Coffee Shop: Advanced customer segmentation with Python

Introduction

Customer Segmentation is the subdivision of a market into discrete customer groups that share similar characteristics. Customer Segmentation can be a powerful means to identify unsatisfied customer needs. Using the above data companies can then outperform the competition by developing uniquely appealing products and services.

The most common ways in which businesses segment their customer base are:

1. **Demographic information**, such as gender, age, familial and marital status, income, education, and occupation.
2. **Geographical information**, which differs depending on the scope of the company. For localized businesses, this info might pertain to specific towns or counties. For larger companies, it might mean a customer's city, state, or even country of residence.
3. **Psychographics**, such as social class, lifestyle, and personality traits.
4. **Behavioral data**, such as spending and consumption habits, product/service usage, and desired benefits.

Advantages of Customer Segmentation

1. Determine appropriate product pricing.
2. Develop customized marketing campaigns.

3. Design an optimal distribution strategy.
4. Choose specific product features for deployment.
5. Prioritize new product development efforts.

Data acquisition and cleaning

The dataset consists of three separate JSON files:

1. **Customer profiles** — their age, gender, income, and date of becoming a member.
2. **Portfolio —** Offers sent during the 30-day test period, via web, email, mobile or social media channels, or a combination thereof. The offers have varying levels of difficulty (minimum spend) and reward, and fall into one of three categories: *Discount*, *Buy-one-get-one (BOGO)*, *Informational*
3. **Transcript** — A list of offer interactions (receive/view/complete), and all other transactions during the test period.

Data wrangling

Before I could visualize and model the data, I've had to do some pre processing both outside, and in Python.

Among others, I have:

1. **Removed empty lines** in transcript.json using search \n\n & replace \n in Visual Studio Code
2. **Imputed** empty income values with the mean ($65,404), and added a separate feature that tracks missing income values with 1s and 0s.
3. **Engineered** a new feature for the year when the user became a member
4. **One-hot-encoded** channels using the *MultiLabelBinarizer*
5. **One-hot-encoded** offer types, genders, years joined and event types using *get_dummies*
6. **Dropped age outliers** (a number of outlier customers had their age set to 118, and were missing data for several of the other fields)
7. **Engineered** first receipt, first view and first completion time features (a customer can receive and interact with the same offer multiple times)
8. **Dropped misattributions** (completion without view, completion before view, or view before receipt)
9. **Calculated RFM** *Recency* and *Frequency* scores— a common method used for analyzing customer value in retail and e-commerce
10. **Engineered** view and conversion rate features for each offer type
11. **Merged all data** into one dataframe grouped by customers, including means and sums for all available data, as well as additional columns for the average number of exposures per offer-type.

Cluster analysis

1. Feature scaling

Both the *PCA* and *k-means* algorithms which we will use below are sensitive to the relative scale of the data.

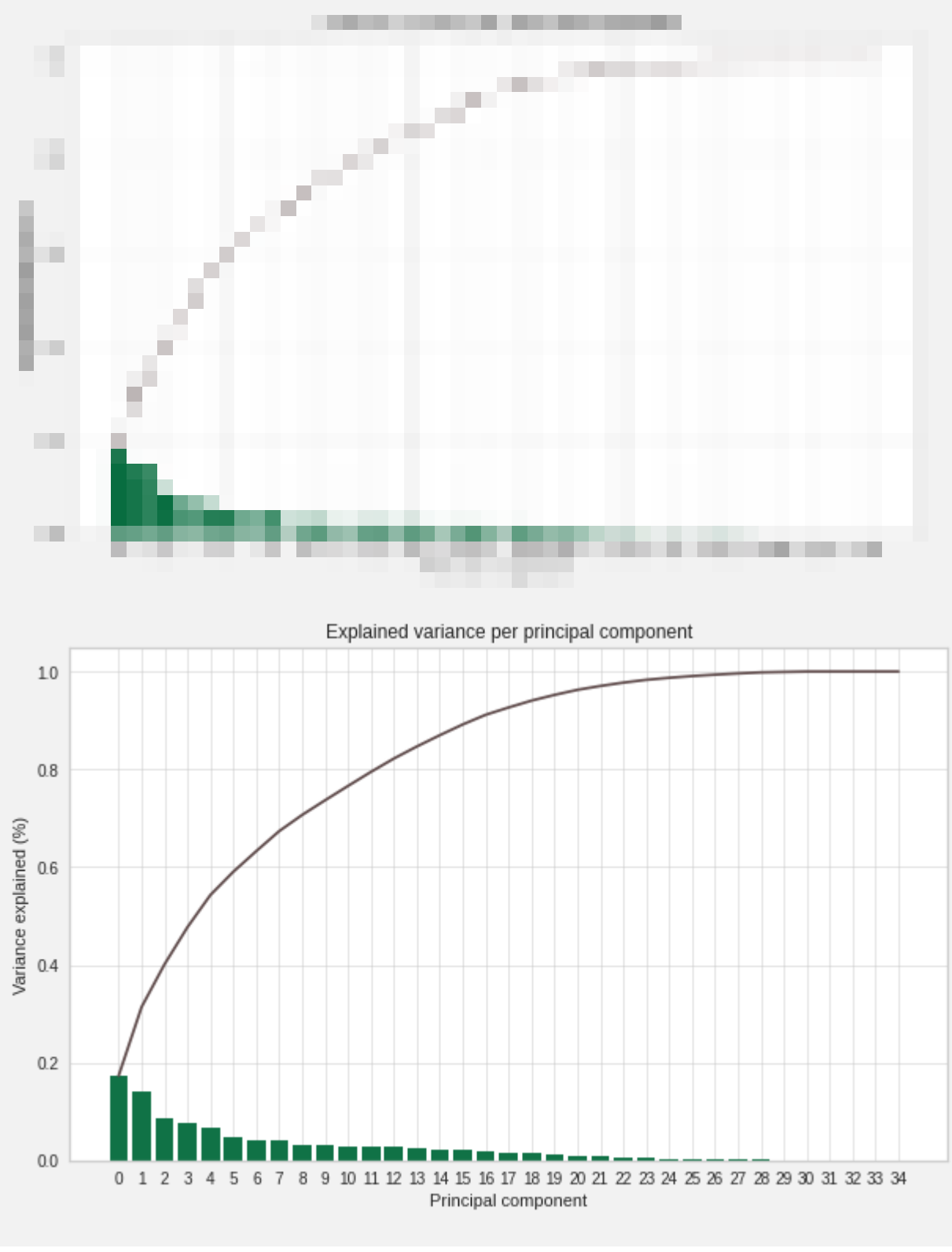
For example, our boolean columns range from 0 to 1, whereas the income column ranges from 30,000 to 120,000, a different magnitude which would negatively affect the clustering.

To solve this problem, I've used *StandardScaler* to transform data such that its distribution will have a mean value of 0, and a standard deviation of 1.

2. Dimensionality reduction

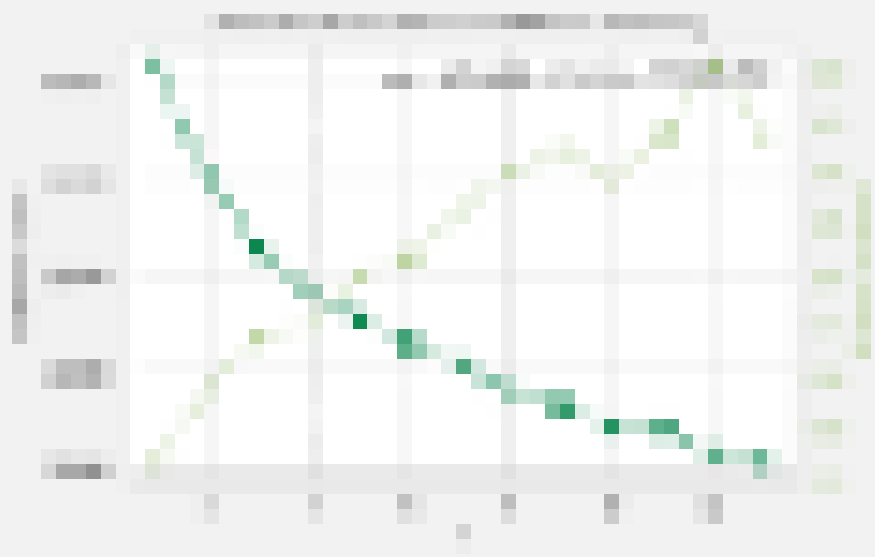
The *k-means* algorithm is both more effective and more efficient with a small number of dimensions, that is, the number of features used to predict the right cluster for each customer.

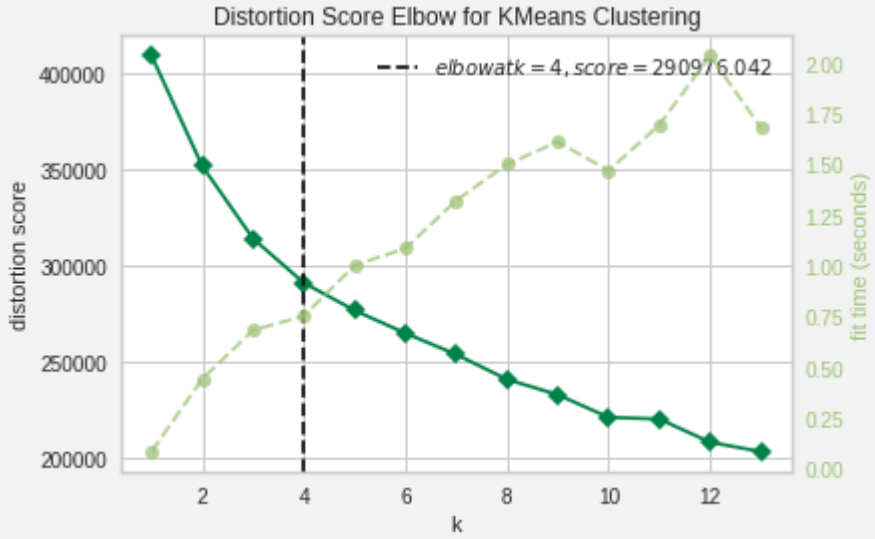
To reduce dimensionality, I've used *Principal Component Analysis (PCA)* — a method which identifies variables that are responsible for most of the variance in the data.

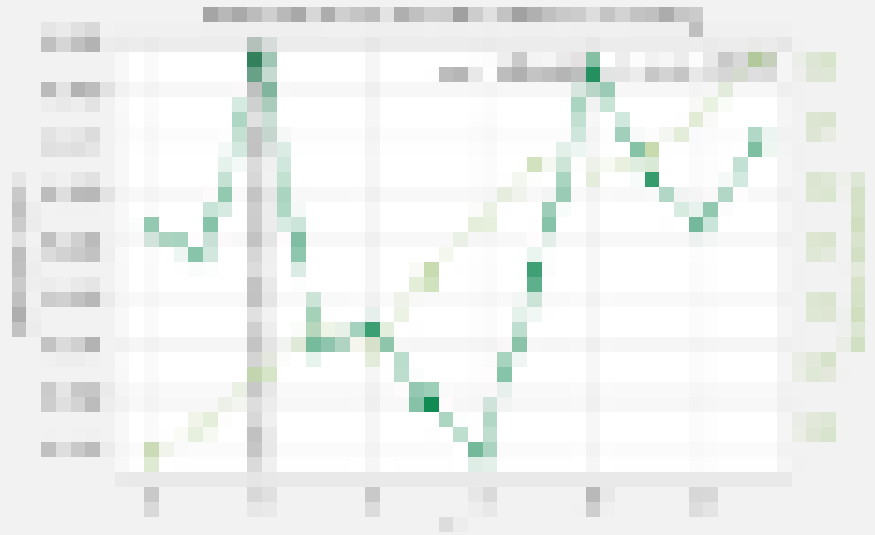


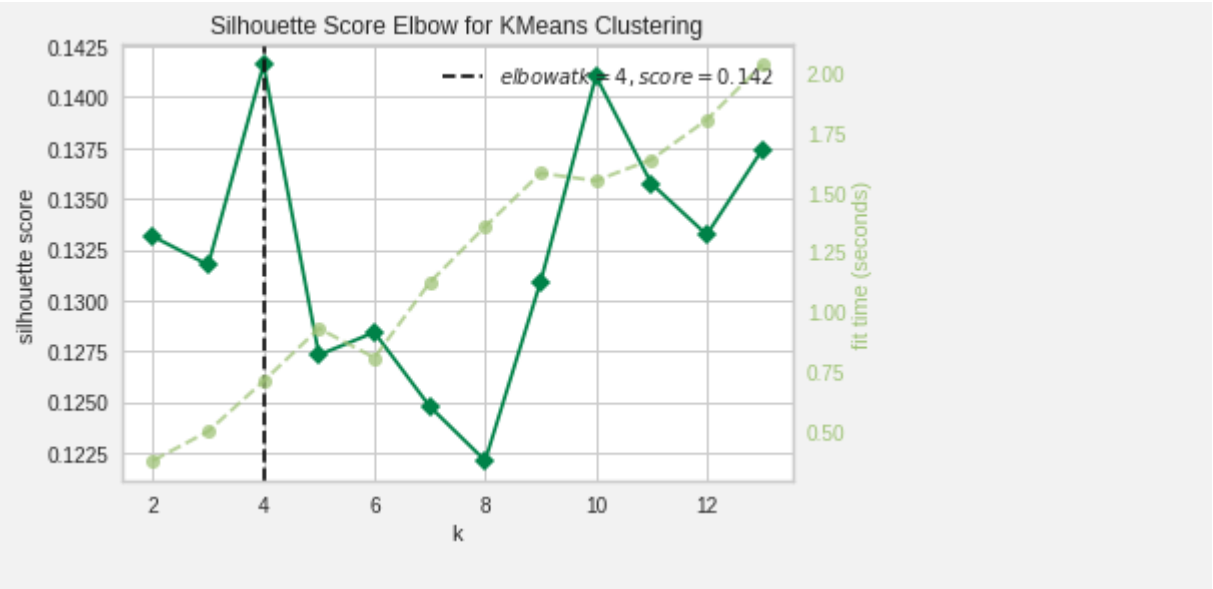
3. Clustering

I have first used the common *k-means* algorithm to classify the data, settling on four clusters based on the *elbow method* and *silhouette score* heuristics.

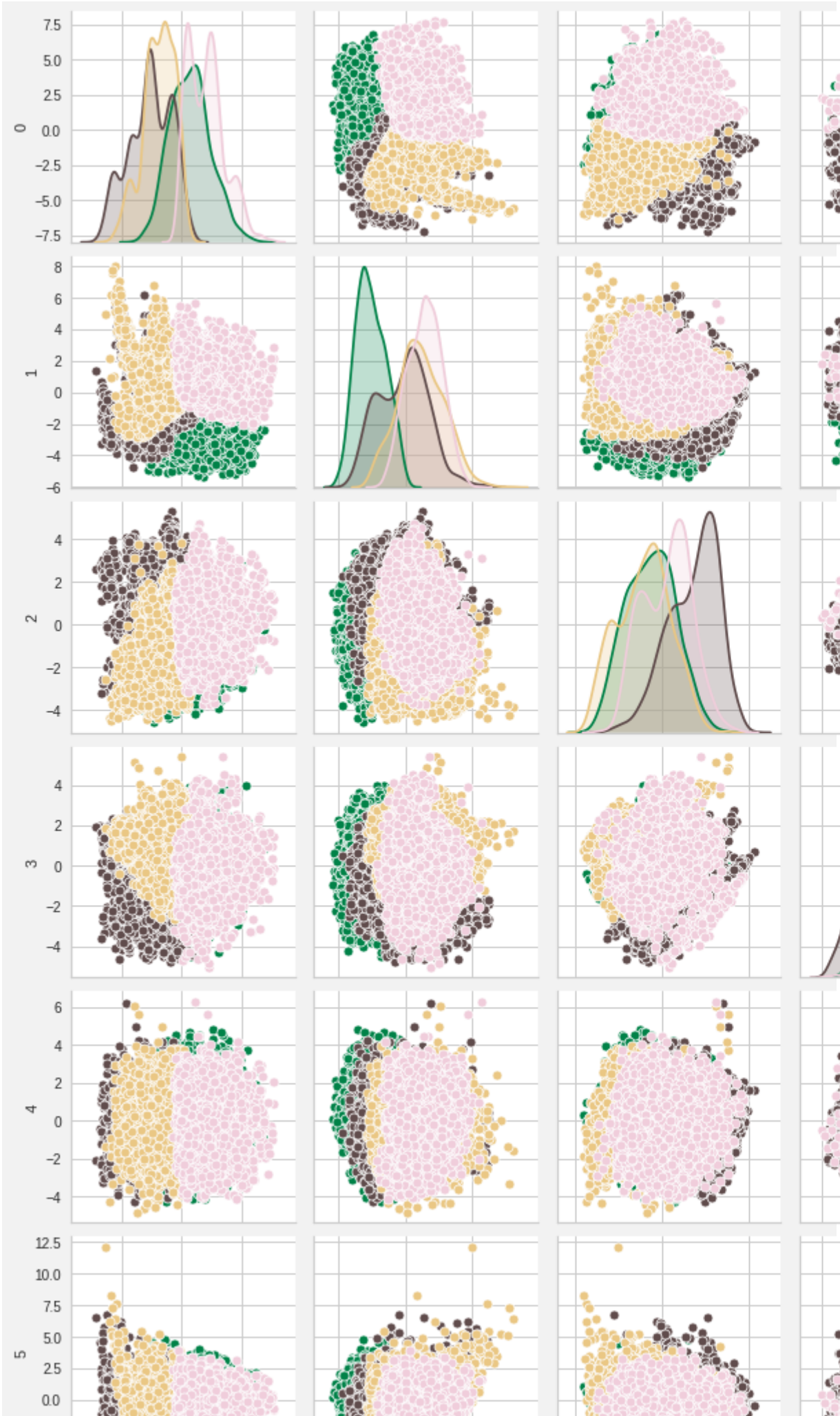



Distortion Score Elbow for KMeans Clustering
elbow at k = 4, score = 290976.042
distortion score
fit time (seconds)
k
400000
350000
300000
250000
200000
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
12



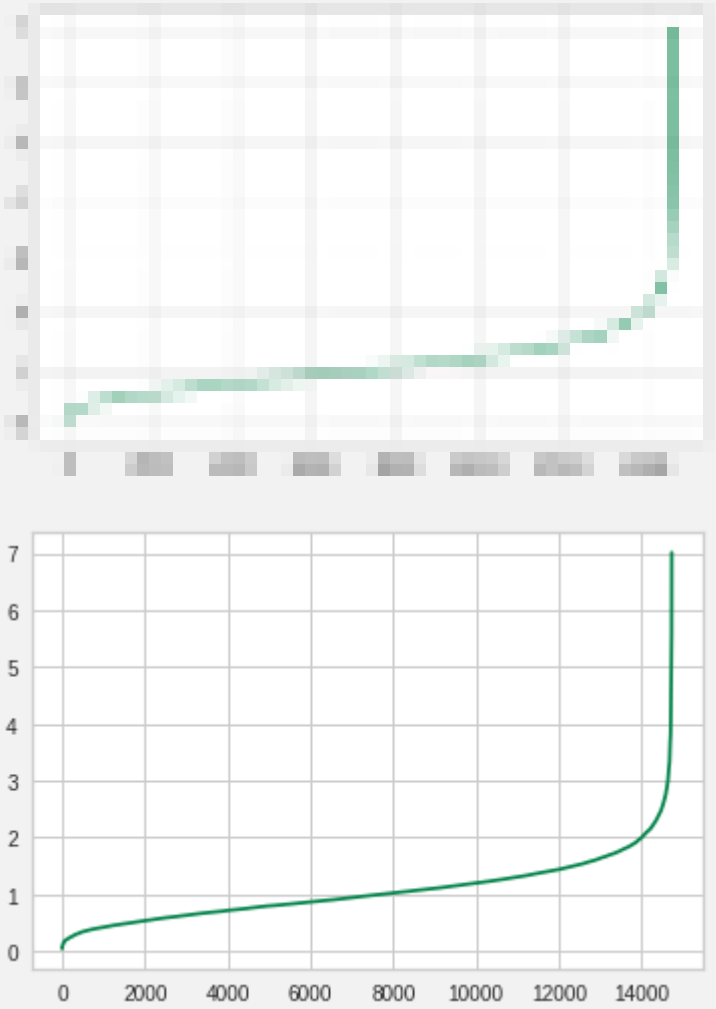


k-means, with n_clusters=4:


0
1
2
3
4
5

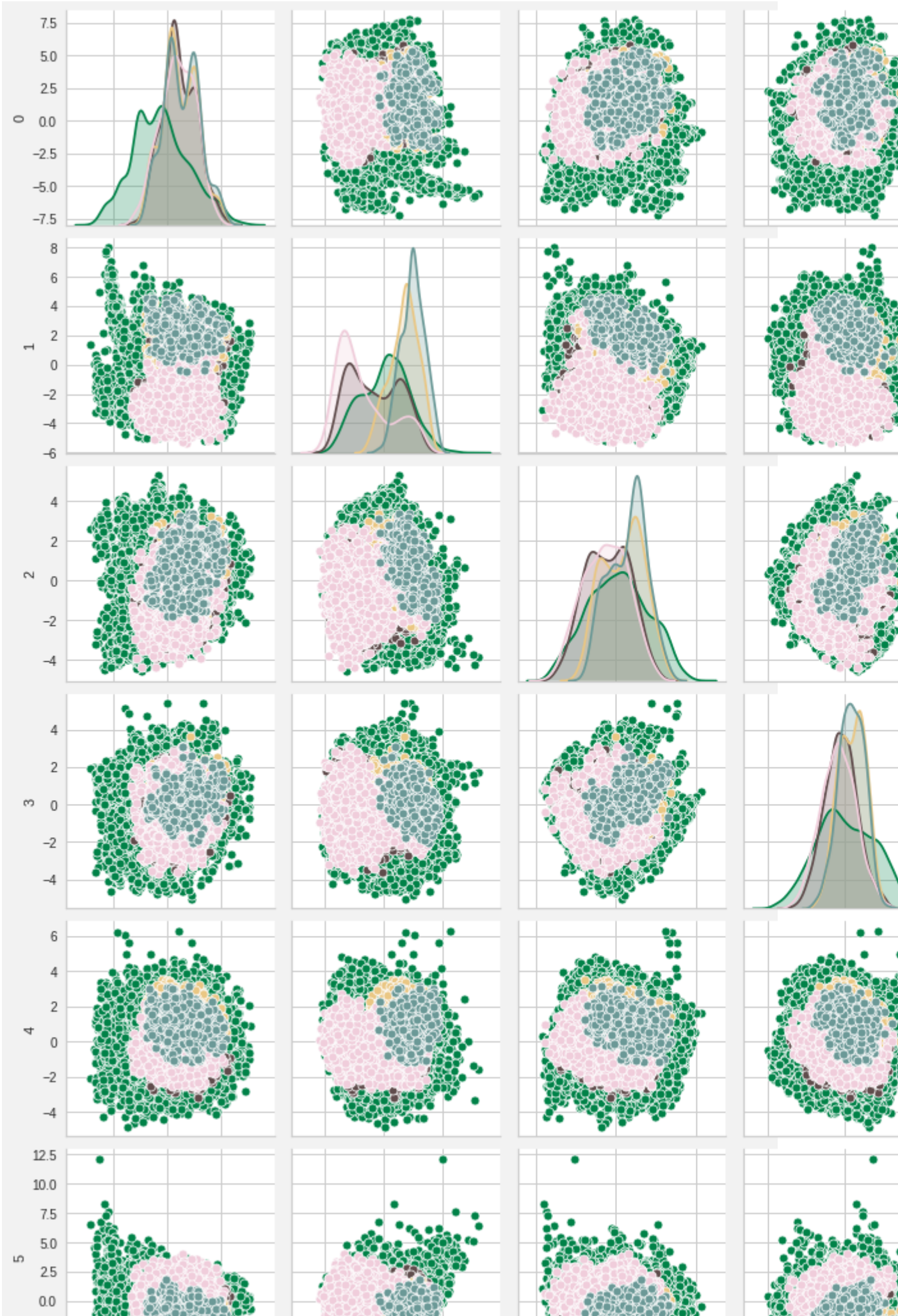
I have then experimented with *DBSCAN* and *OPTICS*, two density-based clustering algorithms.

For *DBSCAN*, I identified 2.5 as the optimal value for the eps parameter, using the elbow method.



For both *DBSCAN* and *OPTICS*, I then experimented with a variety of min_samples values that would generate a reasonable number of well-differentiated clusters.

DBSCAN, with eps=2.5 and min_samples=150:


0
1
2
3
4
5

OPTICS with min_samples=40*:*

Evaluation and validation

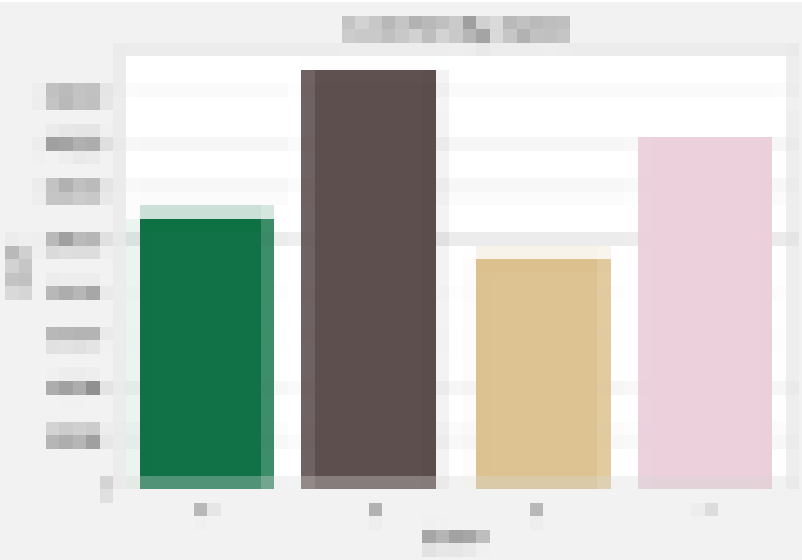
Although the density-based algorithms may appear to perform better in the charts above, the clusters generated using *k-means* show more distinct characteristics that make sense in our business context.

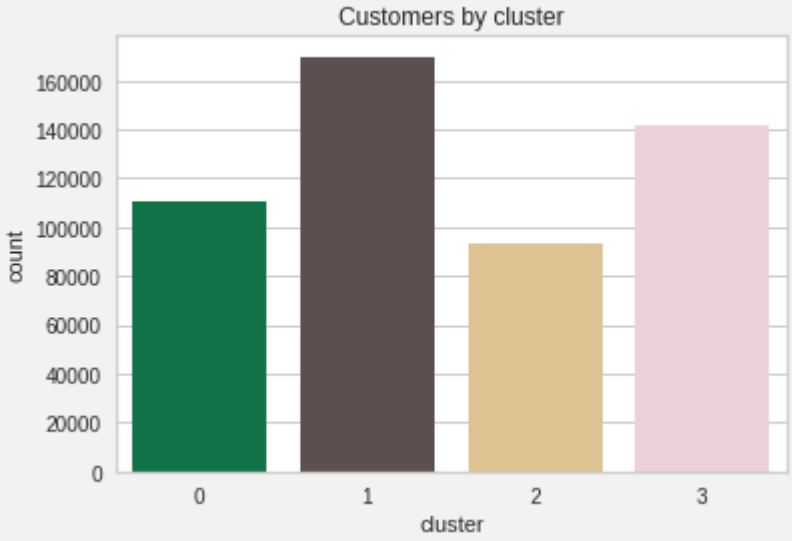
The main problem appears to be that *DBSCAN* and *OPTICS* overemphasize the gender and membership year of the customers, as those variables are more densely clustered.

The resulting data shows minimal variance in our view rate and conversion rate metrics, and is therefore not actionable in our marketing.

OPTICS:

Compare this to the clusters generated using k-means, well differentiated not just in their demographics, but also conversion rates for each individual offer type:




Customers by cluster
160000
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
2
3
cluster


age
bogo_cnt
bogo_cr
bogo_vr
difficulty_avg
discount_cnt
discount_cr
discount_vr
email_cnt
exposures_completed_avg
exposures_received_avg
exposures_viewed_avg
frequency
gender_F
gender_M
gender_O
income
informational_cnt
informational_vr
mobile_cnt
monetary_value
offers_completed_cnt
offers_received_cnt
offers_viewed_cnt
recency
reward_avg
social_cnt
total_cr
web_cnt
year_joined_2013
year_joined_2014
year_joined_2015
year_joined_2016
year_joined_2017
year_joined_2018
value
cluster

Above, we can immediately identify four distinct segments with clear business implications:

Segment 1

Customers in this segment receive regular **BOGO offers**, and practically no discount offers. These BOGO offers involve **more valuable rewards** than for customers in other segments.

Their frequency and average order value are not unusual, which suggests these customers are **conditioned to BOGOs**, and we might have to continue sending them regular offers to keep their patronage.

BOGOs convert really well with customers in this segment, so this is a great lever in times when we need to quickly generate additional sales.

Segment 2

Customers in this segment **receive a higher than average number of offers**, and **convert really well** for both BOGOs and discounts. Demographically, a higher than average share of these customers selected their **gender as *Other***.

Their average order value is not unusual, in line with the average, but their **frequency is above average**, probably as a result of the regular offers they receive and act on.

This is another segment we can target to quickly generate additional sales.

Segment 3

Customers in this segment receive **no BOGO offers**. They do get occasional discount offers, on which they convert about average, as well as slightly more informational messages than other customers.

These customers have about average frequency and average order value, and **would likely continue to frequent Starbucks even if we stopped sending them offers**.

Segment 4

Customers in this segment receive **regular offers**, which they **open, but never convert**.

Demographically, they are predominantly male, and **lower than average income**. They also visit Starbucks less frequently, and make smaller average purchases.

Given the low LTV and low conversion rates for this group, we may be best to **avoid targeting them** in our marketing.

Conclusion

This was a complex but fascinating project.

As usual in data science, cleaning and feature engineering took 80% of the time, but the resulting clustering was well worth the effort.

We've identified four segments showcasing distinct purchasing habits and reactions to marketing offers.

Most importantly, we've identified an entire segment of subpar targets that we can exclude in our paid marketing campaigns to optimize our Customer Acquisition Cost.

Many challenges in working with this dataset resulted from repeat exposures to the same offer over different channels, and imperfect conversion attribution. In future experiments, it would be desirable to generate more accurate data on the source of each conversion, and confirm completion through coupon codes or a separate redemption mechanism in the Starbucks Rewards app.

Additionally, many customers had missing profile fields. Understandably, we cannot force members to respond to some of these questions for ethical and legal reasons.

Above, I have imputed missing income values with the mean, and engineered a separate feature tracking customers who did not respond to this question. This approach could be further improved by imputing the data using a supervised Machine Learning algorithm (predicting income based on the other demographic traits), or using the mean income of residents in the customer's neighbourhood (not provided in this dataset).

It would also be fascinating to explore more of the data, and segment customers further by product line. For example, many Starbucks BOGO offers involve new and seasonal drinks, which may receive very different reactions depending on how conservative or adventurous is the recipient of the offer.