# Customer Segmentation Using Gaussian Mixture Models and Expectation-Maximization (EM) Algorithm

Summary: IBM has the capability in machine learning and data science to solve challenges in the business world. In this paper, we will explore one instance where IBM practitioner performed customer segmentation with synthetic product offering marketing data using machine learning algorithm Gaussian Mixture Models and Expectation-maximization. The algorithm was able to create customer cohorts and uncover hidden insights for each of the groups. The additional insights will add business value and aid business’ ongoing marketing efforts to attract potential new customers.

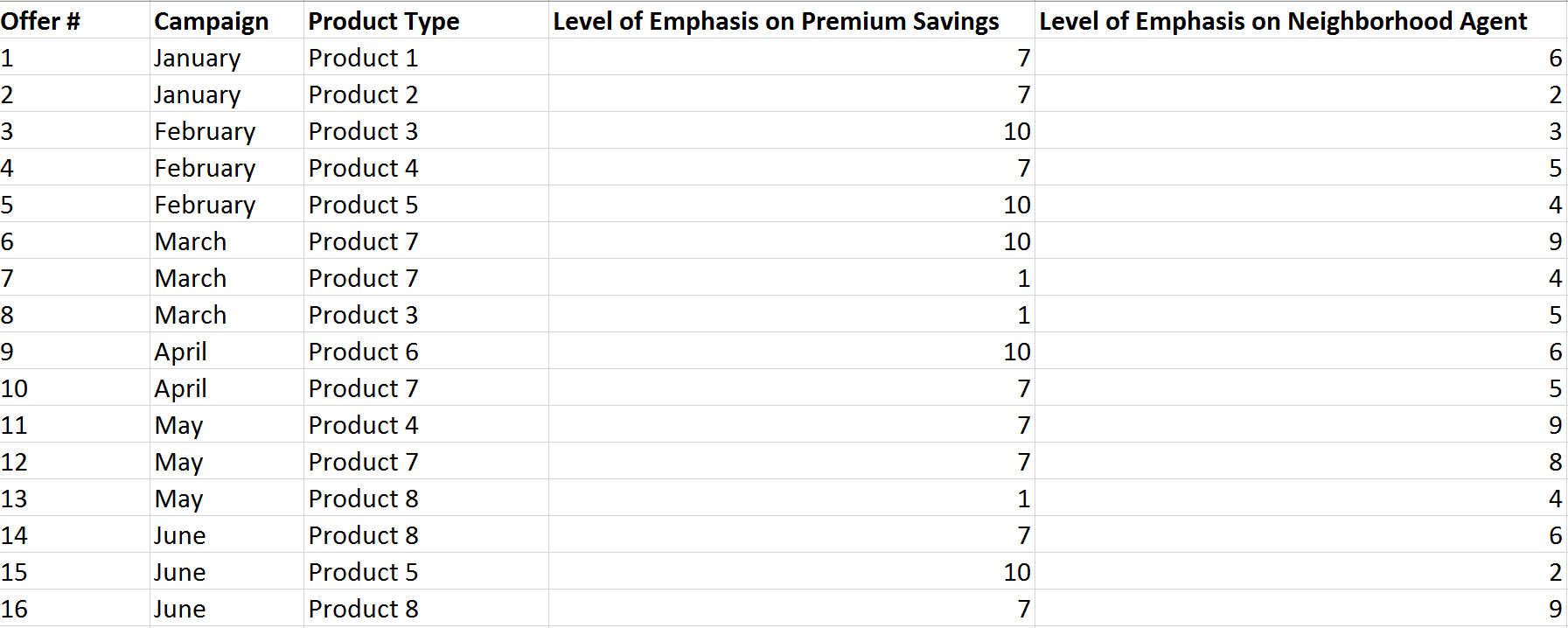
## Business Problem

Customer segmentation is a common practice used to divide customer base into groups of individuals that are similar in specific ways. It is a helpful approach in marketing to provide additional customer insights.

To simulate a real-world business problem, we generated a synthetic dataset that contains both information on marketing materials sent by an insurance company for different product offers as well customer responses or inquires on these offers. Our goal is to group customers into different segments, and analyze their similarities and differences to uncover hidden insights. We believe a better understanding of customer segments will help insurance companies and financial institutions in general customize their product offerings and marketing approaches to potential new customers.

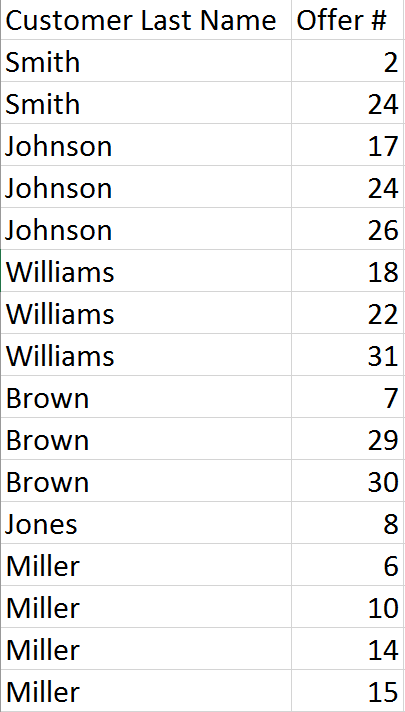
## Type of Data

For our version of the algorithm implementation, we will need numeric data type for input. Below is a screenshot of the synthetic sample dataset for Insurance product offering:



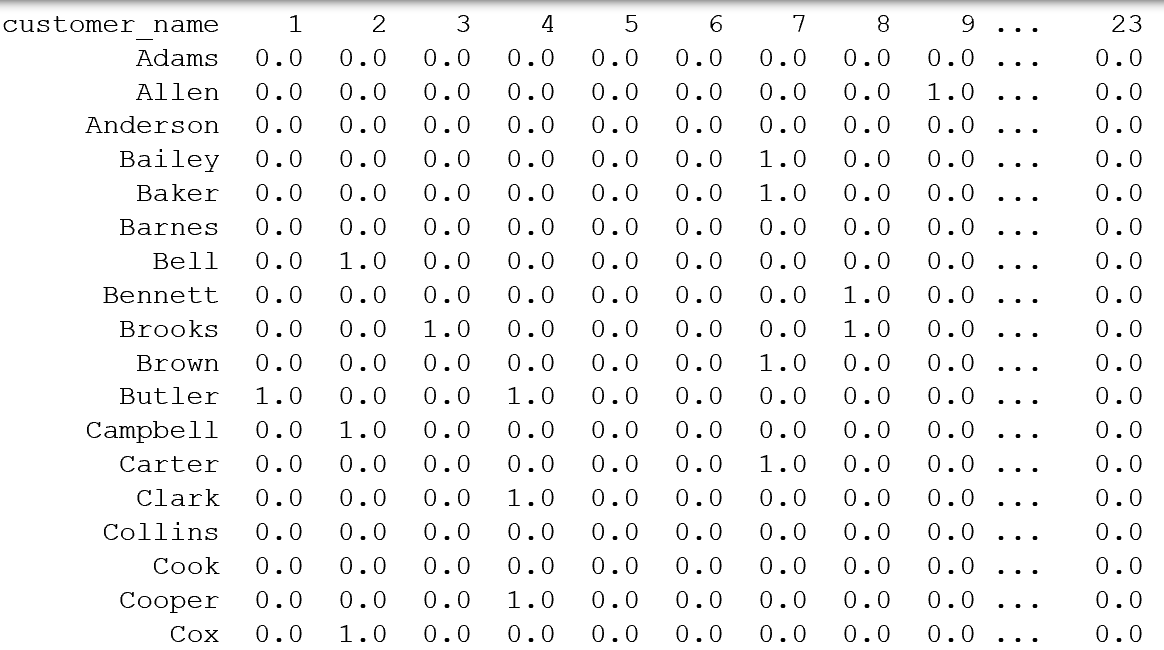
The Offer # column holds the unique identifier for each product offer. Campaign represents the month of the year the offer was sent out to potential customers. Product Type holds nine different products. In addition, we have the columns Level of Emphasis on Premium Savings and Level of Emphasis on Neighborhood Agent. Premium savings and accessibility to a neighborhood agent are two important attributes customers consider when purchasing insurance. Some customer may value one significantly over the other, while others may care about attributes other than these two. Level of Emphasis on Premium Savings is a rating of 1 to 10. It shows how strongly a particular product offer campaign emphasized on the aspect of premium savings, 1 being minimal emphasis and 10 being strong emphasis. Similarly, Level of Emphasis on Neighborhood Agent is also a rating of 1 to 10. It represents how strongly a certain product offer campaign emphasized on the aspect that there are neighborhood agents near the customer. In our dataset, we can see that a product type may be marketed in several different offers in different months. Each offer has different level of emphasis on premium savings or neighborhood agent to cater to various customer preferences.

We also have the customer response/inquiries data. If a customer responded or inquired about a product offering, we consider this customer interested in the offer.



Based on customer response data, we will segment them into groups and derive additional insights.

In the data preparation stage, we loaded the two files into iPython Notebook, merged and pivoted the tables into the format below:



Each row corresponds to a customer, and each column represents a product offer id. For example, customer Allen responded to Product Offer #9, so the second row for Allen and column 9 has the flag 1, while the remaining entries are 0 representing that the customer did not express an interest in those product offers.

Once we have the data in the matrix format, we are ready to pass it on to our algorithm written in Python. We can see that many real-world problems, from customer transaction data, to facial recognition, can all be presented in a similar matrix format. Therefore, our Gaussian Mixture Models with Expectation-Maximization algorithm has the capability to solve those challenges as well.

## Technical Background

In this section, we will dive deeper into the algorithm to see how mathematics and machine learning played a role behind the scene in constructing the model logic.

The machine learning algorithms we are using for this paper are Gaussian Mixture Models (GMM) with Expectation-Maximization (EM) algorithm. We will go over the steps of EM algorithm and the logic behind GMM to provide a better understanding on how our algorithm can solve the customer segmentation challenge.

### Expectation Maximization (EM) Algorithm

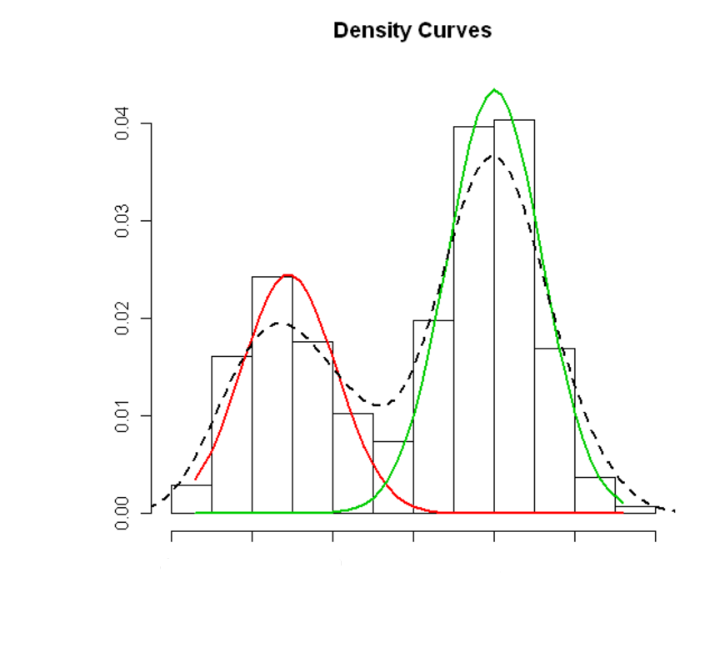
The EM algorithm is an iterative method that enables maximum likelihood estimation of the parameters in probabilistic models with incomplete data. It can be applied to Hidden Markov Models, Bayesian Networks, Gaussian Mixture Models and so on to estimate parameters in the absence of certain information within the data.

Every model has certain parameters. When we do not have the parameters readily available, we can estimate them in the following steps of the EM algorithm:

1. Estimate the parameters with our best initial guess.
2. In Expectation step (E-step), we find the probability distribution using the current parameters.
3. In the Maximization step (M-step), new parameters are determined based on what we have from Step 2.
4. After several repetitions of the E-step and M-step, the algorithm converges.

### Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMM) is a mixture of several Gaussian distributions. Individual distributions (referred to as mixture components) have different means and variances, and different mixture weights. The final distribution for our data is obtained by multiplying each mixture component by its associated mixture weight before adding them together (mixture weights must sum to one).

To make GMM is a little more intuitive, let’s look at a simple example. Imagine if you have the measurement of heights for 100 adults, both male and female, but the label for gender is lost. We don’t know if a data point is from the male population or the female population. When we plot the distribution of heights, we see two clear peaks - most likely corresponding to a peak in male heights and a peak in female heights. Unfortunately, it is difficult to see from the messy data what the average height of men and the average height of women would be. It is even more difficult to say if a particular observation was a man or a woman.

Density

Height

GMM assumes that the observed data is made up of a mixture of several Gaussian distributions. In this example, we would want two Gaussian distributions - one for men and one for women - with different means and perhaps different variances, and the mixture weights correspond to the probability of a random individual being male or female (roughly 0.5 in each case).

Now let’s translate this simple example into the customer segmentation scenario. In customer segmentation, we do not know which segment a customer belongs to, much like we do not know the gender of an individual in our population height example. In our solution, we will assume each of our customer segments is a Gaussian distribution with its own set of parameters.

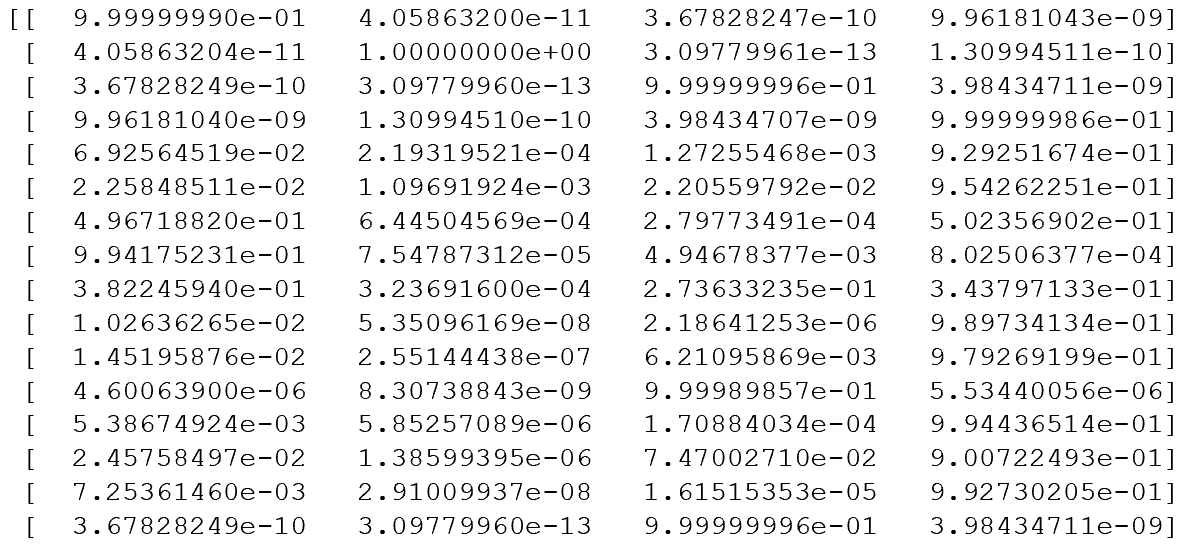
### Gaussian Mixture Models(GMM) using Expectation-Maximization(EM) algorithm

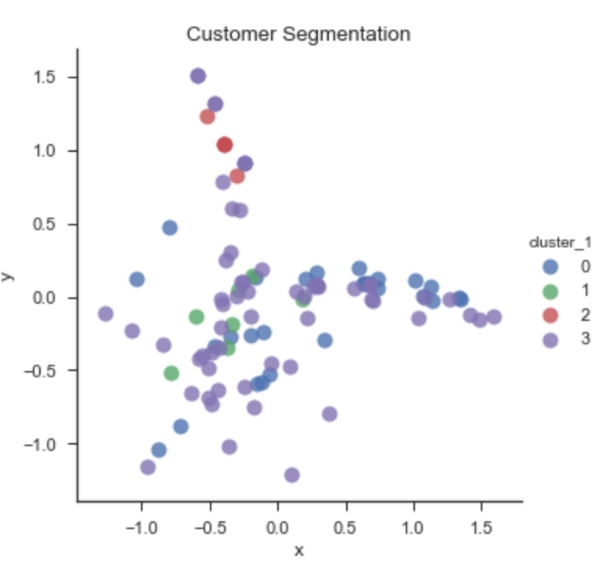
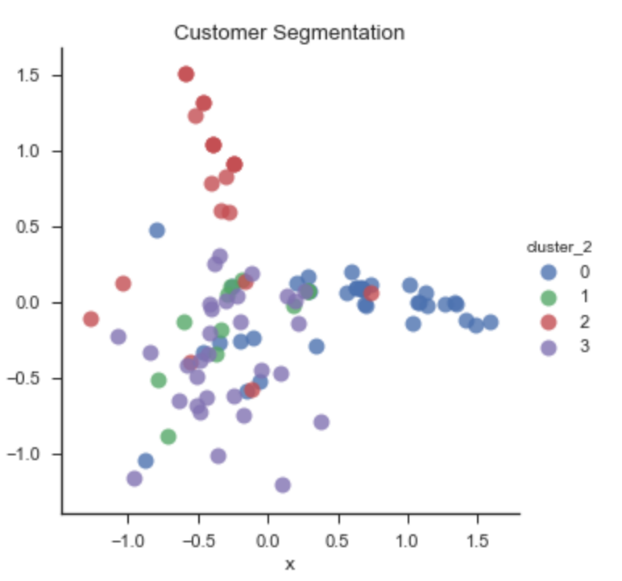
Let’s bring GMM and EM together in this section to see how they can solve our challenges. We will start with the slightly more intuitive population height example.

1. Estimate the parameters with our best initial guesses for the Gaussian distribution of female heights, the Gaussian distribution of male heights, and the likelihood of a data point being from one of the two gender populations.
2. In Expectation step (E-step), we find the probability distribution of each data point belonging to each gender using the current parameters.
3. In the Maximization step (M-step), new parameters for the Gaussian distribution of female heights, the Gaussian distribution of male heights, and the likelihood of a data point being from one of the two gender populations are determined based on what we have from Step 2.
4. After several repetitions of the E-step and M-step, the algorithm converges.

IBM practitioner has written the Python code that will perform the above algorithms automatically. Let’s visualize the result of the code with the synthetic insurance data we generated during the section Type of Data.

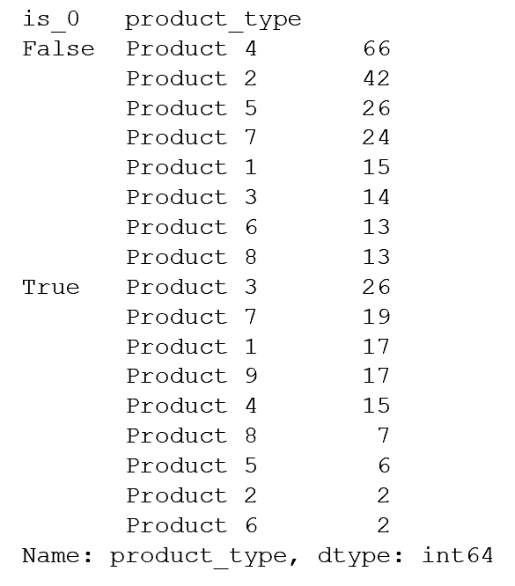
Once we feed the insurance product offering marketing data to the model, after the first iteration of E-step and M-step, we have a matrix of probabilities as the output, where each row represents a unique customer, and each column represents a cluster. Each entry within the matrix is a probability that represents the likelihood of this customer belonging to the specific cluster. For our example, we elected to have four clusters, and hence we have four columns, Cluster 0, Cluster 1, Cluster 2 and Cluster 3. Each row is a unique customer.

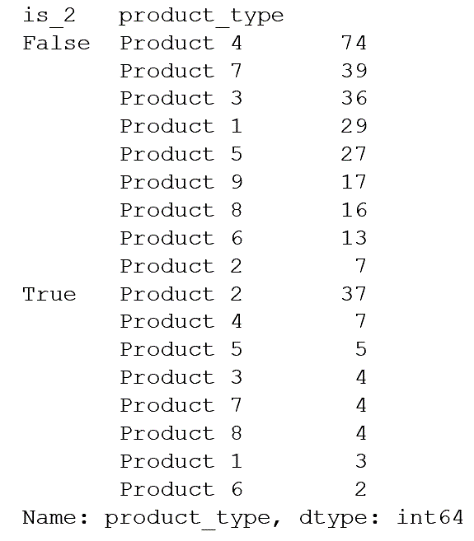


To provide a better visualization, I assigned each customer to the cluster that he/she has the highest probability of being a member (row max), and plotted the customers and their cluster assignment. Keep in mind that our data is high-dimensional. For purpose of creating this graph, we did principle component analysis which is a popular dimension reduction algorithm, and reduced our data points (the customers) to 2-dimensional space. We can see that during the first iteration, most customers were assigned to Cluster 3, and very few were assigned Cluster 2.

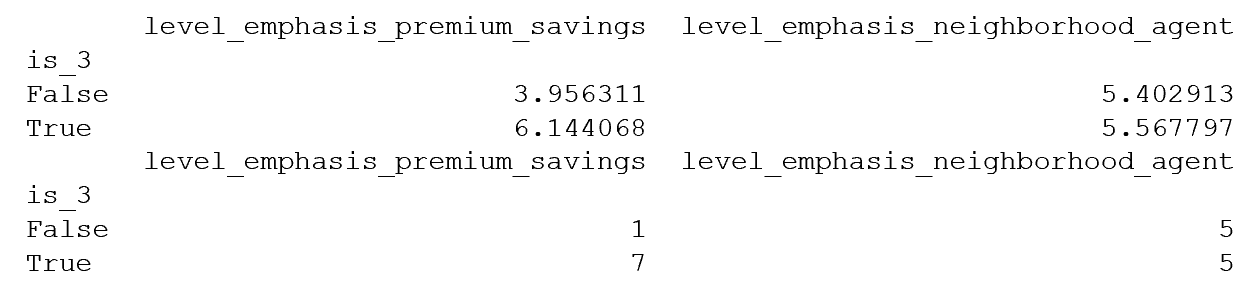
As the EM algorithm suggests, we repeat the E-step and M-step for GMM iteratively until the algorithm converges. So we performed EM many times on the data. After the final iteration, we once again have the probabilities of each data point belonging to each cluster. We again assigned the customers to the clusters with the highest probabilities, and plotted the cluster assignment again. We can see that the number of customers belonging to Cluster 3 decreased, and customers that are ‘close’ to each other in the dimensional space are assigned to the same cluster, as the same colors now appear in a much more concentrated manner. We still see data points, or customers, with different cluster assignments overlapping each other. However, the data points have been reduced to a 2-dimensional space to enable data visualization. In reality, points that seemingly overlap each other in a 2-dimensional space may be far apart once we bring back the other dimensions. After evaluating our final result, we consider our GMM with EM algorithm a success in clustering the customers into different segments.

## Business Values Added

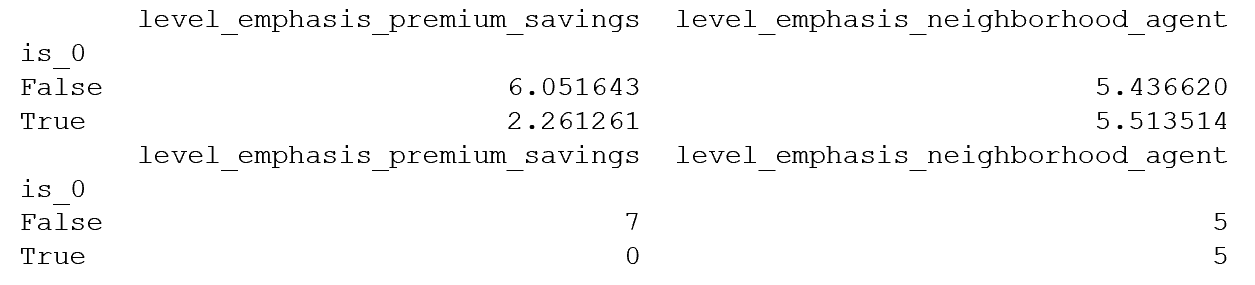
Customer segmentation can help uncover hidden insights and identify future marketing opportunities. Based on the analysis we performed in the last section, we have discovered some interesting trends and patterns that may create additional values for businesses. For example, after grouping the customers for Cluster 0 together and compare with the rest of the customers, we realized that Cluster 0 customers are the only ones that have expressed interest in Product 9. In addition, contrary to the rest of the customers, Cluster 0 customers do not have as much interested in Product 2, so perhaps the marketing team for the company can target this group for offers for Product 9, and shift away from Product 2.

On the other hand, Cluster 2 customers are the ones marketing should focus on for Product 2 per this analysis, as 37 customers from Cluster 2 expressed interest in Product 2, and only 7 did so from other clusters.

In our original dataset, we also have information on the level of emphasis each marketing offer placed on two attributes that are commonly of interest to the insurance customers: premium savings and the accessibility to neighborhood agents. Some customers care more about one aspect over the other, while others may care about both equally or neither. With customer segmentation, we can also identify these characteristics for each customer segment, or cluster, and thus customize unique offers that emphasize on the aspect of the product that customers truly care about, and funnel marketing expenses to deliver the right message. For instance, after running the mean and median of the offers that Cluster 3 expressed interest on, we realized these customers care greatly about saving money in insurance.



Meanwhile, premium savings does not seem to be as important for customers in Cluster 0, so perhaps there are some other aspects of the product offerings that can stimulate these customers’ interest.



In conclusion, we have demonstrated IBM’s analytical and machine learning capabilities using customer segmentation as an example. We successfully clustered customers into groups using only information from what offers the customers expressed interests in. We identified different product types we can focus on for different customer groups, and uncovered aspects of the marketing campaign we should emphasize on for each group. We can conduct even more supplementary analysis on the customer segments if we have information on customer age, income level and so on to discover additional hidden insights.

Our algorithm can also be applied to solve other real-world challenges such as anomaly detection or facial recognition. For instance, we used the same Python code to perform facial recognition for images of famous politicians. The applications of Gaussian Mixture Models with Expectation-Maximization algorithm are endless, and we can apply it in many more machine learning and data analytics projects to solve clients’ challenges.

## References

<http://blog.yhat.com/posts/customer-segmentation-using-python.html>

<http://ieeexplore.ieee.org/document/1633722/?reload=true>

<http://www.nature.com/nbt/journal/v26/n8/full/nbt1406.html#f1>

<https://www.researchgate.net/post/How_does_GMM_work_in_principle>

<http://datasciencecourse.org/anomaly_detection.pdf>

<http://cs229.stanford.edu/notes/cs229-notes8.pdf>