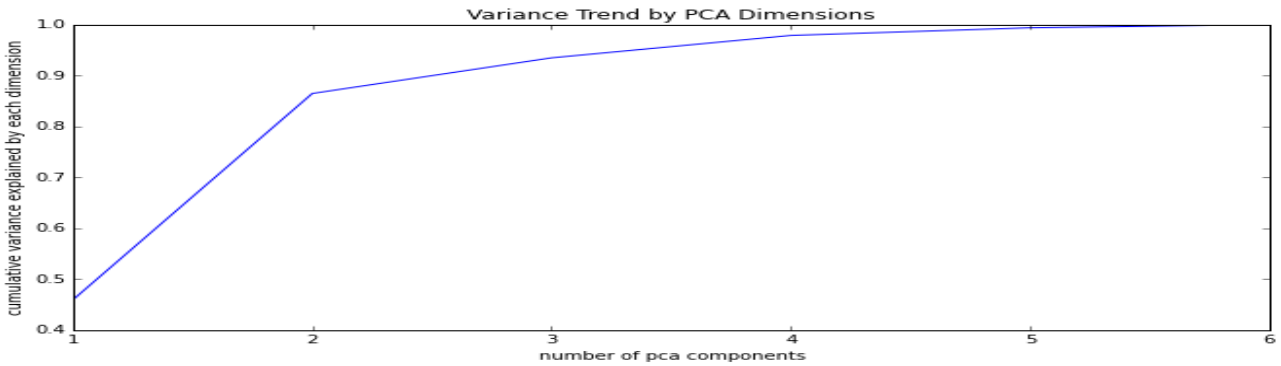
Proportion of variance explained by each PCA dimension

```
**********************************
PCA Components
**********************************
[[-0.97653685 -0.12118407 -0.06154039 -0.15236462  0.00705417 -0.06810471]
 [-0.11061386  0.51580216  0.76460638 -0.01872345  0.36535076  0.05707921]
 [-0.17855726  0.50988675 -0.27578088  0.71420037 -0.20440987  0.28321747]
 [-0.04187648 -0.64564047  0.37546049  0.64629232  0.14938013 -0.02039579]
 [ 0.015986    0.20323566 -0.1602915   0.22018612  0.20793016 -0.91707659]
 [-0.01576316  0.03349187  0.41093894 -0.01328898 -0.87128428 -0.26541687]]

**********************************
PCA Explained Variance Ratio
**********************************
[ 0.45961362  0.40517227  0.07003008  0.04402344  0.01502212  0.00613848]
```

When the number of component selected is set to the number of dimensions (i.e. 6), the variance drop off by between the first and the second component, i.e. from 0.46 to 0.40 is relatively small. This is represented a drop of 13%, calculated as (0.4-0.46)/.46. However, the drop between the second and third component is very large, 82% from 0.4 to 0.07. This is the point of diminishing returns.

Two components contain 86% (0.46+0.40) of the variance of the 6 original values. 'Component 1' explains 46% of the variation, 'component 2' explains 40%, and the rest of the components explain 14% of the variation. If we assume we want to retain 85% of the variance, then we should reduce the feature set to the first 2 components.



This is also shown by the graph above; there is a steep increase of the cumulative variance explained by each dimension (y-axis) until the second PCA component. Then the curve plateaus.

‘Component 1’ corresponds to the ‘Fresh’ feature as it represents more than 97% of the variance. ‘Component 2’ corresponds to ‘Milk’, ‘Grocery’ and ‘Detergents_Paper’. They account respectively 51%, 76% and 36% of the variance.

This is also shown by the correlation below. The cell highlighted in yellow show abs(correlation) > 0.5. The ‘Fresh’ only depends on itself. The ‘Milk’, ‘Grocery’ and ‘Detergents_Paper’ show high degree of correlation:

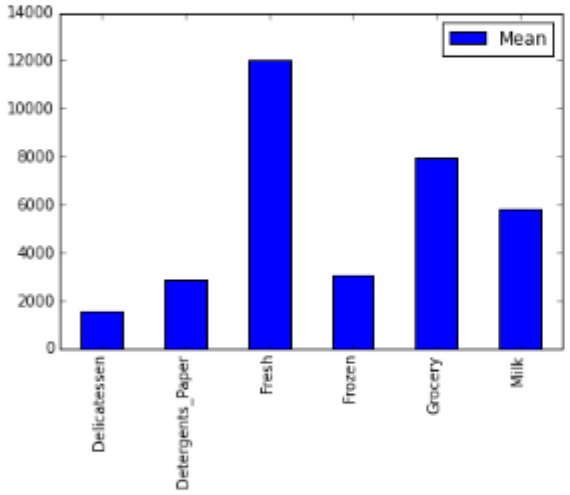
- ‘Milk & Grocery’ shows a correlation of 73%
- ‘Milk & Detergents Paper ‘shows a correlation of 66%
- ‘Detergents_Paper & Grocery ‘shows a correlation of 92%

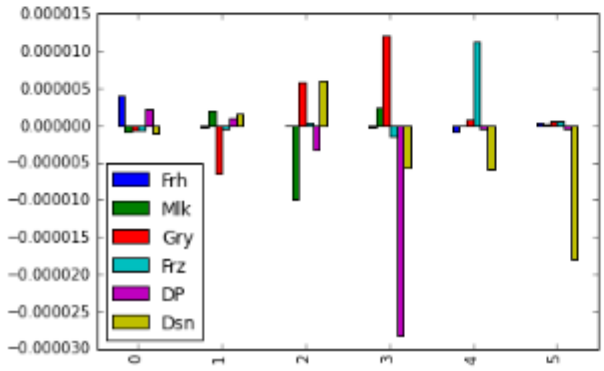
```
**********************************
Correlation Matrix
**********************************
                   Fresh      Milk   Grocery    Frozen  Detergents_Paper
Fresh           1.000000  0.100510 -0.011854  0.345881        -0.101953
Milk            0.100510  1.000000  0.728335  0.123994         0.661816
Grocery        -0.011854  0.728335  1.000000 -0.040193         0.924641
Frozen          0.345881  0.123994 -0.040193  1.000000        -0.131525
Detergents_Paper -0.101953  0.661816  0.924641 -0.131525         1.000000
Delicatessen    0.244690  0.406368  0.205497  0.390947         0.069291
```

This information is useful as a feature reduction tool (with minimum loss of information) to feed into a machine learning algorithm. By reducing the number of factors, a simpler (and usually faster) machine learning algorithm can be selected.

Proportion of variance explained by ICA

```
***********************************
Component data and plot
***********************************
[[  3.97599355e-06  -8.59886636e-07  -6.28711730e-07  -6.77060709e-07
    2.07067530e-06  -1.04098223e-06]
 [ -2.10115505e-07   1.88251426e-06  -6.45028663e-06  -4.08141212e-07
    8.61829574e-07   1.46521187e-06]
 [ -1.53807253e-07  -9.84447608e-06   5.81239885e-06   3.63403734e-07
   -3.32305393e-06   6.05623018e-06]
 [ -2.99003233e-07   2.31324193e-06   1.20452467e-05  -1.46364692e-06
   -2.82034564e-05  -5.73086477e-06]
 [ -8.65159287e-07  -1.40577395e-07   7.73992444e-07   1.11462064e-05
   -5.54399614e-07  -5.95213935e-06]
 [  3.86475758e-07   2.19541928e-07   6.00294516e-07   5.22080384e-07
   -5.09796428e-07  -1.80924000e-05]]
```





The Independent Component Analysis (ICA) attempts to decompose a multivariate signal into independent non-Gaussian signals. It is primarily used to 'untangle 'observed data and calculate source signals which, in varying linear combinations, can make the observed data. In other words, as the absolute value of the elements of the un-mixing matrix increases, the corresponding feature has a strong effect on that component. Therefore the magnitude of the coefficient represents prevalence in that feature.

ICA can help transforming the data (i.e. making each component independent) before running machine learning algorithm that requires data independency (e.g. the regression model or the Mixture of Gaussians clustering)

For these ICA graph we can state the following:

- The ‘Fresh’ dimension (‘0’ on the x-axis) is positively correlated to the ‘Fresh’ feature and 'Detergents Paper', and negatively correlated to all the other features.
- The ‘Milk’ dimension (‘1’ on the x-axis) is positively correlated to the ‘Milk’, 'Detergents Paper' and ‘Delicatessen’, and negatively correlated to all the other features.
- The ‘Grocery’ dimension (‘2’ on the x-axis) is positively correlated to the ‘Grocery’ and ‘Delicatessen ‘, and negatively correlated to all the other features.
- The ‘Frozen’ dimension (‘3’ on the x-axis) is positively correlated to the ‘Milk’, ‘Grocery’ and ‘Grocery’, and negatively correlated to all the other features.
- The ‘Detergents Paper’ dimension (‘4’ on the x-axis) is positively correlated to the ‘Frozen’ and ‘Grocery’, and negatively correlated to all the other features.
- The ‘Delicatessen’ dimension (‘5’ on the x-axis) is positively correlated to all but the 'Detergents Paper'.

Looking at the graph, it looks like the dimensions with the highest positive correlations are the 3rd (‘Frozen’) and 4th (‘Detergents Paper’) . The results only agree partially with the PCA.

Therefore, this means that including only these two features in the training of the algorithm would account for maximum coverage of the variability in the dataset with a minimum loss of information.

Clustering

Gaussian mixture models and K-means are two canonical approaches to clustering, i.e. dividing data points into meaningful groups.

- K-means works by iteratively reassigning data points to clusters and computing cluster centres based on the average of the point locations.
- K-means has a limited set of parameters corresponding to the number of clusters, where the Mixture of Gaussians can have many.
- K-means is usually used in the following scenarios: as a general-purpose, even cluster size, flat geometry and where they are not too many clusters.

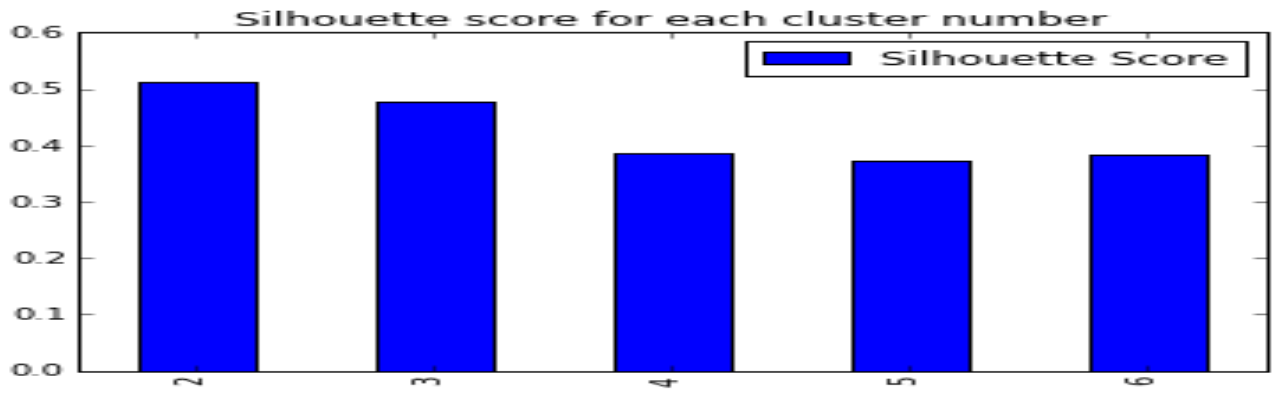
- Gaussians Mixture is a probabilistic model commonly used for clustering: partitioning a set of data points into a set of clusters, where data points within a cluster are similar to one another.
- Gaussians Mixture is not scalable.
- Gaussians Mixture is better at density estimation.
- Gaussians Mixture provides structural information, thus it can measure how wide each cluster is, since it works on probabilities (soft clustering)

In summary, K-means is a more simplistic, more scalable and faster algorithm than the Gaussian mixture models. However, the Mixture of Gaussians estimates a co-variance matrix for every cluster, revealing the shape of every cluster in turn.

For our scenario, it is important to understand the customer size of customer groups (i.e. clusters) to better predict the impact of shops service changes on the overall profitability. For example, a service change could have a major impact on a group of customers. However, if this group is relatively small and have a small negative profit impact, then the change might be pushed forward (assuming this change has positive effects on the other groups) to increase the overall profit. Therefore the Gaussians Mixture will be selected as the machine learning algorithm.

Decide on K means clustering or Gaussian mixture methods

The choice of cluster numbers has been discovered by the 'Silhouette score', as shown in the below graph.
The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar. A higher Silhouette Coefficient score relates to a model with better defined clusters.
The highest 'Silhouette Coefficient' is 0.51 and is represented by 2 clusters. Adding any extra cluster reduces the 'Silhouette Coefficient' (law of diminishing returns).



Clusters Implementation

```
# Import clustering modules
from sklearn.cluster import KMeans
from sklearn.mixture import GMM
```

```
# First we reduce the data to two dimensions using PCA to capture variation
pca_2 = PCA(n_components=2)
reduced_data =  pca_2.fit_transform(data)
print reduced_data[:10]  # print upto 10 elements
```

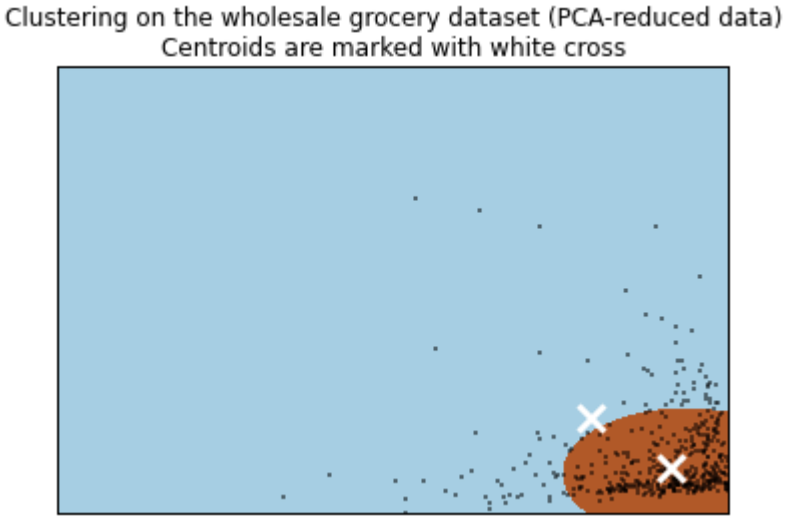
```
[[  -650.02212207   1585.51909007]
 [  4426.80497937   4042.45150884]
 [  4841.9987068    2578.762176  ]
 [  -990.34643689  -6279.80599663]
 [-10657.99873116  -2159.72581518]
 [  2765.96159271   -959.87072713]
 [   715.55089221  -2013.00226567]
 [  4474.58366697   1429.49697204]
 [  6712.09539718  -2205.90915598]
 [  4823.63435407  13480.55920489]]
```

```
# Implement the clustering algorithm here, and fit it to the reduced data for visualization
# The visualizer below assumes your clustering object is named 'clusters'
gmm = GMM(n_components=2)
clusters = gmm.fit(reduced_data)

print clusters
```

```
GMM(covariance_type='diag', init_params='wmc', min_covar=0.001,
  n_components=2, n_init=1, n_iter=100, params='wmc', random_state=None,
  thresh=None, tol=0.001, verbose=0)
```

Graphical View



Central objects in each cluster represent the centroids. They represent the "average" customer in that grouping.
The graph above shows two distinct data clusters. This means there are two groups of customers that have different consuming patterns. This is shown by the fact that one of the group of consumer (in light brown) is concentrated around its centroid, whereas the second group (in blue) is more sparsely allocated around its centroid.
This certainly means brown cluster represents small family run shops, whereas the blue cluster represents large volume customers.

Some of the clusters are not very well distinguished, for example in the middle section of the ‘brown’ area. We could make them appear by increase the cluster number. But this would bear the risk of increasing overfitting.

Conclusions

Natural data fit

- PCA provided with a very clear insight on the main features impacting the model. The point of using PCA was for feature reduction. The dataset provided list 6 features, we saw that retained two of them('Fresh' and 'Groceries' would maintain an overall variance of 86%. In other words, these two variables explains 86% of the variance in the data.
- Gaussians Mixture enabled the bucketing of the customers into their respective market segments and thus facilitates better business decision making. Thanks to the feature reduction provided by the PCA, we could embark on using a Gaussians Mixture clustering algorithm to organise customers in different segments.

Other Technique available…

One popular technique would be to run an AB test. It directly compares a variation against a current experience. The analyst can then produce focused questions about changes to the domain, and then collects data about the impact of that change.

In our case, the following steps could be defined:

- To create a control and experiment group within the market segments. As different retailer sizes provide different customer experience.
- And calculate the p-value for to accept/reject predefined hypothesis relating to consumer pattern given a type of retail premise (e.g. large or small).

Based on the result of this analysis, a marketing strategy could be put in place to satisfy the different customer segments.

Data usage to predict future customer needs

By defining customer segments, it is now possible to we can run different tests (e.g. regression) to improve profit margin and target the supply products of interest to the particular segments.
The clustering produces labels that could be fed into supervised model to predict the customer pattern based on this classification. This could prove useful for the launch of a new shop in a new area.