Unsupervised Learning Assignment - Customer Personality Analysis

Overview - Main objective

We are working for a company that owns and manages a grocery store and we are part of its data analytics division that supports various functions such as marketing, purchasing and warehousing. The head of the analytics department has assigned us a project from the marketing team, to identify different customer personality clusters, in other words to segment clients in terms of their purchases and shopping habits. The goal of this analysis is to help the company better understand its customers and allow the marketing team to better promote products to specific client types. It also makes it easier for the business to understand the needs, behaviours and concerns of the respective customer segments and more easily modify the product line.

For this analysis, we will leverage various unsupervised learning techniques that will allow us to cluster the customers in groups. The aim is to find which model will produce reasonable segments that the marketing team will be able to act upon, by creating more personalized suggestions. The main objective of this report is to provide an overview of the data that was used for this analysis, go through the steps taken to clean, analyze and understand the data, the models that were trained and how these were compared against each other as well as our final recommendation towards the head of analytics and the marketing team. The rest of the report is structured as follows: the next two sections describe a) the data used for this analysis and b) the steps taken to clean and analyze the data. This will allow us to anticipate potential issues with our data and treat them accordingly so that we can prepare it for our clustering algorithms. Potential hypotheses about the data will be presented and their validity will be reviewed in the following sections. Then we will provide details on the different clustering algorithms that were used in our analysis and compare their benefits and shortcomings. Finally, our findings are presented and potential flaws are recognized, including suggestions for future research.

Data

For this analysis, we leverage data on customer personality that can be found here. There are entries about 2240 customers, each one identified by a unique ID in the dataset. For each customer, a variety of data is available, spread across 28 columns. Information includes year of birth, number of kids at home (and at what relative age, e.g. kids, teens, etc.), marital status, level of education, income and various metrics on purchase habits such as amount of meat/fruits/vegetables/etc. bought. See below a complete data dictionary:

- **People**
  ID: Customer's unique identifier
  Year_Birth: Customer's birth year
  Education: Customer's education level
  Marital_Status: Customer's marital status
  Income: Customer's yearly household income
  Kidhome: Number of children in customer's household
  Teenhome: Number of teenagers in customer's household
  Dt_Customer: Date of customer's enrollment with the company
  Recency: Number of days since customer's last purchase
  Complain: 1 if the customer complained in the last 2 years, 0 otherwise
- **Products**
  MntWines: Amount spent on wine in last 2 years
  MntFruits: Amount spent on fruits in last 2 years
  MntMeatProducts: Amount spent on meat in last 2 years
  MntFishProducts: Amount spent on fish in last 2 years
  MntSweetProducts: Amount spent on sweets in last 2 years
  MntGoldProds: Amount spent on gold in last 2 years
- **Promotion**
  NumDealsPurchases: Number of purchases made with a discount
  AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
  AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
  AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
  AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
  AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
  Response: 1 if customer accepted the offer in the last campaign, 0 otherwise
- **Place**
  NumWebPurchases: Number of purchases made through the company's website
  NumCatalogPurchases: Number of purchases made using a catalogue
  NumStorePurchases: Number of purchases made directly in stores
  NumWebVisitsMonth: Number of visits to company's website in the last month

We note that the data is almost fully complete. For 24 customers income information is not available, representing around 1% of the overall dataset. Summary statistics for all variables can be seen below, while distribution details will be covered in the following section:

People

statistic str	ID f64	Year_Birth f64	Education str	Marital_Status str	Income f64
”count”	2240.0	2240.0	”2240”	”2240”	2216.0
”null_count”	0.0	0.0	”0”	”0”	24.0
”mean”	5592.159821	1968.805804	null	null	52247.251354
”std”	3246.662198	11.984069	null	null	25173.076661
”min”	0.0	1893.0	”2n Cycle”	”Absurd”	1730.0
”25%”	2829.0	1959.0	null	null	35322.0
”50%”	5462.0	1970.0	null	null	51390.0
”75%”	8427.0	1977.0	null	null	68487.0
”max”	11191.0	1996.0	”PhD”	”YOLO”	666666.0

statistic str	Kidhome f64	Teenhome f64	Dt_Customer str	Recency f64	Complain f64
”count”	2240.0	2240.0	”2240”	2240.0	2240.0
”null_count”	0.0	0.0	”0”	0.0	0.0
”mean”	0.444196	0.50625	”2013-07-10 10:01:42.857000”	49.109375	0.009375
”std”	0.538398	0.544538	null	28.962453	0.096391
”min”	0.0	0.0	”2012-07-30”	0.0	0.0
”25%”	0.0	0.0	”2013-01-16”	24.0	0.0
”50%”	0.0	0.0	”2013-07-09”	49.0	0.0
”75%”	1.0	1.0	”2013-12-30”	74.0	0.0
”max”	2.0	2.0	”2014-06-29”	99.0	1.0

Products

statistic str	MntWines f64	MntFruits f64	MntMeatProducts f64	MntFishProducts f64
”count”	2240.0	2240.0	2240.0	2240.0
”null_count”	0.0	0.0	0.0	0.0
”mean”	303.935714	26.302232	166.95	37.525446
”std”	336.597393	39.773434	225.715373	54.628979
”min”	0.0	0.0	0.0	0.0
”25%”	24.0	1.0	16.0	3.0

statistic	MntWines	MntFruits	MntMeatProducts	MntFishProducts
str	f64	f64	f64	f64
”50%”	174.0	8.0	67.0	12.0
”75%”	504.0	33.0	232.0	50.0
”max”	1493.0	199.0	1725.0	259.0

statistic	MntSweetProducts	MntGoldProds
str	f64	f64
”count”	2240.0	2240.0
”null_count”	0.0	0.0
”mean”	27.062946	44.021875
”std”	41.280498	52.167439
”min”	0.0	0.0
”25%”	1.0	9.0
”50%”	8.0	24.0
”75%”	33.0	56.0
”max”	263.0	362.0

Promotion

statistic	NumDealsPurchases	AcceptedCmp1	AcceptedCmp2	AcceptedCmp3	AcceptedCmp4
str	f64	f64	f64	f64	f64
”count”	2240.0	2240.0	2240.0	2240.0	2240.0
”null_count”	0.0	0.0	0.0	0.0	0.0
”mean”	2.325	0.064286	0.013393	0.072768	0.074554
”std”	1.932238	0.245316	0.114976	0.259813	0.262728
”min”	0.0	0.0	0.0	0.0	0.0
”25%”	1.0	0.0	0.0	0.0	0.0
”50%”	2.0	0.0	0.0	0.0	0.0
”75%”	3.0	0.0	0.0	0.0	0.0
”max”	15.0	1.0	1.0	1.0	1.0

statistic	AcceptedCmp5	Response
str	f64	f64
”count”	2240.0	2240.0
”null_count”	0.0	0.0
”mean”	0.072768	0.149107

statistic str	AcceptedCmp5 f64	Response f64
”std”	0.259813	0.356274
”min”	0.0	0.0
”25%”	0.0	0.0
”50%”	0.0	0.0
”75%”	0.0	0.0
”max”	1.0	1.0

Place

statistic str	NumWebPurchases f64	NumCatalogPurchases f64	NumStorePurchases f64	NumWebVisitsMonth f64
”count”	2240.0	2240.0	2240.0	2240.0
”null_count”	0.0	0.0	0.0	0.0
”mean”	4.084821	2.662054	5.790179	5.316518
”std”	2.778714	2.923101	3.250958	2.426645
”min”	0.0	0.0	0.0	0.0
”25%”	2.0	0.0	3.0	3.0
”50%”	4.0	2.0	5.0	6.0
”75%”	6.0	4.0	8.0	7.0
”max”	27.0	28.0	13.0	20.0

Data Exploration and Data Cleaning

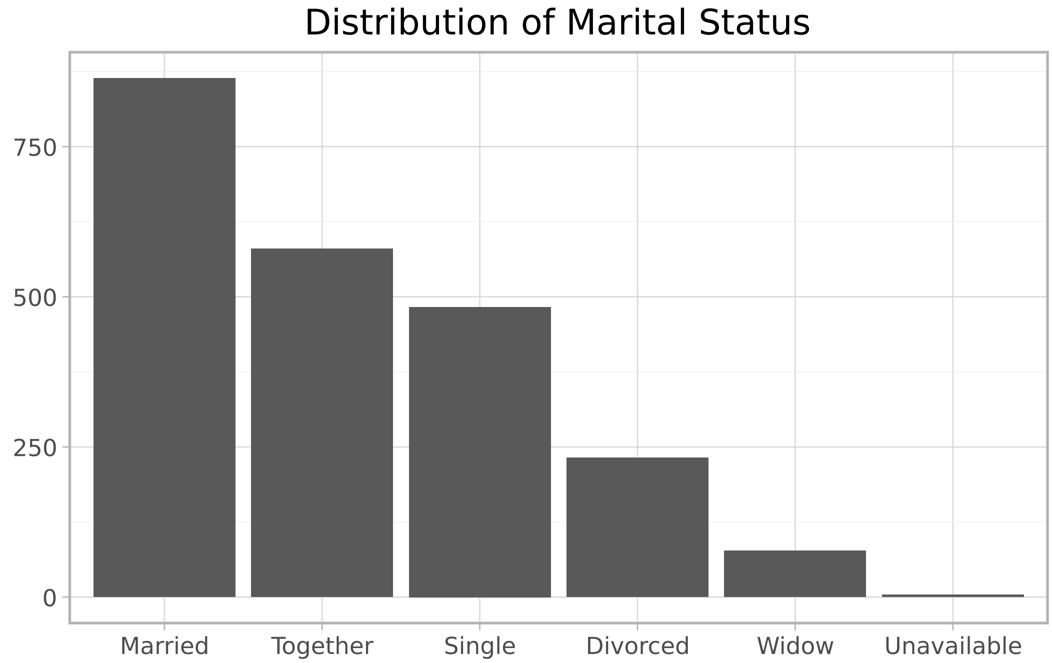
The high level view of our data already highlighted potential issues that we need to look into and remedy before proceeding with our modeling approach (e.g. unexpected replies in marital status). They might arise from a deficiency in the design of the data collection process, in the case of marital status the use of a text field instead of providing pre-determined answers for example.

Let us start by exploring the two categorical variables in our dataset, marital status and education. For marital status, looking at a count of the different values, we observe a few cases where unexpected answers are given (Alone, Absurd, YOLO). Alond by itself is not entirely unexpected, but the typical answer is Single.

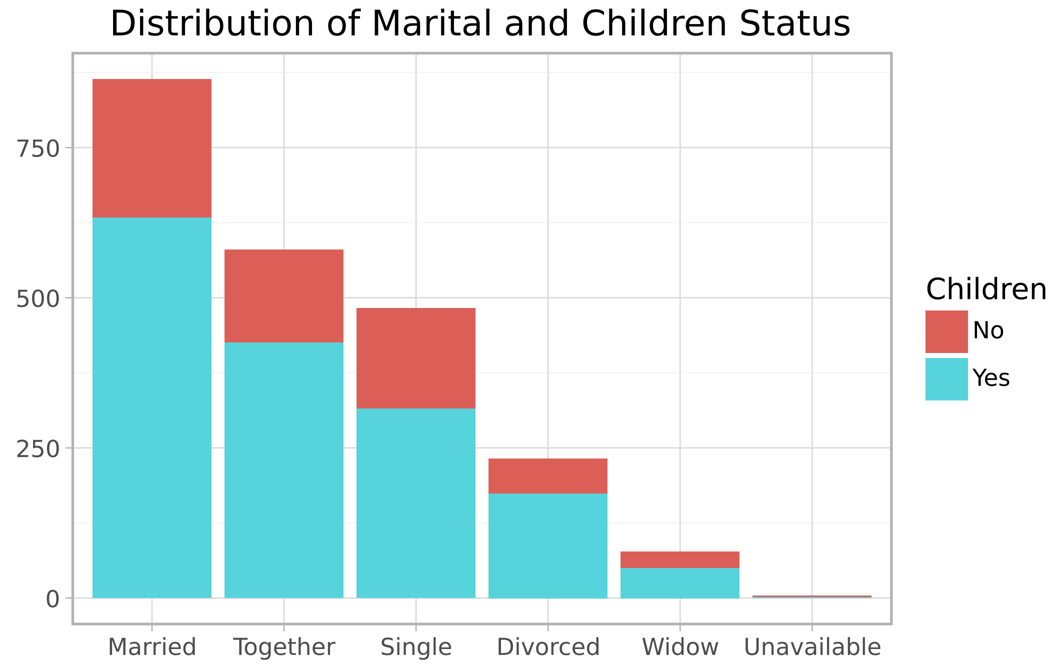
Marital_Status str	count u32
”Married”	864
”Together”	580

Marital_Status str	count u32
”Single”	480
”Divorced”	232
”Widow”	77
”Alone”	3
”Absurd”	2
”YOLO”	2

For our analysis we will recode Alone into Single and set unexpected values to Unavailable. In the next section, when we apply our modeling strategies, we assume that unexpected values also belong in the Single category.

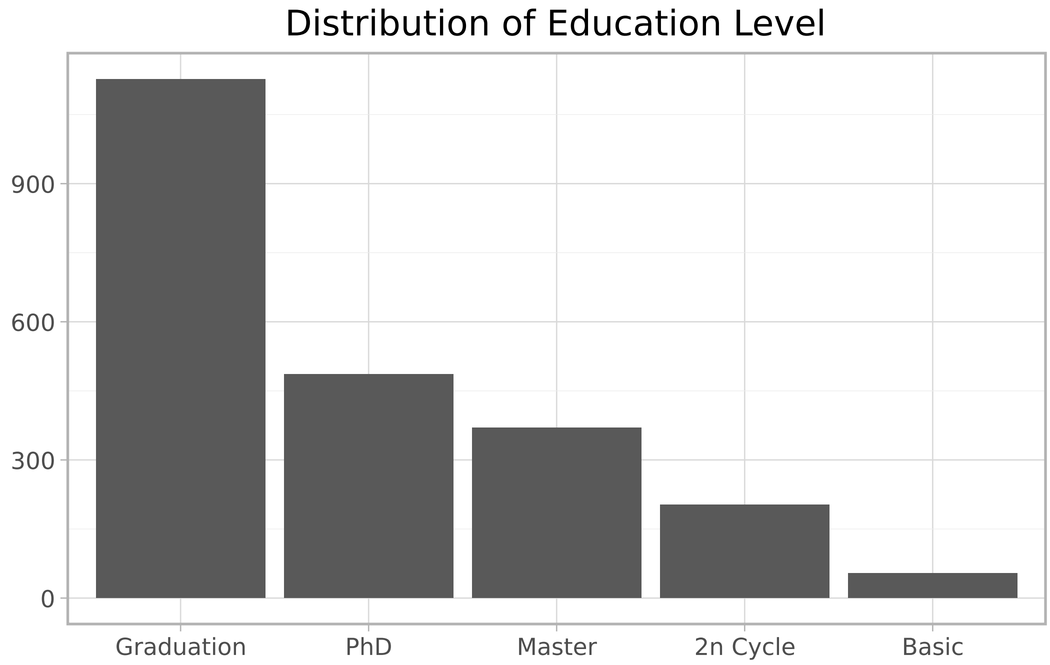


We observe that the majority of customers in our data are either married or in a relationship, while the rest are single, divorced or widowed. We can assume that the majority of our customers shop for a household of at least two people. We will create a new variable for the total number of childen (Children = Kidhome + Teenhome) Let’s check how many are the customers in each marital category have children (Children variable, recoded as Yes and No in the charts only):

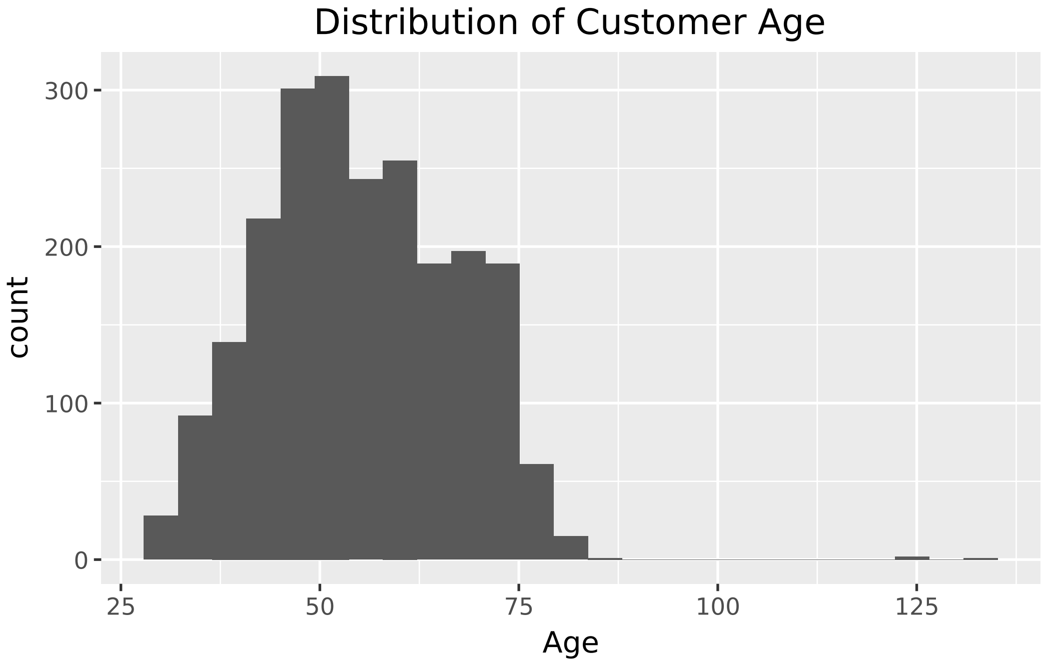


The majority of customers in our data have children, which is true for all categories, including those that are single. We can potentially infer that a portion of people who provide single as marital status and have children were previously in a relationship in which a child was born.

In terms of education level, half of our customers have graduated highschool, while more than one third have a Master or Phd. Less than three percent have only basic education.



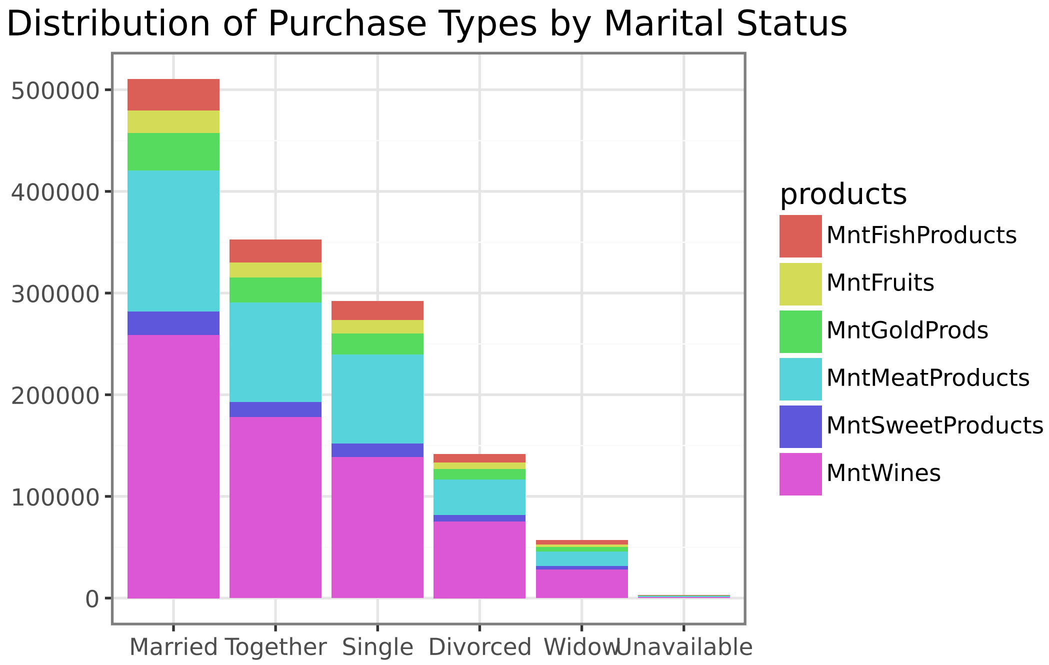
We now turn our attention to some of the other variables in our dataset. We can calculate the Age of our customers as of today (year 2024) by subtracting the current year from the Year_Birth column. Below is the distribution of ages for our customers:



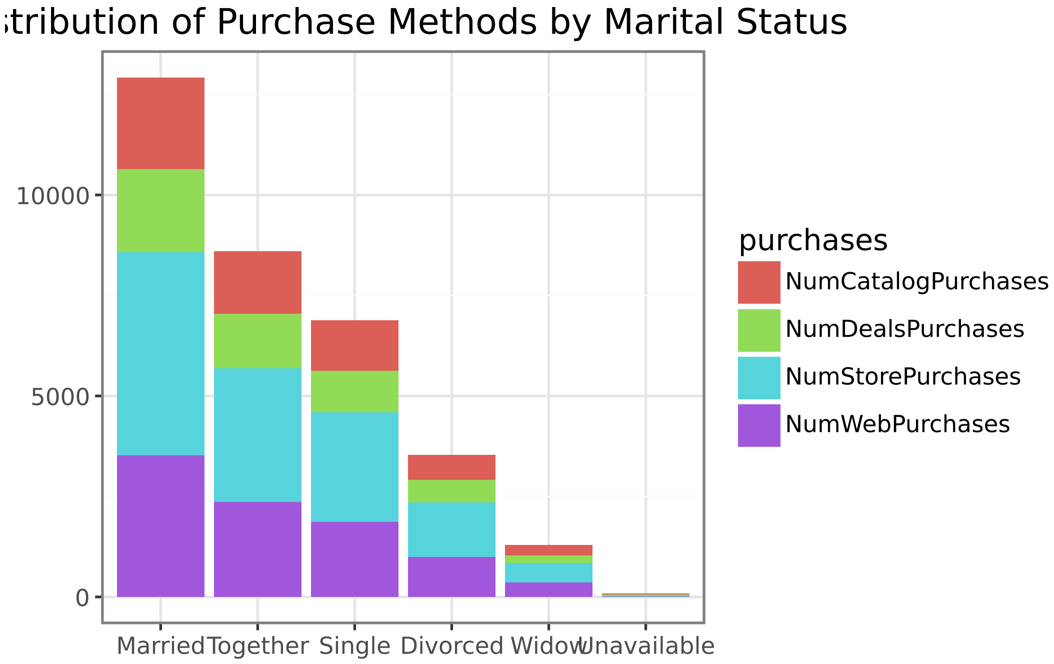
We immediately notice that there are some outliers in terms of age, some customers seem to be more than 120 years old (oldest customer is 131). We will exclude those from our dataset before we train any of our models as these ages are unlikely to be correct and might have adverse effects for our clustering strategy. Other than that, the age distribution of our customers is relatively normal, somewhat fatter on the higher end. The median age of our customers is 54 years old.

Our data includes information on the five promotional events and whether our clients accepted offers on any of them. We realize the caveat of storing data like this and using it to create customer segments is problematic, as the company will probably hold more events in the future and new clients will not have participated in previous events but might be interested in future ones. We choose to create new variables that will be used in our models, Accepted_Offers which is 1 if a customer accepted any offer in the past and 0 if they do not, as well as Mean_Offer_Accepted, which is the mean rate with which customers accept offers. This allows us to recalculate these metrics for our clientbase and include in our models new customers. One potential issue of this approach is that existing customers might exhibit different behaviour in the future, resulting in them moving from one segment to the other. This, however, is expected over the customer lifetime. It is important for the business to be able to identify when customer habits change so that marketing strategy can be adjusted.

Next, we check spending habits of our customers in terms of how they purchase products and on what types of products they spend on. Since we have aggregate spending habits of customers, we realized that for existing customers the spending is only going to increase or remain the same, while new customers will have lower overall spending. For use in modeling, we will create new variables based on the frequency that a customer buys from store, online or via catalog, as well as based on the percentage of overall spending a customer assigns to various categories (e.g. Meat, Wine, Fish, etc). We will also create a variable to account for the amount of time a customer has been with the company based on the Dt_Customer column, as well as the percentage of a customer's spending in terms of their overall income



We note that majority of spending is on wine and then meat products across all types of customers in terms of marital status.

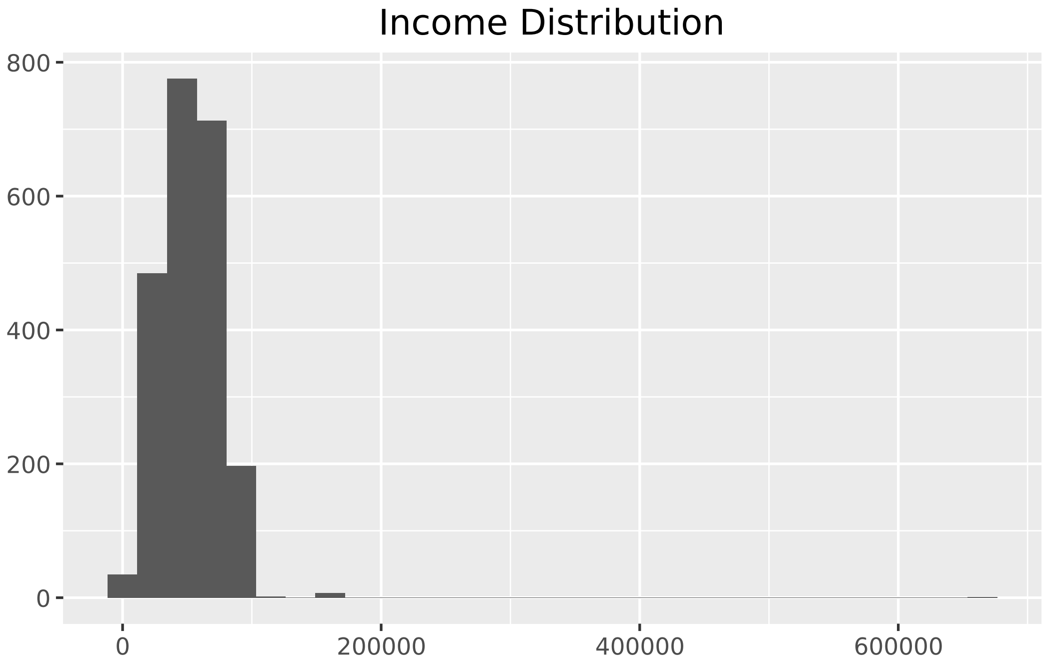


In terms of how customers prefer to purchase their products, most purchases are done directly at the store, while web purchases are also popular across all customer groups (in terms of marital status).

Finally, we check the income distribution and note that a small number of customers has quite higher income compared to the rest. We will try to normalize such outliers by using the logarithm of income going forward.

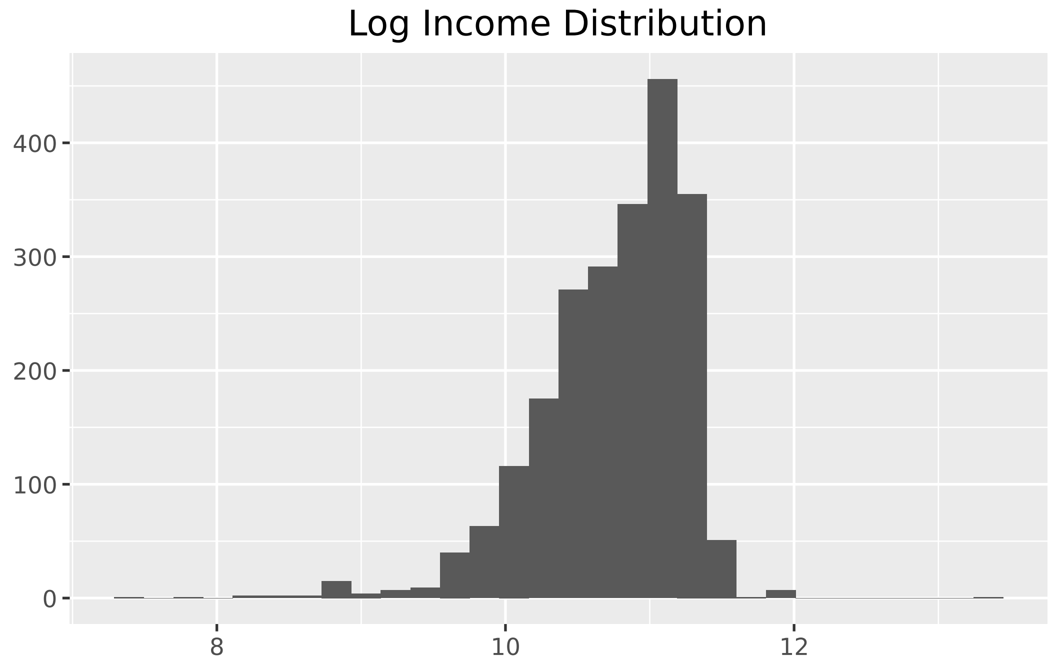
```
(
    ggplot(data, aes(x = "Income")) +
    geom_histogram(bins = 30) +
    ggtitle("Income Distribution") +
    labs(x="", y="")
)
```

```
/home/kpatelis/projects/ibm_ml/.venv/lib/python3.12/site-packages/plotnine/layer.py:284: Plo
```



```
(
    ggplot(data.with_columns(Income = col("Income").log()), aes(x = "Income")) +
    geom_histogram(bins = 30) +
    ggtitle("Log Income Distribution") +
    labs(x="", y="")
)
```

```
/home/kpatelis/projects/ibm_ml/.venv/lib/python3.12/site-packages/plotnine/layer.py:284: Plo
```



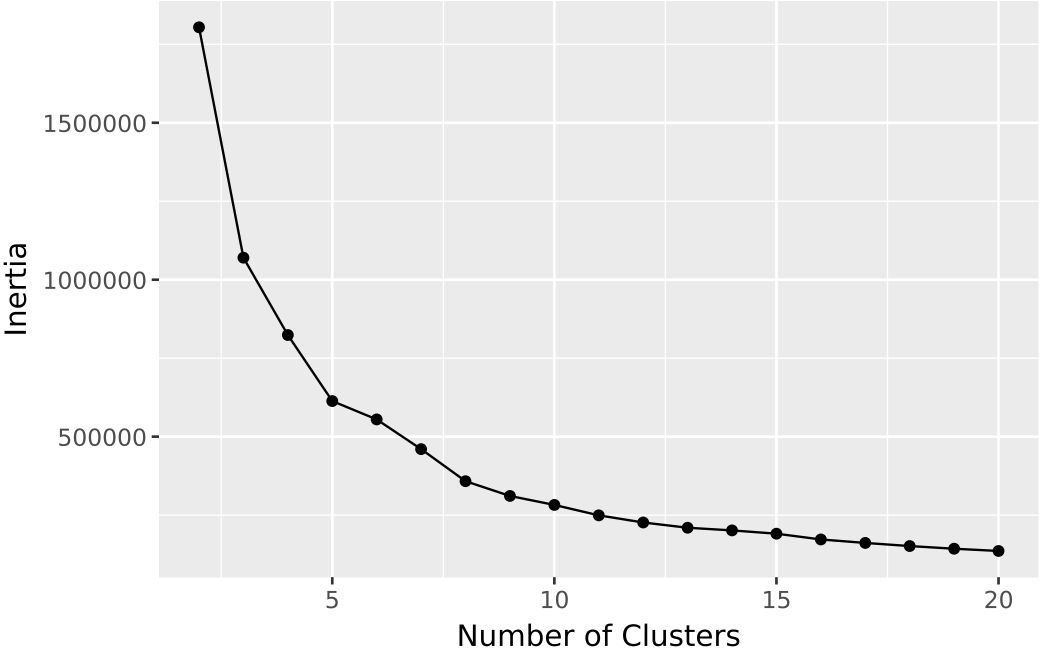
In addition to the data cleaning and feature engineering that was discussed, we also replace the income of the customers where that information is missing with the median value across the dataset (before filtering out customers above 100 years old). We also drop variables Z_CostContact and Z_Revenue as they have zero variance and provide no benefit. Categorical variables of marital status and education will be one-hot-encoded.

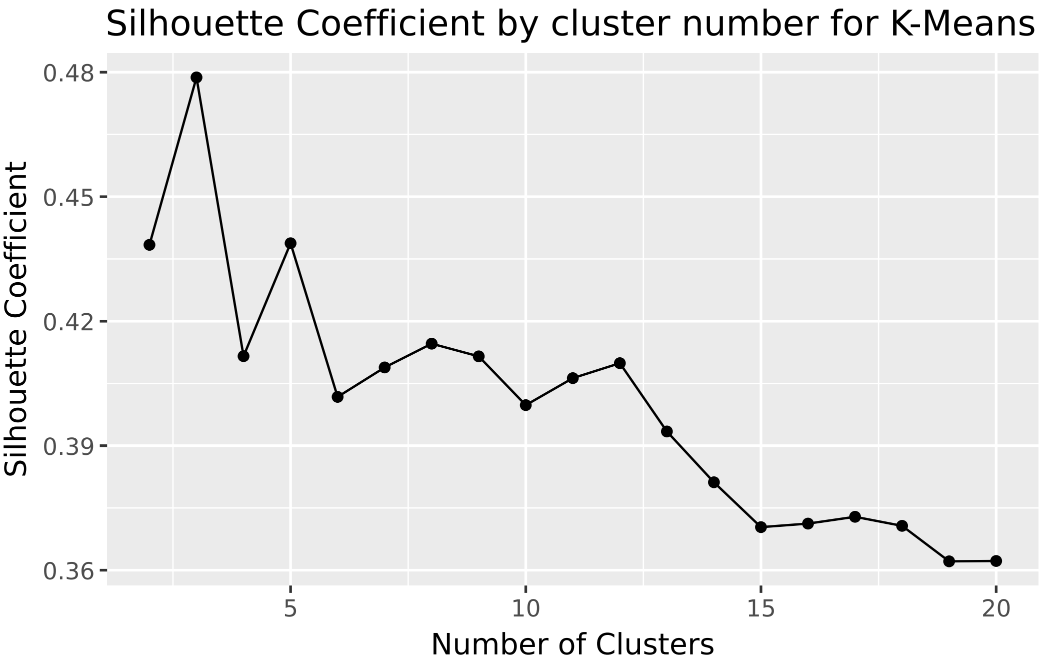
Unsupervised Learning Models - Clustering Algorithms

We will examine three different popular clustering algorithms, KMeans, DBSCAN, and Agglomerative Clustering that have been successfully used in many different domains. For the KMeans algorithm we will also look into methods that would point us towards the number of customer clusters, while DBSCAN can automatically come up with the appropriate number. In all cases, our data will be normalized and the categorical variables will be one-hot-encoded. To reduce the impact from correlated variables and reduce the number of variables we have to work with, we will also be using the t-SNE dimensionality reduction technique. The aim of this analysis is to find the most appropriate clustering algorithm, so we will not spend additional time examining different dimensionality reduction techniques. That can be a future extension of the project.

KMeans

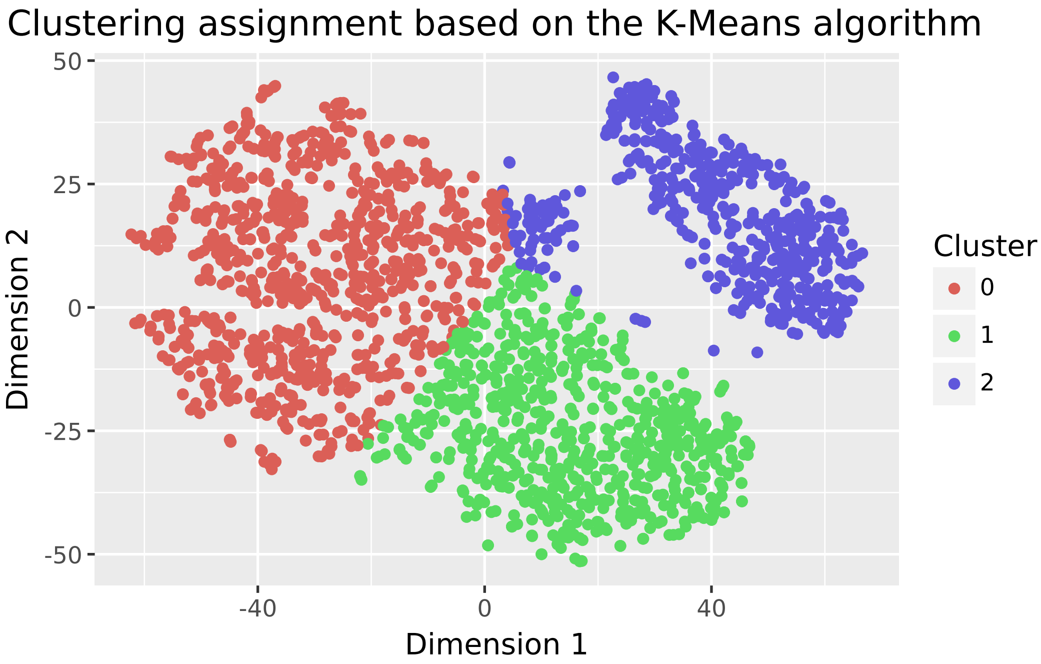
First, we explore the KMeans algorithm, where we need to define the number of clusters. Sometimes, depending on the domain or problem at hand, we are able to ascertain the number of clusters beforehand. When that is not the case, there are a couple techniques that are typically used to identify the appropriate number, such as the the elbow method, which relies on identifying the number of clusters up to which inertia drops at a faster rate, or such as using the silhouette score. See below plots of the inertia and silhouette score against the number of clusters:





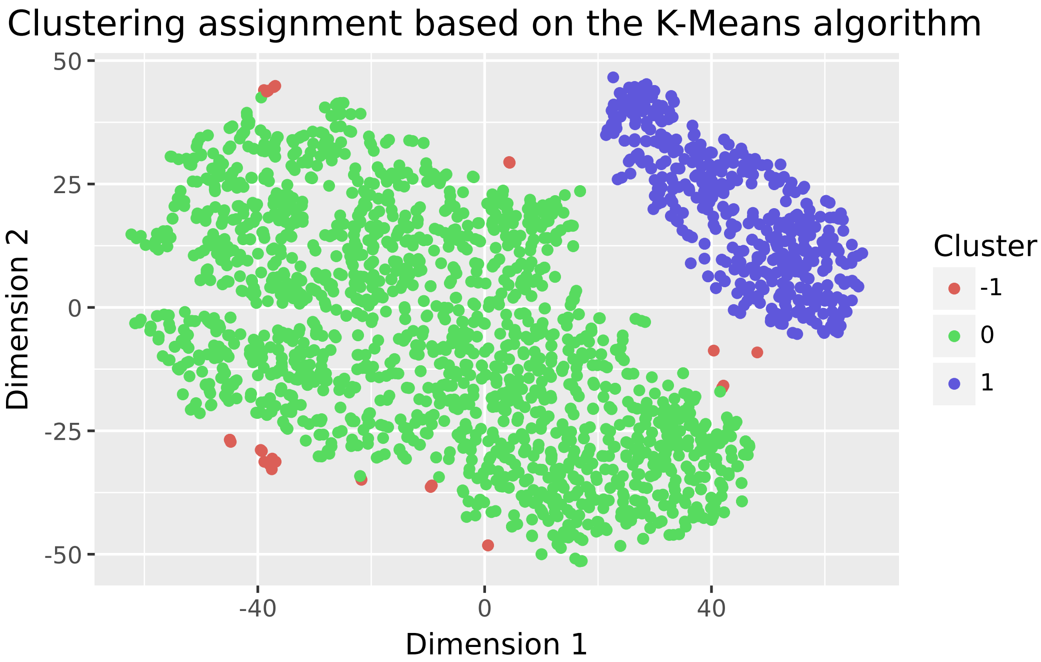
We can see that from the silhouette coefficient plot that three clusters seem to work best, although two also seem like viable options. The coefficient is the mean of the silhouette sample, which compares the distance of a point to other instances of the same cluster, against the distance of the point to instances of the next closest cluster. We can also use the silhouette coefficient across our analysis as a metric to help us select the most appropriate clustering algorithm, but we have to keep in mind that it tends to be higher for convex clusters such as those produced by K-Means, compared to density based clusters as those produced by DBSCAN.

Before moving on to the next algorithm, let us also plot the clusters against the data points to see how well the data is separated:



DBSCAN

DBCAN works differently compared to K-Means. In this case we do not need to define the number of clusters, the model will be able to deduce that. We only need to define the minimum number of samples in each cluster and the maximum distance between two samples for one to be considered in the neighbourhood of the other. We select the minimum number of samples to 10 and then distance to 5. See below the clusters assigned by the model:

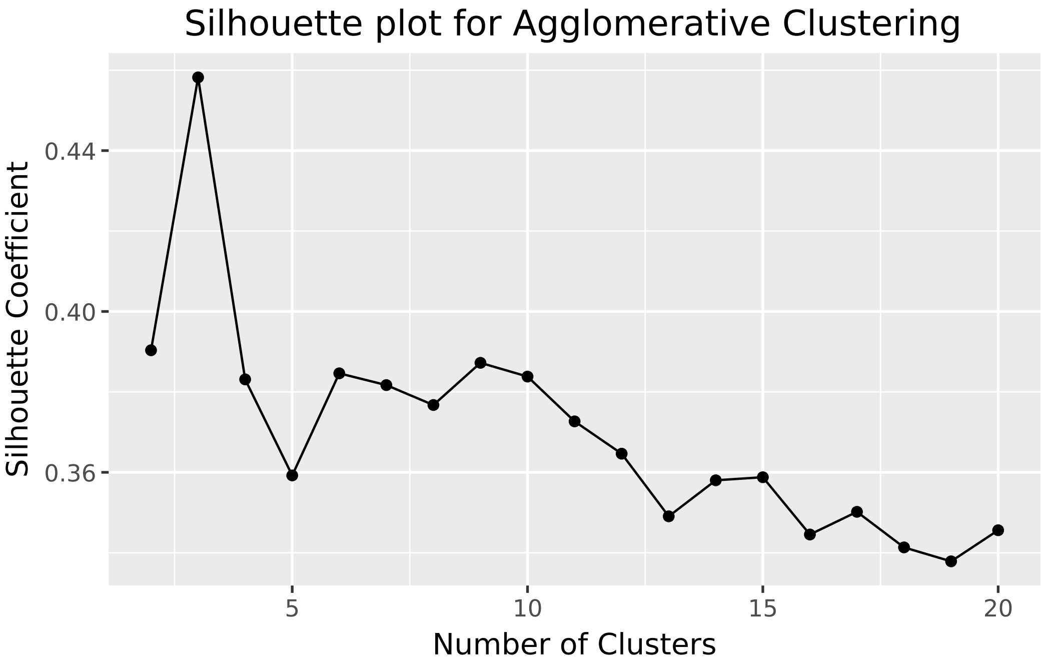


We observe that the algorithm split the data into three classes, which are clearly separable. There are also several points that received a label of -1. This means that these points are deemed as outliers by the algorithm. The algorithm seems more capable of separating the two “island” clusters that can be seen in the plot, compared to K-Means. It also identifies a small cluster of points that can be seen on the left side of the plot.The silhouette coefficient for this model is 0.226, which is less than half of K-Means at 0.478. This suggests that K-Means is more appropriate to DBSCAN. However, we already mentioned that the coefficient is generally lower for density based models such as DBSCAN. In addition, one reason for the lower score is that many points were deemed as outliers. Visually, we observe that outliers would most likely be assigned to the green cluster if we relaxed model parameters. If we assume cluster -1 points are assigned to cluster 0, then the coefficient score jumps to 0.382.

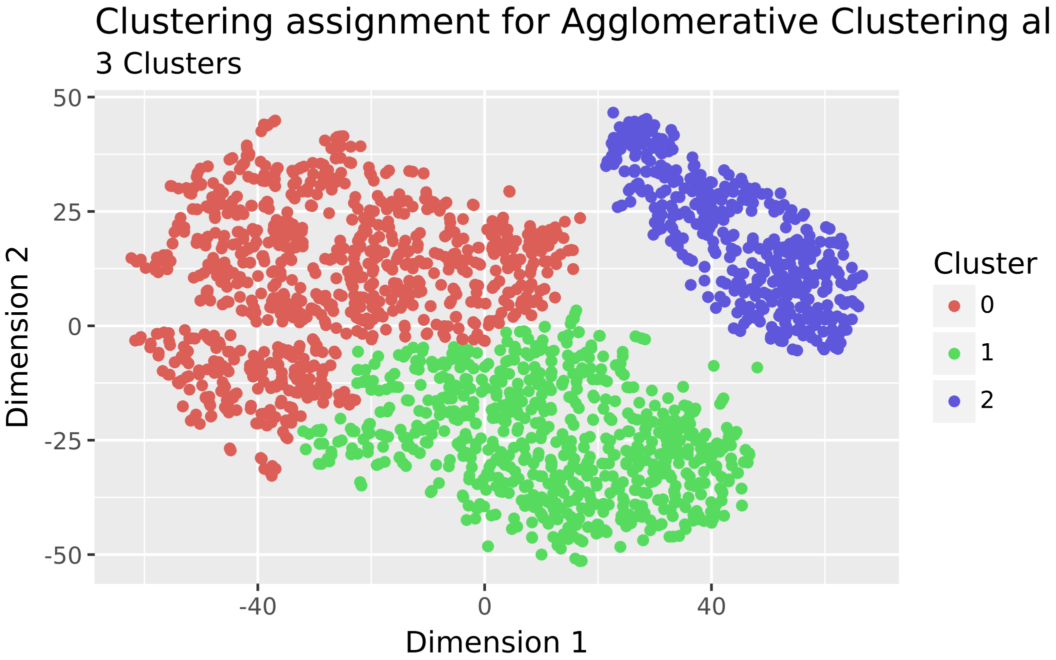
```
(np.float64(0.47876675019750153), np.float64(0.22157789916232212))
```

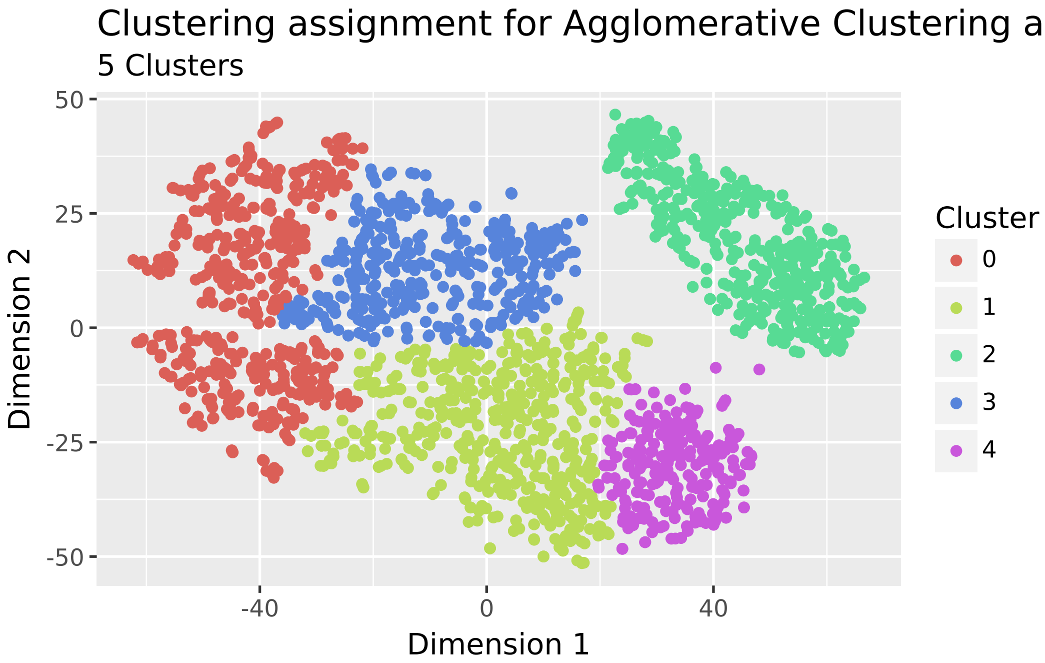
Agglomerative Clustering

For the hierarchical agglomerative clustering technique, we once again need to define the number of clusters. We make use of the silhouette plot to help us identify the appropriate number.



Similar to K-Means, the most appropriate option seems to be 3 clusters, although we could also go for 5 clusters. In the following plots we present both cases.





Perhaps in future analysis we will try to identify more granular customer segments, but for now we will opt for 3 instead of 5 clusters and compare against the other techniques. The silhouette coefficient when using three clusters is 0.394, somewhat lower than the K-Means score but higher compared to DBSCAN. Furthermore, this model more or less identifies one cluster that is the same as outlined by DBSCAN, while it splits into two the other large cluster identified by DBSCAN.

```
np.float32(0.3592003)
```

In the next section we gather and summarise our findings and pick the final model to be used for customer segmentation.

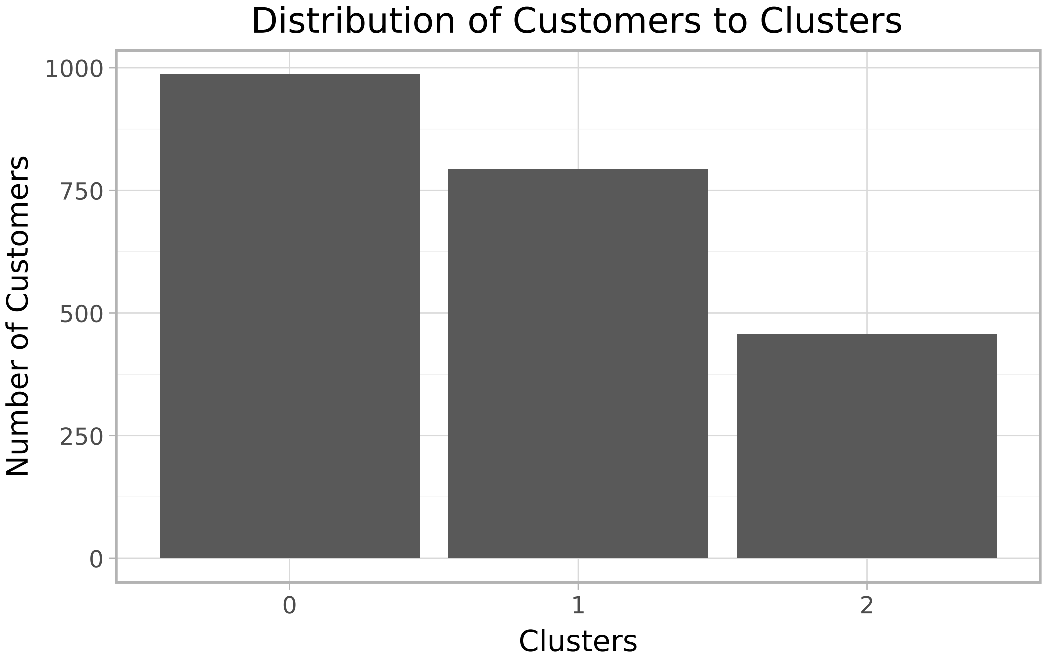
Findings

We analyzed our customer data using three different models and were able to extract useful information. All three models are able to identify at least two clearly distinct segments. Both DBSCAN and Agglomerative Clustering are able to distinguish between these two regions quite clearly, while K-Means also assigns some points from the “left” area to the cluster that we can visually see on the right, which makes us think the model is not fully capable of segmenting clients correclty. K-Means also seems to have almost linear boundaries between clusters. On the other hand, DBSCAN is only able to separate the data into two main classes, but also classifies many clients as outliers, making it difficult for us to come up with an effective marketing strategy for them. As we aim to classify all of our client base, our solution should also be able to do the same, which is why we do not think DBSCAN is a valid option.

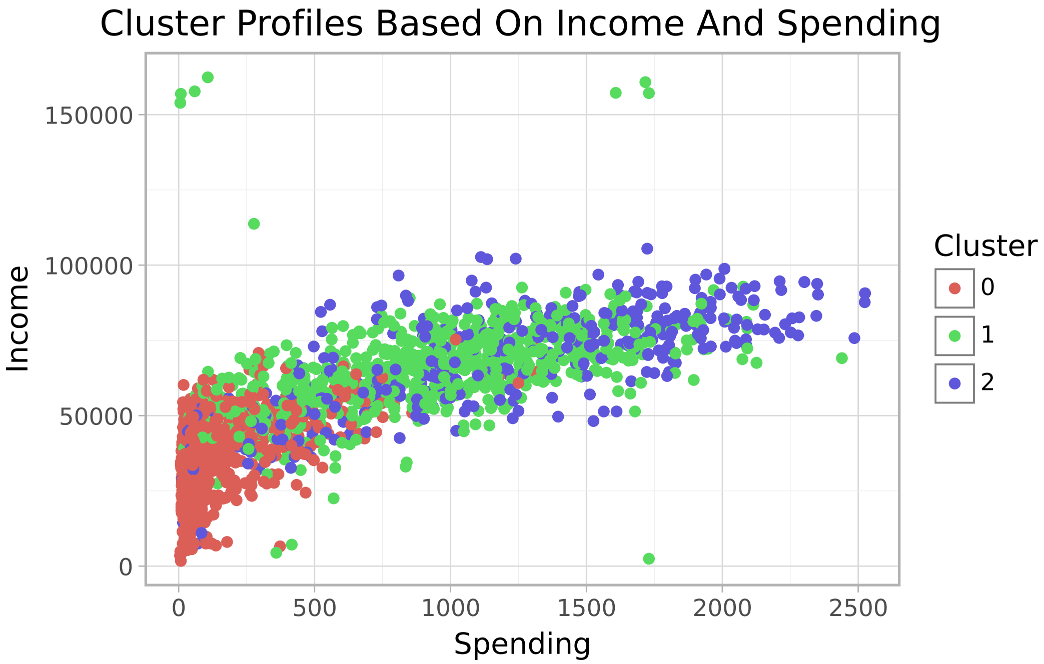
Out of the three techniques, Agglomerative Clustering seems to be able to clearly segment our customers into groups. Visually, the groups are well defined, with no areas where clusters blend together. This approach also seems to provide meaningful clusters on a more granular level, if that is something desired in the future. Its silhouette coefficient is only slightly lower compared to K-Means, which can also be partially attributed to the fact that the score is typically higher for circular based clustering approaches. For these reasons, we believe Agglomerative Clustering to be the most appropriate algorithm.

Let us use these clusters to identify characteristics about the three customer segments identified.

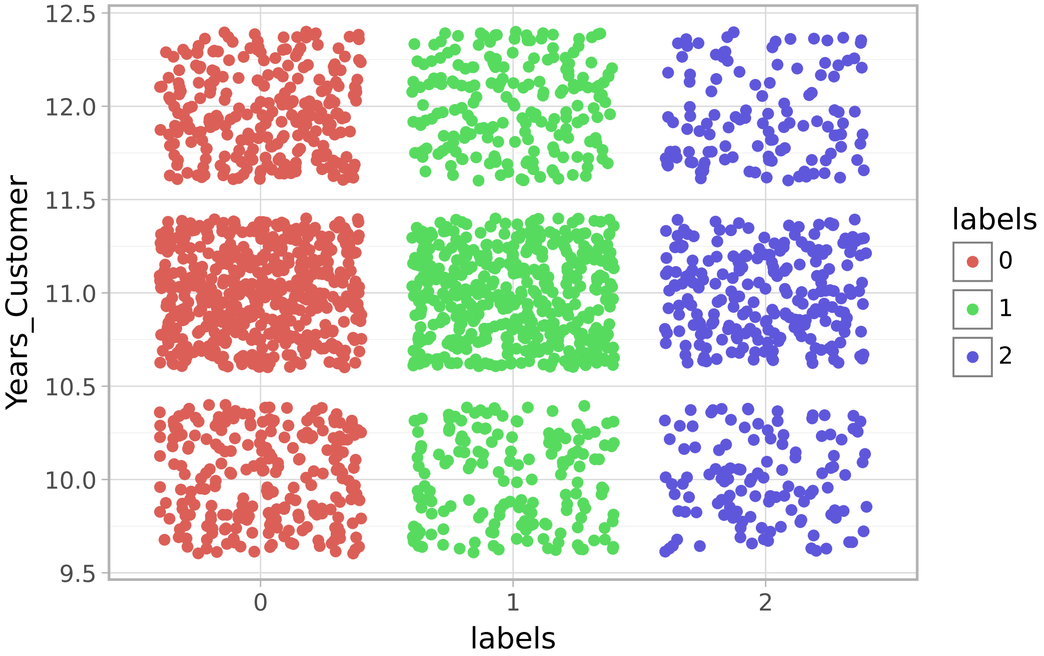
The clusters seem fairly distributed.



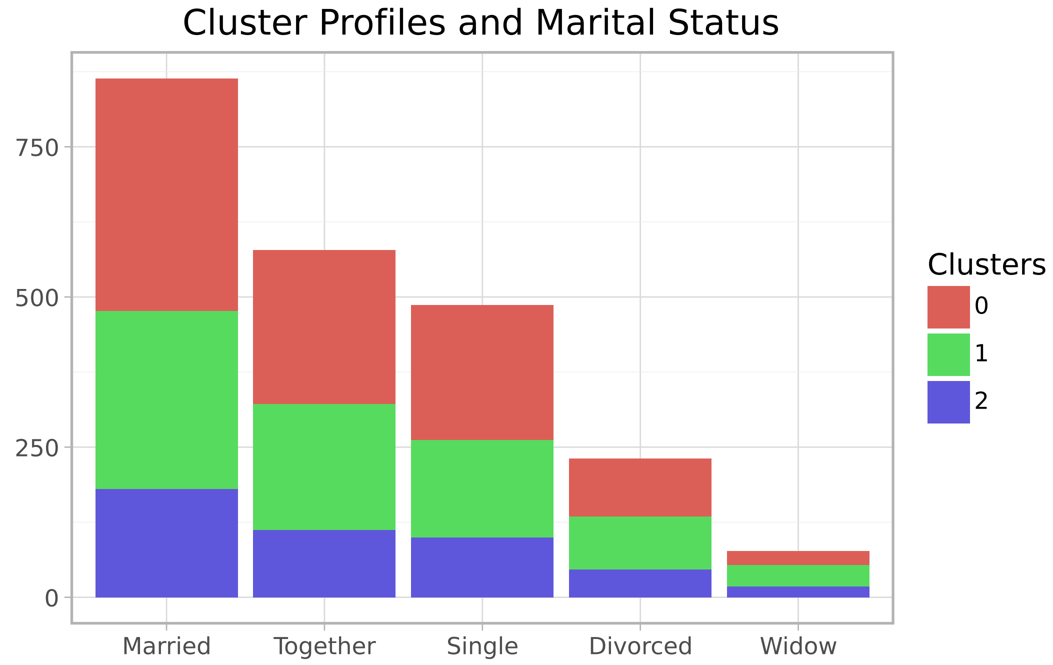
Next, we visualize customer income against their spending. We observe that cluster 1 is closely related mostly to customers with limited spending, as well as low income customers (not always the case).

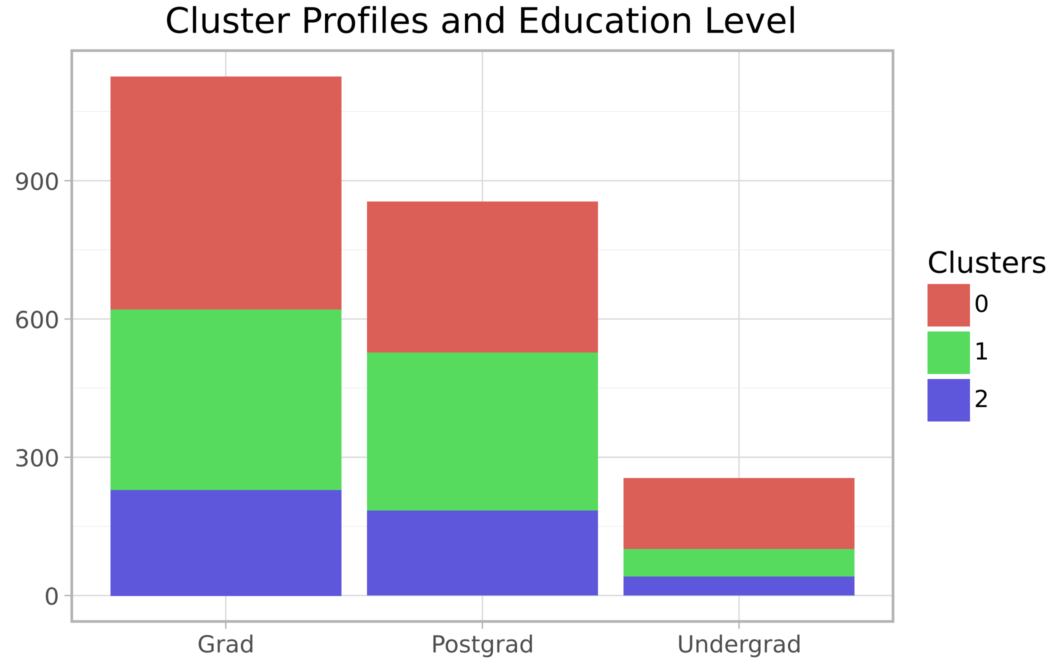


We also observe that cluster 1 contains most people that have been customers for the least amount of time, typically most that have been customers less than 10 years are in that group.

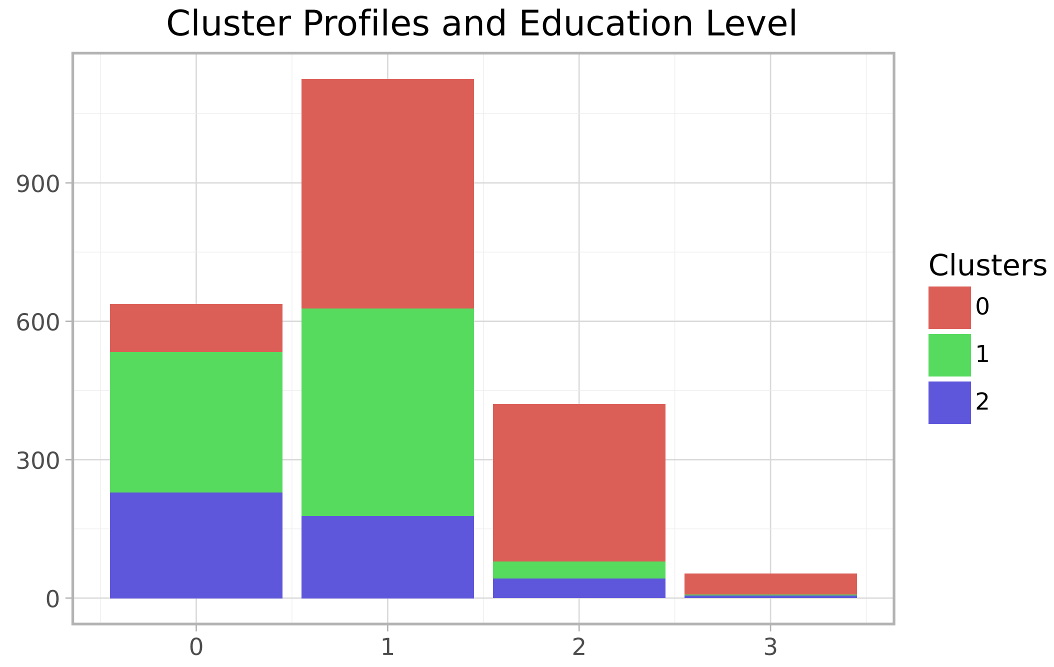


However, none of our techniques seem to provide useful clusters with regards to family situation (single, couple, married, with/without children) or education level.

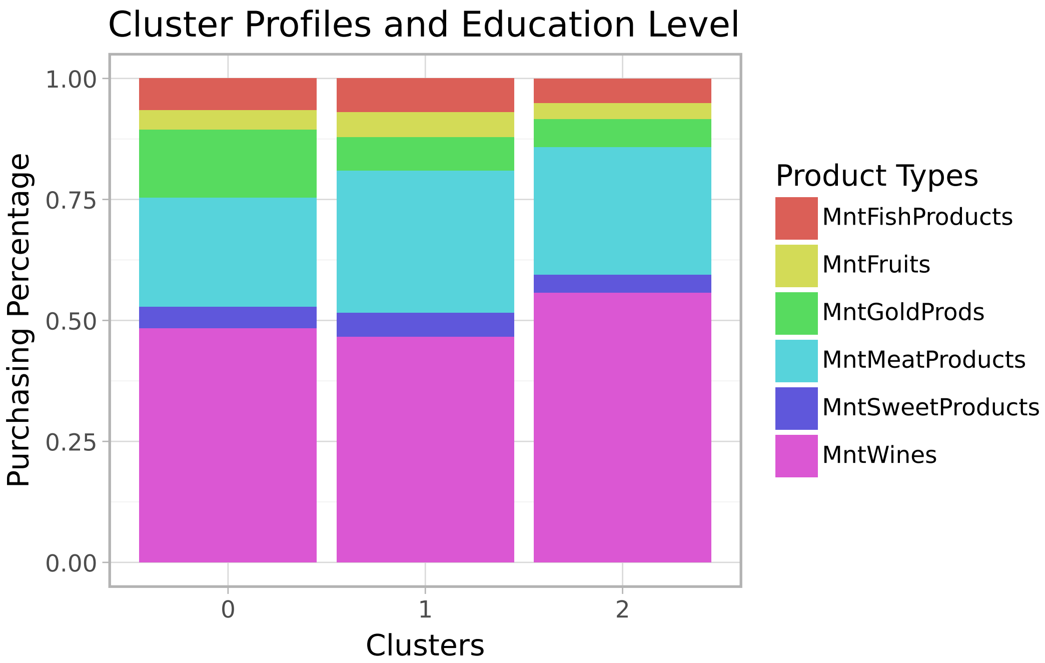




Clusters 0 and 2 are mostly comprised by customers with no or at most one child, while the majority of cluster 1 one or more children typically.



In terms of purchasing habits, cluster 2 seems to purchase more wine products, while cluster 1 purchases more gold products compared to the other clusters.



To summarise, we believe agglomerative clustering to be the most appropriate technique, while we were able to distinguish some characteristics that define these clusters. In the next section we will discuss some of the flaws in our research and potential next steps.

Flaws and Future Improvements

We were able to get some valuable insights on our customers based on our clustering analysis. However, there are a couple flaws worth highlighting.

First of all, we were not able to cluster our customer base on the basis of their demographics, primarily marital status, education level and number of chilren. This may be due to the way we encoded our variables for use in our models, typically using one-hot-encoding. In the next iteration of our study we will try different techniques, such as label encoding or frequency based encoding in combination with other variables.

We should also aim to even better cluster our customers on the basis of their spending habits and the frequency with which they accept deals. One possible way to do this is to increase the number of clusters our models produce. We observed that agglomerative clustering is able to create well defined clusters even as we increase their number. The same was not possible with DBSCAN and seemed to have less success with K-Means. We can try to reproduce our analysis with 5 clusters. Additionally, we can try creating more sophisticated variables via feature engineering that would allow better segmentation.

Conclusion

To summarise, agglomerative clustering models seem to work best at segmenting our client base. We achieve clearly defined clusters from which we are able to draw useful conclusions. These first findings are already going to be of practical use to our marketing team. We realize our models still lack sophistication to segment based on common demographics and some purchasing habits. We plan to address these open points by invetigating modifications to our agglomerative clustering model as well as improved feature engineering.