Creating Customer Segments ¶

In this project you, will analyze a dataset containing annual spending amounts for internal structure, to understand the variation in the different types of customers that a wholesale distributor interacts with.

Instructions:

- Run each code block below by pressing **Shift+Enter**, making sure to implement any steps marked with a TODO.
- Answer each question in the space provided by editing the blocks labeled "Answer:".
- When you are done, submit the completed notebook (.ipynb) with all code blocks executed, as well as a .pdf version (File > Download as).

In [133]:
```
# Import libraries: NumPy, pandas, matplotlib
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Tell iPython to include plots inline in the notebook
%matplotlib inline

# Read dataset
data = pd.read_csv("wholesale-customers.csv")
print "Dataset has {} rows, {} columns".format(*data.shape)
print data.head()  # print the first 5 rows
```

```
Dataset has 440 rows, 6 columns
   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
```

##Feature Transformation

1) In this section you will be using PCA and ICA to start to understand the structure of the data. Before doing any computations, what do you think will show up in your computations? List one or two ideas for what might show up as the first PCA dimensions, or what type of vectors will show up as ICA dimensions.

Answer:

PCA dimensions that will show up 1st are the ones that explain maximum number of variance in the dataset. The features that drive the data most (in terms of variance). PCA will be focused more on the global picture of the dataset (aspects that define large scale terrain).

PCA dimensions that might show up 1st are: fresh as one dimension and grocery as second dimension. Fresh has the maximum range 3 to 112,151 and standard deviation of ~12,647, this seems to suggest maximum variance between customers, and therefore might describe their differences more. Same logic goes for grocery with maximum range 3 to 92,780 and standard deviation of ~9503. However basing both assumptions only on min/max range and std. deviation does not necessarily guarantee that we will get some results from PCA. Another possibility is that fresh will still be the first dimension, but the 2nd dimension will be detergents_paper, this is based on the fact that while all the other features are food (fresh, milk, grocery, frozen, delicatessen), detergents are obviously not food and there might be a large variance between stores (customers) that sell food and nonfood items in this case detergents_paper feature.

ICA dimensions will show up as maximally independent vectors, ICA will try to differentiate between mixed signals (dimensions). Each vector will be a maximally independent feature (description) of the given dataset. ICA will be focused more on the detailed aspects of the dataset.

Given the above hypothesis that basically there are fundamentally only two items (food and nonfood) that the customers sell, ICA is likely to come to this conclusion too as it does look for maximally independent structures (customers) and it makes sense that customers that deal mostly with food (fresh, milk, grocery, frozen, delicatessen) and customers that deal mostly with nonfood (detergents_paper) will be maximally independent.

###PCA

In [190]:
```
# TODO: Apply PCA with the same number of dimensions as variables in t
from sklearn.decomposition import PCA

pca = PCA(n_components=data.shape[1])
pca.fit_transform(data)

# Print the components and the amount of variance in the data containe
#print pca.components_
print pd.DataFrame(pca.components_, columns=data.columns)
print pca.explained_variance_ratio_
```

```
      Fresh      Milk   Grocery    Frozen  Detergents_Paper  Delicat
essen
0 -0.976537 -0.121184 -0.061540 -0.152365          0.007054     -0.0
68105
1 -0.110614  0.515802  0.764606 -0.018723          0.365351      0.0
57079
2 -0.178557  0.509887 -0.275781  0.714200         -0.204410      0.2
83217
3 -0.041876 -0.645640  0.375460  0.646292          0.149380     -0.0
20396
4  0.015986  0.203236 -0.160292  0.220186          0.207930     -0.9
17077
5 -0.015763  0.033492  0.410939 -0.013289         -0.871284     -0.2
65417
[ 0.45961362  0.40517227  0.07003008  0.04402344  0.01502212  0.0061
3848]
```

2) How quickly does the variance drop off by dimension? If you were to use PCA on this dataset, how many dimensions would you choose for your analysis? Why?

```
Answer:

The variance drops off very quickly while 1st and 2nd dimensions are
0.45961362 and  0.40517227 respectively the 3rd dimension is only
0.07003008 that's a drop of almost 6 times from 2nd dimension to 3rd
dimension. 4th, 5th and 6th are 0.04402344, 0.01502212  and
0.00613848 respectively, a drop of almost 1/2 from 3rd to 4th, a drop
of about 3 times from 4th to 5th and a drop of another ~3 times from
5th to 6th.

Running PCA on this dataset, selecting 1st and 2nd dimension should
provide for most of dataset explanation, 1st and 2nd dimensions have
by far the most variance. And together represent ~86% of total
variance.
```

3) What do the dimensions seem to represent? How can you use this information?

Answer:

The 1st (0.45961362) (fresh with some frozen and milk, and others to lesser extent) and 2nd (0.40517227) (detergents_paper with some delicatessen and frozen, and others to lesser extent) PCA dimensions represent the two most varied dimensions the ones that account for most of the variance of the entire dataset. Fresh (with some frozen and milk) and detergents (with some delicatessen and frozen) represent the most descriptive aspects of the warehouse customers. Those features have the most variance between customers.

Given the fact that 1st (fresh with some frozen and milk...) and 2nd dimensions (detergents_paper with some delicatessen and frozen...) represent most of the data (~86% of the total dimensions) and the next dimension represents only ~7%, we can use only the 1st two dimensions to describe most of the data, we can rerun PCA to compress the data to two dimensions to more easily model the data and visualize it (identify customer types), without losing too much of the information.

###ICA

In [195]:
```
# TODO: Fit an ICA model to the data
# Note: Adjust the data to have center at the origin first!
from sklearn.decomposition import FastICA
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
data_scaler = scaler.fit_transform(data)

ica = FastICA(n_components=data.shape[1], random_state=42)
ica.fit(data_scaler)

# Print the independent components
#print ica.components_
print pd.DataFrame(ica.components_, columns=data.columns)
```

```
      Fresh      Milk   Grocery    Frozen  Detergents_Paper  Delicat
essen
0 -0.010908 -0.001086  0.007308  0.054056         -0.002541     -0.0
16757
1  0.002538 -0.012328  0.069129  0.001424         -0.013749     -0.0
05441
2 -0.004906 -0.001539 -0.005621 -0.002525          0.002384      0.0
50929
3 -0.003363  0.018630  0.108990 -0.007232         -0.133386     -0.0
16023
4 -0.050266  0.006472  0.007482  0.003224         -0.011471      0.0
02708
5 -0.001939 -0.072455  0.056476  0.001674         -0.017140      0.0
16956
```

4) For each vector in the ICA decomposition, write a sentence or two explaining what sort of object or property it corresponds to. What could these components be used for?

Answer:

The components that arise from ICA are 6 maximally independent vectors that pickup on the detailed aspects of data (from the 6 input dimensions that might have been mixed obfuscating the information/possible structure). 6 maximally independent features of the dataset.

ICA is not used to compress dimensionality (unlike PCA), it's used to clear up the "signal", to separate most independent information, to potentially reveal patterns that might have been hidden by possible mixture/noise in the dataset.

IC-0 suggests more frozen vs milk or detergents_paper

IC-1 suggests more grocery vs frozen or fresh

IC-4 suggests more fresh vs delicatessen or frozen

IC-5 suggests more milk vs frozen or fresh

Given above interpretation of vectors the customers differ by purchases of fresh vs delicatessen, frozen vs detergents and milk vs fresh or frozen. This can be interpreted by the supposition that customers differ in being all types of shops from local supermarket to walmart type, from specialty shop to convenience store.

##Clustering

In this section you will choose either K Means clustering or Gaussian Mixed Models clustering, which implements expectation-maximization. Then you will sample elements from the clusters to understand their significance.

###Choose a Cluster Type

5) What are the advantages of using K Means clustering or Gaussian Mixture Models?

Answer:

K-means is a hard assignment each iteration it's sure to what cluster the data belongs. K-means advantage is that it scales to large number of samples (meaning it is faster), it will always converge but may be to a local minimum (this depends on the initial location of the centroids). The assumption is the clusters are convex and isotropic, so some disadvantages are that it does not respond well to elongated clusters or irregular shapes.

Gaussian mixture is soft assignment, each iteration it assigns probability to each point (how sure it is that its in particular cluster), some advantages are it will not diverge and works with any distribution, and gives more information on structure of the cluster. However it does have some disadvantages, for example it can get stuck and will need to have a random restart.

Gaussian mixture will be selected, as simple cluster shapes (more suitable for K-means) are not expected, clearly defined clusters are not expected, more of a Gaussian distribution is expected, as customers can be expected to be spanning a relatively large distribution with no clear class association. A probabilistic model is more suitable.

There is no clear way to tell the number of clusters in the dataset, so different cluster numbers will be tried and adjusted taking the visualizations into consideration.

6) Below is some starter code to help you visualize some cluster data. The visualization is based on this demo (http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html) from the sklearn documentation.

In [202]:
```
# Import clustering modules
from sklearn.cluster import KMeans
from sklearn.mixture import GMM
```

In [218]:
```
# TODO: First we reduce the data to two dimensions using PCA to captur
pca = PCA(n_components=2)

reduced_data = pca.fit_transform(data)

#test
#ica = FastICA()
#reduced_data = ica.fit_transform(reduced_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]]
```

In [219]:
```
# TODO: Implement your clustering algorithm here, and fit it to the re
# The visualizer below assumes your clustering object is named 'cluste
#clusters = KMeans(n_clusters=5)

#scaling data test
#scaler = StandardScaler()
#reduced_data = scaler.fit_transform(reduced_data)

#Range of clusters numbers ran (2-6)
clusters = GMM(n_components=5, random_state=42)
clusters.fit(reduced_data)

print clusters
```

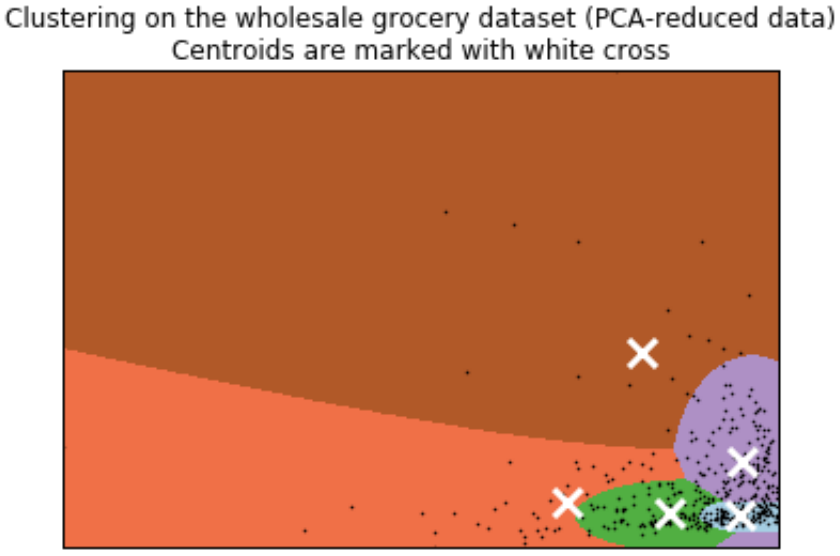
```
GMM(covariance_type='diag', init_params='wmc', min_covar=0.001,
  n_components=5, n_init=1, n_iter=100, params='wmc', random_state=4
2,
  thresh=None, tol=0.001, verbose=0)
```

In [220]:
```
# Plot the decision boundary by building a mesh grid to populate a gra
x_min, x_max = reduced_data[:, 0].min() - 1, reduced_data[:, 0].max()
y_min, y_max = reduced_data[:, 1].min() - 1, reduced_data[:, 1].max()
hx = (x_max-x_min)/1000.
hy = (y_max-y_min)/1000.
xx, yy = np.meshgrid(np.arange(x_min, x_max, hx), np.arange(y_min, y_m

# Obtain labels for each point in mesh. Use last trained model.
Z = clusters.predict(np.c_[xx.ravel(), yy.ravel()])
```

In [221]:
```
# TODO: Find the centroids for KMeans or the cluster means for GMM

centroids = clusters.means_
print centroids
```

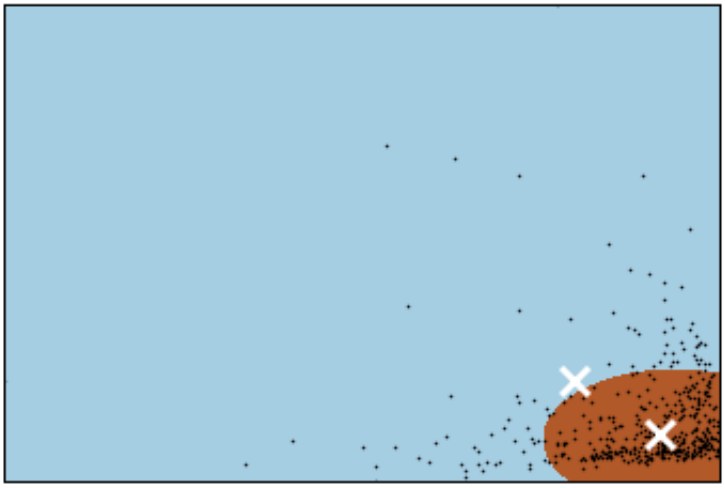
```
[[  6696.36676866  -6467.64552512]
 [ -4647.84385127  -5938.71387776]
 [-21594.97891001  -3437.82018436]
 [  7189.91286381   6031.87459686]
 [ -9453.97791748  32203.92662496]]
```

In [222]:
```
# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(1)
plt.clf()
plt.imshow(Z, interpolation='nearest',
           extent=(xx.min(), xx.max(), yy.min(), yy.max()),
           cmap=plt.cm.Paired,
           aspect='auto', origin='lower')

plt.plot(reduced_data[:, 0], reduced_data[:, 1], 'k.', markersize=2)
plt.scatter(centroids[:, 0], centroids[:, 1],
            marker='x', s=169, linewidths=3,
            color='w', zorder=10)
plt.title('Clustering on the wholesale grocery dataset (PCA-reduced da
          'Centroids are marked with white cross')
plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.xticks(())
plt.yticks(())
plt.show()
```

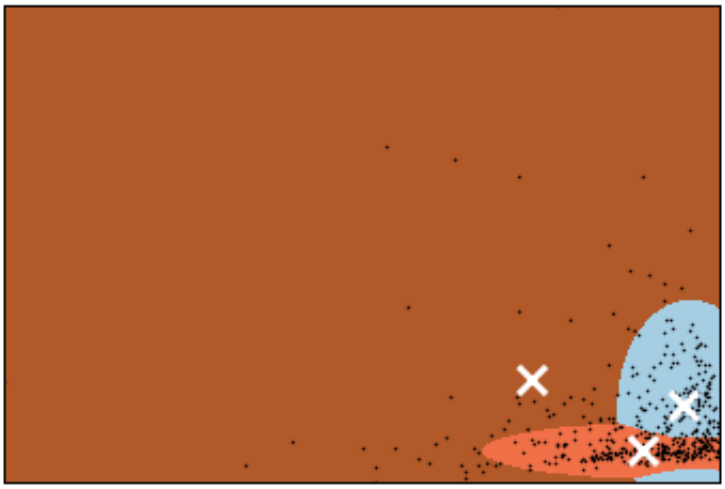


In [160]:
```
def GMMTests(reduced_data = 0, n_cluster = 1):
    clusters = GMM(n_components=n_cluster, random_state=42)
    clusters.fit(reduced_data)

    # Plot the decision boundary by building a mesh grid to populate a
    x_min, x_max = reduced_data[:, 0].min() - 1, reduced_data[:, 0].ma
    y_min, y_max = reduced_data[:, 1].min() - 1, reduced_data[:, 1].ma
    hx = (x_max-x_min)/1000.
    hy = (y_max-y_min)/1000.
    xx, yy = np.meshgrid(np.arange(x_min, x_max, hx), np.arange(y_min,

    # Obtain labels for each point in mesh. Use last trained model.
    Z = clusters.predict(np.c_[xx.ravel(), yy.ravel()])

    centroids = clusters.means_

    # Put the result into a color plot
    Z = Z.reshape(xx.shape)
    plt.figure(1)
    plt.clf()
    plt.imshow(Z, interpolation='nearest',
               extent=(xx.min(), xx.max(), yy.min(), yy.max()),
               cmap=plt.cm.Paired,
               aspect='auto', origin='lower')

    plt.plot(reduced_data[:, 0], reduced_data[:, 1], 'k.', markersize=
    plt.scatter(centroids[:, 0], centroids[:, 1],
                marker='x', s=169, linewidths=3,
                color='w', zorder=10)
    plt.title('Clustering on the wholesale grocery dataset (PCA-reduce
              'Centroids are marked with white cross, cluster number:
    plt.xlim(x_min, x_max)
    plt.ylim(y_min, y_max)
    plt.xticks(())
    plt.yticks(())
    plt.show()

#trying different cluster number
GMMTests(reduced_data, 2)
GMMTests(reduced_data, 3)
GMMTests(reduced_data, 4)
GMMTests(reduced_data, 5)
GMMTests(reduced_data, 6)
```

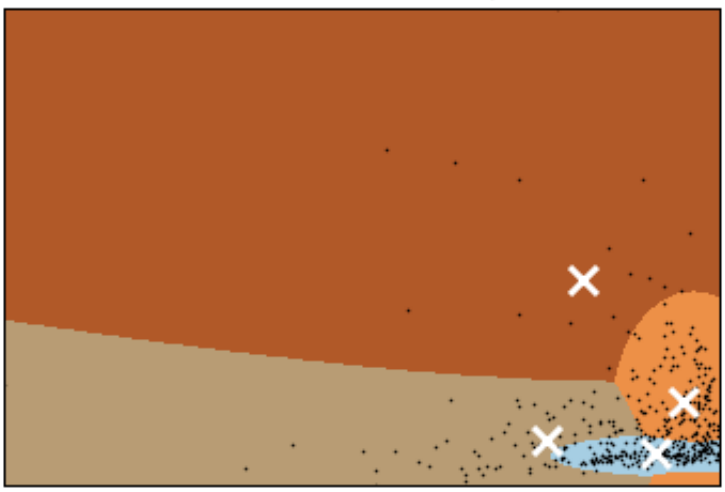
Clustering on the wholesale grocery dataset (PCA-reduced data)
Centroids are marked with white cross, cluster number: 2

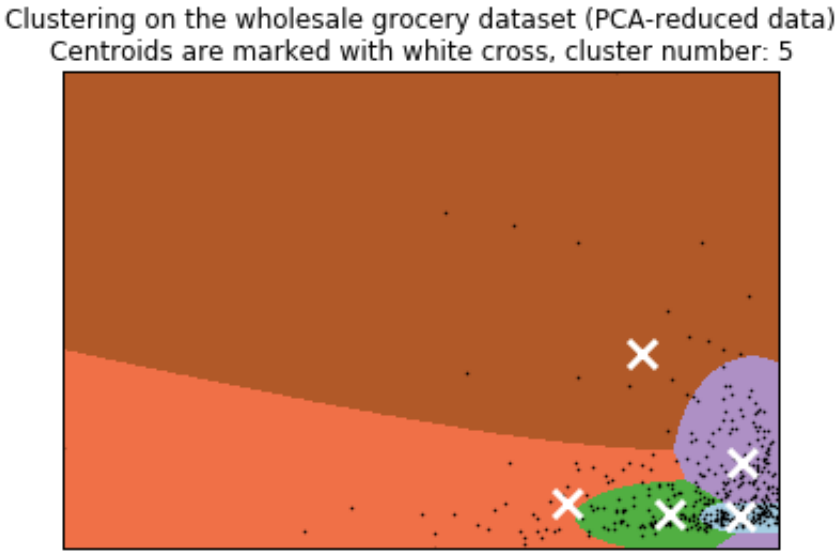


Clustering on the wholesale grocery dataset (PCA-reduced data)
Centroids are marked with white cross, cluster number: 3



Clustering on the wholesale grocery dataset (PCA-reduced data)
Centroids are marked with white cross, cluster number: 4





7) What are the central objects in each cluster? Describe them as customers.

Answer:

Finally we get a plot of points (customers) with identified clusters (customer types), the plot is graphed in terms of 2 dimensions (one dimension “X” - fresh with some frozen and milk, and others to lesser extent and one dimension “Y” - detergents_paper with some delicatessen and frozen, and others to lesser extent) provided by PCA. Five clusters seem to make sense for this dataset, the “x” in each cluster represents the mean of the cluster or the average customer for each cluster. There are no extremely defined regular clusters in the visualization, yet I would argue there are three relatively defined clusters (customers), the most right and bottom (in light blue) one (customer type 1), the most bottom center horizontally elongated (horizontally variad, in green) one (customer type 2) and the most to the right center vertically varied (in purple) one (customer type 3). The other two clusters are almost outliers, yet they represent the largest customers on both sides, this is why the second pair of clusters (customer types 4 & 5) also need to be included as classes. In the end it looks like there are four major types of customers clustered by two parameters sales of food and nonfood items:

    X - fresh with some frozen and milk, and others to lesser extent
    Y - detergents_paper with some delicatessen and frozen, and others to lesser extent

    Customer Types:

    1 - low volumes of X and Y (light blue)
    2 - low volumes of X and average volumes of Y (green)
    3 - low volumes of Y and average volumes of X (purple)
    4 - high volumes of X and low volumes of Y (brown)
    5 - high volumes of Y and low volumes of X (dark pink)

Additionally looking at the graph, a few points that fall into customer type 4 & 5 are high volume X and Y, one can make a good argument to put them into customer type 6, however running GMM with 6 clusters does not provide this result (as there are not enough type 6 customers to make a cluster). Nevertheless there is a good argument for the warehouse company to have this customer type.

Running ICA on the PCA output (and then using the ICA output with Gaussian mixture) might make the clusters more pronounced/defined. ICA should be able to remove some of the noise/mix in the data. As well as running a scalar function before plotting the graph.

###Conclusions

8) Which of these techniques did you feel gave you the most insight into the data?

```
Answer:

Two techniques have made this model possible PCA and Gaussian
mixture.

PCA identified the number of dimensions most responsible for the data
variance by running PCA in the same dimension as the dataset, PCA
identified and ordered the dimensions by maximum variance ratio,
where the variance ratio was used in the next step of PCA (reduction
of dimensions). In this second step, PCA reduced the number of
dimensions to those that most explain the maximum variance (two
dimensions in this case). It was important to reduce the number of
dimensions (for visualization purposes as well as reducing the
running time of GMM) while keeping most of the data.

Gaussian mixture using the reduced dimension dataset identified
clusters/customers by using probabilistic assignment of points to a
given cluster (soft assignment). The clusters then can be visualized
for visual inspection of the dataset and its clusters/customer types.

Additionally it was invaluable to identify the final cluster number
by visually inspecting the graph with initial test/guess cluster
number.
```

9) How would you use that technique to help the company design new experiments?

Answer:

We can use the same techniques to identify and categorize customers using other data to find other customer types, for example customer types based on payment preference correlated to types of orders or other customer data.

If another experiment was conducted and certain customer preferences identified we can run similar customer models before implementing particular measures to make sure the measures proposed will not negatively impact a customer subset, or to positively impact particular customer subset.

In particular we can do A/B testing. We can split the customers into groups (we can randomly allocate customers into each group), perform our experiments on one group to see how the customers will react to particular change and then after we get customer reactions extrapolate the reactions to the second group (using the model we have created in this report, or by running similar models if our parameters change).

10) How would you use that data to help you predict future customer needs?

Answer:

By identifying the classes/types of customers (as we have done above) and knowing each of the existing customers preference on delivery we can match class of customer to delivery preference. Therefore we can predict delivery preferences of future customers by the types of orders those customers have a preference for or have placed. In addition we can predict the customer orders by knowing the delivery preference. We can use supervised learning to predict delivery preferences of new customers by training naive bayes or SVM or some other classifier with the original data with added labels for each customer (as we have classified here using PCA and GMM), in this case labels can be type 1 to 6 customer we have talked about in section 7. After the classifier is fitted with this data, we can predict with new data from new customers, what type of customer the new client is.

Note: there might be other preferences that can be predicted if we can correlate them with the order types or vice versa.