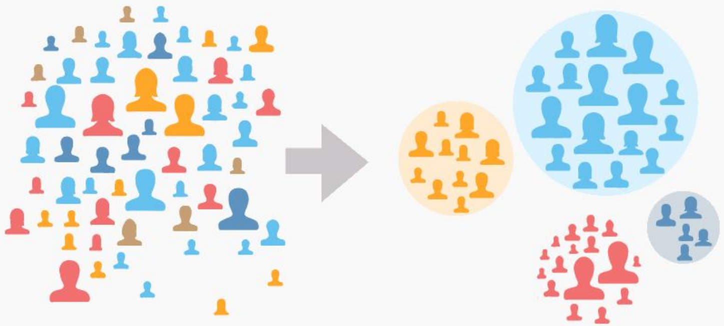
Customer Behavior Analysis and Prediction



By:
Andrew Cash, Samantha Cole, Jeffrey Felger, and Esha Soni

The Goal of Our Project

Our primary goal is to create a data pipeline that facilitates the analysis and prediction of customer behavior.
Ultimately, we want to segment different customer demographics and identify "high value" customers within our dataset, and also find any interesting purchase patterns that may be present.

To do this, we will 1. Prepare our data, 2. Perform exploratory data analysis to find insights on our data, 3. Apply a clustering algorithm on our data, 4. Mine association rules from those returned clusters, 5. Report interesting association rules, and 6. Visualize the results.

Tools we used:

Programmed in Python

Pandas, Numpy

Matplotlib

Scikit-learn, mlxtend

Plotly

Project can be found at:

https://github.com/acash11/Customer-Behavior-Prediction

First Step: Preprocessing the Data

- Ensure there are no missing attributes or duplicate entries
- Select relevant attributes to be clustered
- Standardize the data so all attributes are consistently scaled
- Turn categorical data into numerical data if it is relevant for clustering

Example: Test Data

Input

```
time in store,money spent,Cluster_Label,Home State,Hobby
43.20416453623548,45.189790710106436,1,Pennsylvania,sports
28.089361809874255,25.19476707126188,0,West Virginia,reading
17.13622302390135,25.726241434576135,0,Indiana,cooking
22.85846609272064,28.120992423542784,0,West Virginia,writing
25.981259930621942,30.65412917408302,0,Pennsylvania,food
22.89628427307887,25.062174232138464,0,Kentucky,art
25.20571235287003,44.741550275204844,0,Pennsylvania,art
52.37070112787636,50.887365951412,1,West Virginia,sports
35.34555259755493,25.193074499727956,0,Ohio,music
57.70417006904963,53.468495743316296,1,Ohio,sports
55.50632625601979,38.240480020524274,1,Pennsylvania,sports
28.083881209803504,26.570796824107592,0,Kentucky,reading
30.997857451271145,14.21769174555563,0,Ohio,writing
22.1935151872733,21.14010499386739,0,Indiana,reading
18.240509515979884,25.026904296968812,0,Virginia,music
28.108504365224317,25.971602735059165,0,Kentucky,sports
43.83984394657416,58.92375602351627,1,Virginia,sports
53.320533930493426,51.896891913733754,1,Michigan,sports
31.286462404621446,24.13112998185445,0,Kentucky,music
46.18535969201308,51.90073001313465,1,Ohio,sports
50.398034424728664,46.731348562616034,1,West Virginia,sports
31.766597904739772,29.040051693175762,0,Kentucky,writing
28.704362431683396,28.566957829710496,0,Michigan,sports
```

- Irrelevant categorical data (Cluster_Label, Home State, Hobby) is dropped completely
- Remaining numerical data is standardized around category's mean, and expressed in SD distance
- Customer IDs are assigned to data points for easier processing and identification in later steps

Output

```
time in store,money spent,Customer ID
-1.0145937280624542,-1.0328442399342361,0
-0.7217400351459241,-0.6874316642625523,1
1.5766501493326788,1.0890392317660857,2
1.409073498634317,1.0793926669778677,3
0.726884459397137,0.026527917991503263,4
-0.514275707653322,-1.4658801315998298,5
-0.5402245650378233,-1.1282027357423547,6
-0.31085592197183537,-1.1369811504797698,7
-0.8819645422716569,-1.7623172214149312,8
0.6602101667243823,1.3372383382256159,9
1.3748813164200329,0.9565849549268179,10
-1.0664325217397297,-1.0456630085092136,11
-1.231032828770389,-1.664958926338958,12
-1.490488375461158,-1.2763238305407707,13
0.7000564951898738,0.37681577565112306,14
1.133708143942475,1.0563333221460194,15
0.8998225962691406,1.2318791232023458,16
```

Example: Shopping Trends

Input

```
Customer ID,Age,Gender,Item Purchased,Category,Purchase Amount (USD),Location,Size,Color,Season,Review Rating,Subscription Status,Pa
1,55,Male,Blouse,Clothing,53,Kentucky,L,Gray,Winter,3.1,Yes,Credit Card,Express,Yes,Yes,14,Venmo,Fortnightly
2,19,Male,Sweater,Clothing,64,Maine,L,Maroon,Winter,3.1,Yes,Bank Transfer,Express,Yes,Yes,2,Cash,Fortnightly
3,50,Male,Jeans,Clothing,73,Massachusetts,S,Maroon,Spring,3.1,Yes,Cash,Free Shipping,Yes,Yes,23,Credit Card,Weekly
4,21,Male,Sandals,Footwear,90,Rhode Island,M,Maroon,Spring,3.5,Yes,PayPal,Next Day Air,Yes,Yes,49,PayPal,Weekly
5,45,Male,Blouse,Clothing,49,Oregon,M,Turquoise,Spring,2.7,Yes,Cash,Free Shipping,Yes,Yes,31,PayPal,Annually
6,46,Male,Sneakers,Footwear,20,Wyoming,M,White,Summer,2.9,Yes,Venmo,Standard,Yes,Yes,14,Venmo,Weekly
7,63,Male,Shirt,Clothing,85,Montana,M,Gray,Fall,3.2,Yes,Debit Card,Free Shipping,Yes,Yes,49,Cash,Quarterly
8,27,Male,Shorts,Clothing,34,Louisiana,L,Charcoal,Winter,3.2,Yes,Debit Card,Free Shipping,Yes,Yes,19,Credit Card,Weekly
9,26,Male,Coat,Outerwear,97,West Virginia,L,Silver,Summer,2.6,Yes,Venmo,Express,Yes,Yes,8,Venmo,Annually
10,57,Male,Handbag,Accessories,31,Missouri,M,Pink,Spring,4.8,Yes,PayPal,2-Day Shipping,Yes,Yes,4,Cash,Quarterly
11,53,Male,Shoes,Footwear,34,Arkansas,L,Purple,Fall,4.1,Yes,Credit Card,Store Pickup,Yes,Yes,26,Bank Transfer,Bi-Weekly
12,30,Male,Shorts,Clothing,68,Hawaii,S,Olive,Winter,4.9,Yes,PayPal,Store Pickup,Yes,Yes,10,Bank Transfer,Fortnightly
13,61,Male,Coat,Outerwear,72,Delaware,M,Gold,Winter,4.5,Yes,PayPal,Express,Yes,Yes,37,Venmo,Fortnightly
14,65,Male,Dress,Clothing,51,New Hampshire,M,Violet,Spring,4.7,Yes,Debit Card,Express,Yes,Yes,31,PayPal,Weekly
15,64,Male,Coat,Outerwear,53,New York,L,Teal,Winter,4.7,Yes,PayPal,Free Shipping,Yes,Yes,34,Debit Card,Weekly
16,64,Male,Skirt,Clothing,81,Rhode Island,M,Teal,Winter,2.8,Yes,Credit Card,Store Pickup,Yes,Yes,8,PayPal,Monthly
17,25,Male,Sunglasses,Accessories,36,Alabama,S,Gray,Spring,4.1,Yes,Venmo,Next Day Air,Yes,Yes,44,Debit Card,Bi-Weekly
18,53,Male,Dress,Clothing,38,Mississippi,XL,Lavender,Winter,4.7,Yes,Debit Card,2-Day Shipping,Yes,Yes,36,Venmo,Quarterly
19,52,Male,Sweater,Clothing,48,Montana,S,Black,Summer,4.6,Yes,Bank Transfer,Free Shipping,Yes,Yes,17,Cash,Weekly
20,66,Male,Pants,Clothing,90,Rhode Island,M,Green,Summer,3.3,Yes,Venmo,Standard,Yes,Yes,46,Debit Card,Bi-Weekly
21,21,Male,Pants,Clothing,51,Louisiana,M,Black,Winter,2.8,Yes,Credit Card,Express,Yes,Yes,50,Cash,Every 3 Months
22,31,Male,Pants,Clothing,62,North Carolina,M,Charcoal,Winter,4.1,Yes,Credit Card,Store Pickup,Yes,Yes,22,Debit Card,Quarterly
23,56,Male,Pants,Clothing,37,California,M,Peach,Summer,3.2,Yes,Cash,Store Pickup,Yes,Yes,32,Debit Card,Annually
24,31,Male,Pants,Clothing,88,Oklahoma,XL,White,Winter,4.4,Yes,Credit Card,Express,Yes,Yes,40,Credit Card,Weekly
25,18,Male,Jacket,Outerwear,22,Florida,M,Green,Fall,2.9,Yes,Cash,Store Pickup,Yes,Yes,16,Debit Card,Weekly
26,18,Male,Hoodie,Clothing,25,Texas,M,Silver,Summer,3.6,Yes,Bank Transfer,Express,Yes,Yes,14,PayPal,Annually
27,38,Male,Jewelry,Accessories,20,Nevada,M,Red,Spring,3.6,Yes,Cash,Next Day Air,Yes,Yes,13,Credit Card,Annually
28,56,Male,Shorts,Clothing,56,Kentucky,L,Cyan,Summer,5.0,Yes,Debit Card,Next Day Air,Yes,Yes,7,Bank Transfer,Every 3 Months
29,54,Male,Handbag,Accessories,94,North Carolina,M,Gray,Fall,4.4,Yes,Debit Card,Free Shipping,Yes,Yes,41,PayPal,Every 3 Months
30,31,Male,Dress,Clothing,48,Wyoming,S,Black,Fall,4.1,Yes,Venmo,Store Pickup,Yes,Yes,14,Credit Card,Weekly
31,57,Male,Jewelry,Accessories,31,North Carolina,L,Black,Winter,4.7,Yes,Bank Transfer,Standard,Yes,Yes,16,Credit Card,Monthly
32,33,Male,Dress,Clothing,79,West Virginia,L,Brown,Winter,4.7,Yes,Venmo,Store Pickup,Yes,Yes,45,Venmo,Monthly
33,36,Male,Jacket,Outerwear,67,Kansas,M,Silver,Summer,4.9,Yes,Bank Transfer,Free Shipping,Yes,Yes,37,Venmo,Annually
34,54,Male,Pants,Clothing,38,Colorado,L,Green,Summer,3.3,Yes,Venmo,Store Pickup,Yes,Yes,45,Cash,Quarterly
35,36,Male,T-shirt,Clothing,91,North Dakota,L,Violet,Spring,4.6,Yes,Debit Card,2-Day Shipping,Yes,Yes,38,PayPal,Quarterly
36,54,Male,Blouse,Clothing,33,Massachusetts,M,Cyan,Summer,4.0,Yes,Bank Transfer,2-Day Shipping,Yes,Yes,48,Credit Card,Bi-Weekly
37,35,Male,T-shirt,Clothing,69,Illinois,M,Maroon,Winter,4.6,Yes,Cash,Free Shipping,Yes,Yes,44,PayPal,Fortnightly
38,35,Male,Jeans,Clothing,45,Indiana,S,Cyan,Summer,2.8,Yes,Debit Card,Store Pickup,Yes,Yes,10,PayPal,Weekly
39,29,Male,Dress,Clothing,37,Florida,M,Red,Winter,3.7,Yes,Debit Card,2-Day Shipping,Yes,Yes,44,Venmo,Every 3 Months
40,70,Male,Pants,Clothing,60,Arizona,S,Turquoise,Summer,4.2,Yes,Bank Transfer,Express,Yes,Yes,18,Credit Card,Monthly
41,69,Male,Handbag,Accessories,76,Louisiana,L,Beige,Winter,4.6,Yes,PayPal,Next Day Air,Yes,Yes,31,Debit Card,Quarterly
42,67,Male,Scarf,Accessories,39,Alaska,M,Orange,Spring,4.5,Yes,Cash,Standard,Yes,Yes,40,Venmo,Annually
43,20,Male,Coat,Outerwear,100,Tennessee,M,Beige,Spring,4.1,Yes,Bank Transfer,Free Shipping,Yes,Yes,15,PayPal,Annually
```

- Irrelevant categories, again, dropped
- Values, again, standardized
- Frequency converted to actual day values before standardization

Output

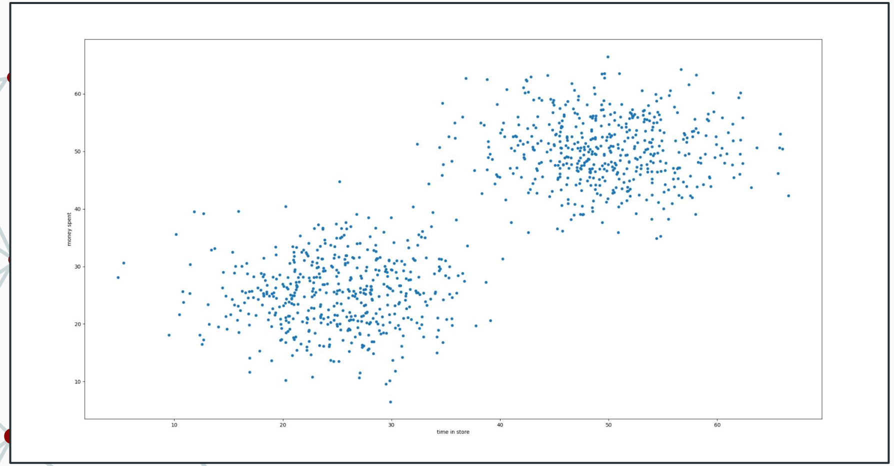
```
Customer ID,Purchase Amount (USD),Frequency of Purchases (per year),Previous Purchases
1,-0.285592018181966,0.9143253903291526,-0.7857299170320132
2,0.17882925394594182,1.042776651014307,-1.6163449950085267
3,0.5588102947778664,0.6376488356302705,-0.1627686085496282
4,1.2765522607937239,2.100436262024655,1.6368973937328175
5,-0.4544724807739325,-0.7962272343755805,0.39097477676804737
6,-1.6788558345656894,2.8495780218543767,-0.7857299170320132
7,1.0654516825537659,-1.0563762363115887,1.6368973937328175
8,-1.0877742154938068,2.141879121616121,-0.43964030120846603
9,1.5720930703296654,-1.1929009732846008,-1.20103745602027
10,-1.2144345624377817,-0.9736559251655412,-1.4779091486791078
11,-1.0877742154938068,0.19563904074484995,0.044885160944500124
12,0.34770971653790833,0.35496688907436086,-1.062601609690851
13,0.5165901791298748,0.3041016350763188,0.806282315756304
14,-0.3700322494779492,2.09891130772377,0.39097477676804737
15,-0.285592018181966,1.8224836962852426,0.5986285462621758
16,0.8965712199617993,-0.050261148598730876,-1.20103745602027
17,-1.0033339841978235,1.277758546063644,1.2908077779092701
18,-0.9188937529018403,-0.24771535936123923,0.7370643925915946
19,-0.4966925964219241,1.9192484004252448,-0.5780761475378849
20,1.2765522607937239,0.7770554658358335,1.429243624238689
21,-0.3700322494779492,-1.307724459582761,1.706115316897527
22,0.09438902264995859,-1.1842950638614698,-0.23198653171433767
23,-0.9611138685498319,-0.6830471921341494,0.4601926999327568
24,1.1921120294977408,2.249388672489736,1.0139360852504324
25,-1.5944156032697063,2.8892185218802107,-0.6472940707025944
26,-1.4677552563257314,-1.1405456428652703,-0.7857299170320132
```

Exploratory Data Analysis - Initial Visualization

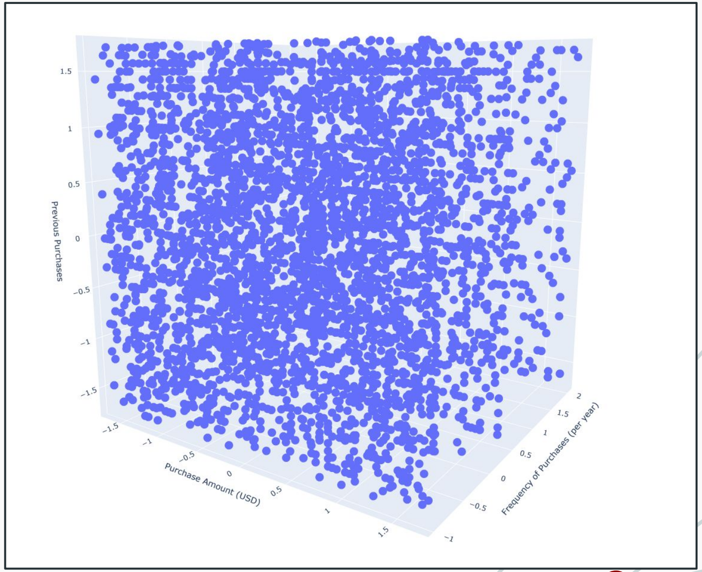
EDA helps to understand the nature of a dataset, namely by revealing things such as the data range, the data distribution, the presence of noises and outliers, and visually obvious correlations.

Exploratory Data Analysis

Test Data - 2D



Shopping Trends: Recency, Frequency, Amount Spent



Clustering Algorithms

We defined two clustering algorithms
in our project

K-means Clustering

```
def sklearn_ml_kmeans(df, original_data, k):
    df = df.drop('Customer ID', axis='columns')
    # Apply K-Means clustering (e.g., 2 clusters)
    kmeans = KMeans(n_clusters=k)


    #make sure processed data was clustered
    #print(df)
    cluster_labels = kmeans.fit_predict(df)
    #print(cluster_labels)
    df['Cluster'] = cluster_labels
    original_data['Cluster'] = cluster_labels

    sil_score = silhouette_score(df, cluster_labels)
    db_score = davies_bouldin_score(df, cluster_labels)

    cluster_dfs = [original_data[original_data['Cluster'] == i] for i in range(kmeans.n_clusters)]

    return(cluster_dfs, sil_score, db_score)
```

Agglomerative Clustering

```
def sklearn_ml_agglomerative(df, original_data, threshold):
    df = df.drop('Customer ID', axis='columns')
    #apply agglomerative clustering - ward linkage method is the default
    agglomerative = AgglomerativeClustering(n_clusters=threshold, linkage='ward', compute_distances=True)
    agglomer = agglomerative.fit(df)

    #make sure processed data was clustered
    #print(df)
    cluster_labels = agglomerative.fit_predict(df)
    #print(cluster_labels)
    df['Cluster'] = cluster_labels
    original_data['Cluster'] = cluster_labels

    sil_score = silhouette_score(df, cluster_labels)
    db_score = davies_bouldin_score(df, cluster_labels)

    cluster_dfs = [original_data[original_data['Cluster'] == i] for i in range(len(set(cluster_labels)))]

    return(cluster_dfs, sil_score, db_score, agglomer)
```

Quality of Clustering

We used two evaluations for the quality of our clusters

Silhouette Score

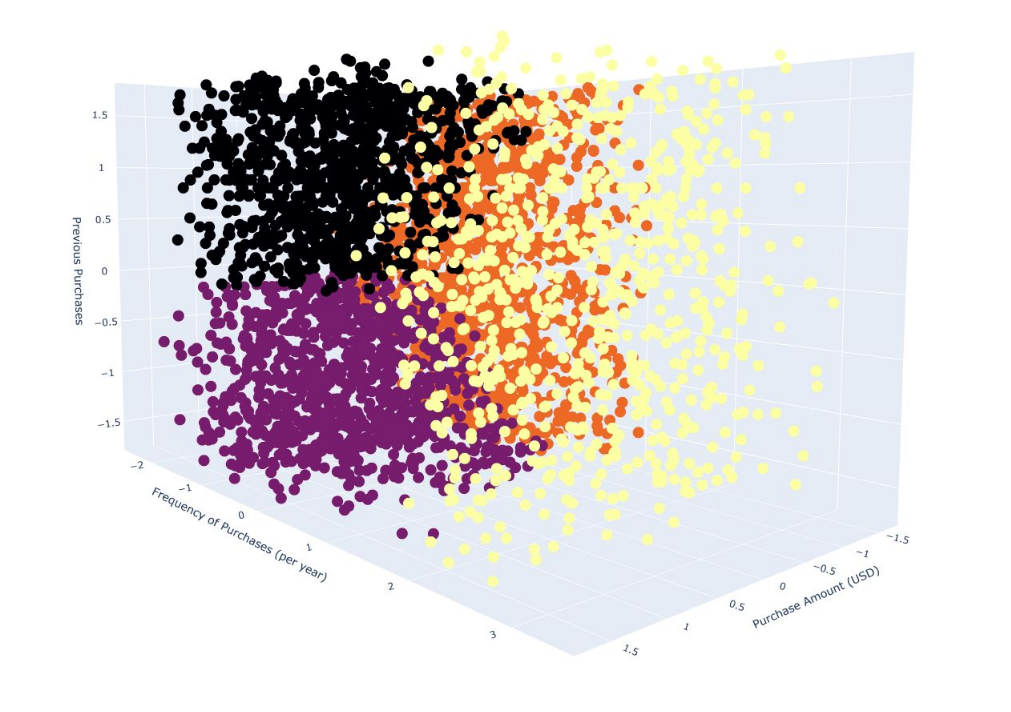
Measures how similar an object is to its own cluster, compared to how similar it is to other clusters. The silhouette ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. A clustering with an average silhouette score of over 0.7 is considered to be "strong", a score over 0.5 "reasonable" and over 0.25 is "weak".

Davies-Bouldin Score

The average similarity of each cluster to its most similar cluster. Similarity is defined as the ratio between inter-cluster and intra-cluster distances. Ranks well-separated clusters with less dispersion as having a better score. Ranges from 0 to infinity, (dependant on the range if the dataset) with lower scores indicating better defined clusters.

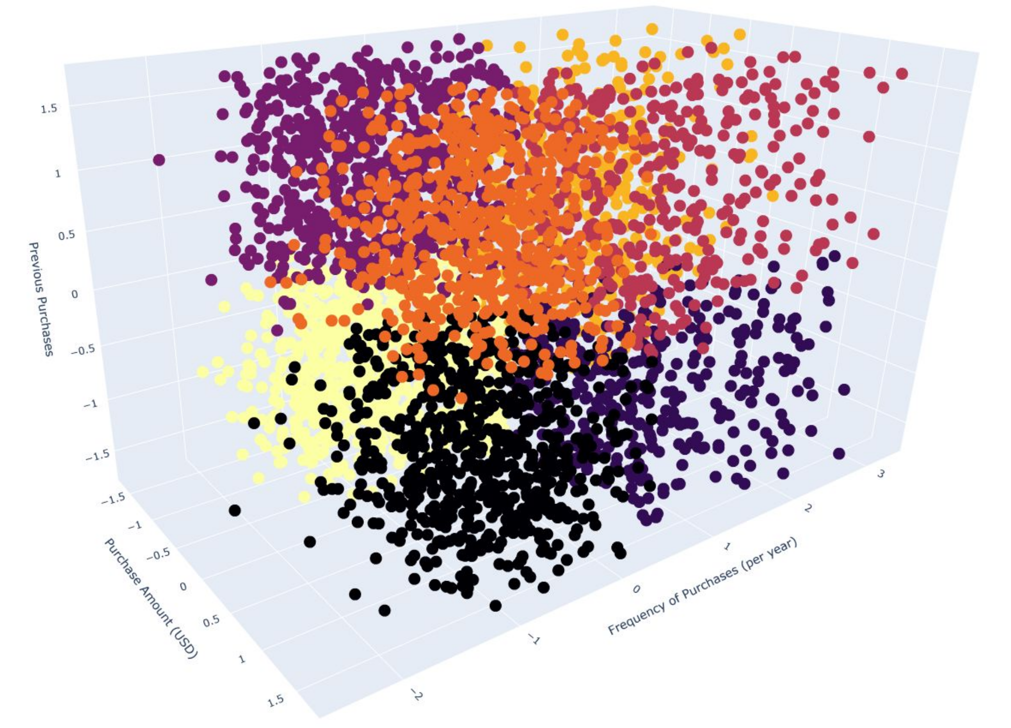
Siloutte Score: 0.39581132379833317 | Davies-Bouldin Score: 0.9594855281768605

k = 4



Siloutte Score: 0.4971803709344656 | Davies-Bouldin Score: 0.8172685461369673

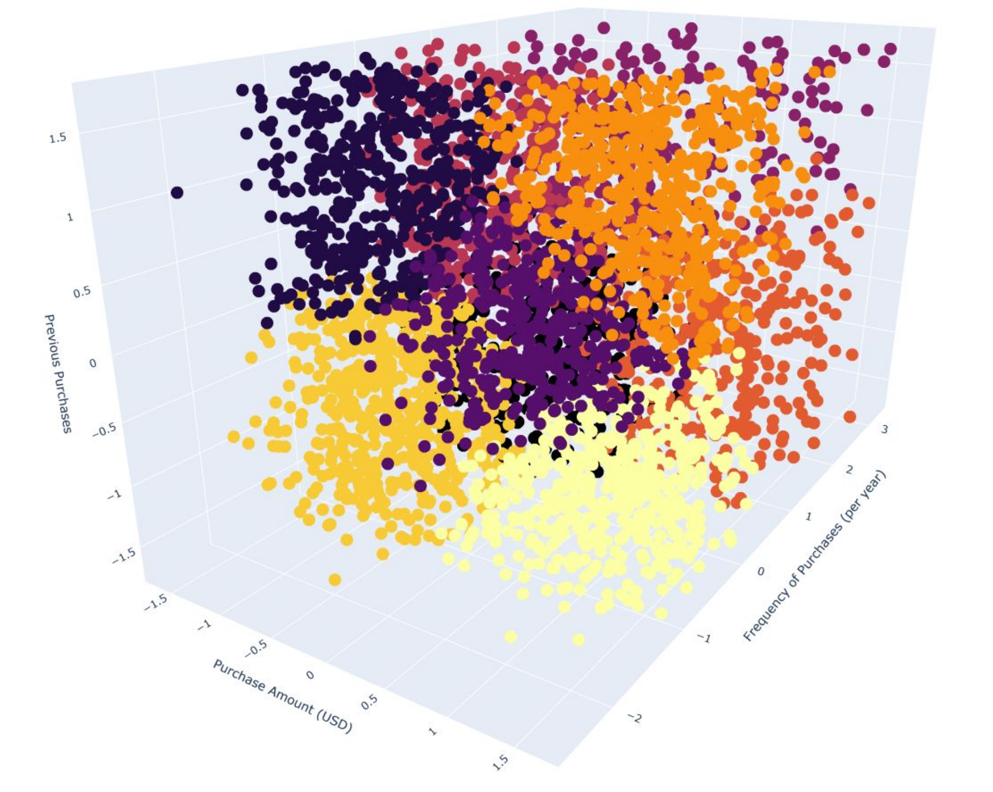
k = 6



Siloutte Score: 0.5113224693143813 | Davies-Bouldin Score: 0.769913505634763

k = 8

When does data become overfit?



Association Rules and Interesting Findings

We used the Apriori algorithm to determine association rules. There were a few methods we used to make the data workable for Apriori:

A). Mapping continuous data attributes to categorical values

B). Convert the new categorical pandas dataframe into a Market Basket data type, so the Apriori algorithm can run on it

C). Running the Apriori algorithm on the dataset

D). Finding Association rules from the frequent itemsets

Association Rules and Interesting Findings

To evaluate our method, we aimed to find association rules for both the whole dataset and for the dataset within the identified most valuable cluster. By finding the difference between the association rules found in the whole dataset and the clustered dataset, we may be able to explain and predict trends in obtaining valuable customers.

We also compared the confidence levels between rules found in both sets, to likewise explain and predict differences between average customers and valuable customers.

Association Rules and Interesting Findings

Script used to find the difference between Association Rules

```javascript
function compareAntecedentsConsequents() {
  var sheet1 = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet4');
  var sheet2 = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet5');
  var sheet3 = SpreadsheetApp.getActiveSpreadsheet().getSheetByName('Sheet6');

  // Clear any existing data in Sheet3 before writing new results
  sheet3.clear();

  var data1 = sheet1.getDataRange().getValues();  // Get data from Sheet1
  var data2 = sheet2.getDataRange().getValues();  // Get data from Sheet2

  var differences = [];

  // Compare each rule in Sheet1 against Sheet2
  for (var i = 0; i < data1.length; i++) {
    var antecedent1 = data1[i][0]; // Assuming antecedent is in column A
    var consequent1 = data1[i][1]; // Assuming consequent is in column B

    var foundInSheet2 = false;

    // Check if antecedent and consequent in Sheet1 exist in Sheet2
    for (var j = 0; j < data2.length; j++) {
      var antecedent2 = data2[j][0]; // Antecedent in Sheet2
      var consequent2 = data2[j][1]; // Consequent in Sheet2

      if (antecedent1 === antecedent2 && consequent1 === consequent2) {
        foundInSheet2 = true;
        break;
      }
    }

    // If no match found, record the rule as "Not Found"
    if (!foundInSheet2) {
      differences.push(['Rule not found in Sheet2: ' + antecedent1 + ' => ' + consequent1]);
    }
  }

  // Now check for rules in Sheet2 not found in Sheet1
  for (var i = 0; i < data2.length; i++) {
    var antecedent2 = data2[i][0]; // Antecedent in Sheet2
    var consequent2 = data2[i][1]; // Consequent in Sheet2

    var foundInSheet1 = false;

    for (var j = 0; j < data1.length; j++) {
      var antecedent1 = data1[j][0]; // Antecedent in Sheet1
      var consequent1 = data1[j][1]; // Consequent in Sheet1

      if (antecedent2 === antecedent1 && consequent2 === consequent1) {
        foundInSheet1 = true;
        break;
      }
    }

    // If no match found, record the rule as "Not Found"
    if (!foundInSheet1) {
      differences.push(['Rule not found in Sheet1: ' + antecedent2 + ' => ' + consequent2]);
    }
  }

  // Output the differences to Sheet3
  if (differences.length > 0) {
    sheet3.getRange(1, 1, differences.length, 1).setValues(differences);  // Write the differences in Sheet3 starting at cell A1
  } else {
    sheet3.getRange(1, 1).setValue('All rules match between Sheet1 and Sheet2');
  }
}
```

Association Rules and Interesting Findings

Rule not found in AllData.35: frozenset({'NoSubscr'}) => frozenset({'M', 'NoDisco'})
Rule not found in AllData.35: frozenset({'NoSubscr'}) => frozenset({'M', 'NoPromo', 'NoDisco'})
Rule not found in AllData.35: frozenset({'NoSubscr'}) => frozenset({'M', 'NoPromo'})
Rule not found in AllData.35: frozenset({'Venmo'}) => frozenset({'Male'})

Interestingly enough, there was also a lot of negative subscription usage associated with valuable data but not with all data. This would imply that the companies promotional offerings and discounts are effective at generating valuable, loyal customers, but the subscription service is inversely effective. If this were a real company, they would likely need to refactor their subscription model.

Rule not found in ValuableData.35: frozenset({'YesDisco', 'M', 'Male'}) => frozenset({'YesPromo'})
Rule not found in ValuableData.35: frozenset({'YesDisco', 'M', 'YesPromo'}) => frozenset({'Male'})
Rule not found in ValuableData.35: frozenset({'YesDisco', 'M'}) => frozenset({'Male'})
Rule not found in ValuableData.35: frozenset({'YesDisco', 'M'}) => frozenset({'YesPromo', 'Male'})
Rule not found in ValuableData.35: frozenset({'YesDisco', 'M'}) => frozenset({'YesPromo'})
Rule not found in ValuableData.35: frozenset({'YesDisco', 'Male', 'YesPromo'}) => frozenset({'M'})
Rule not found in ValuableData.35: frozenset({'YesDisco', 'Male'}) => frozenset({'M'})
Rule not found in ValuableData.35: frozenset({'YesDisco', 'Male'}) => frozenset({'YesPromo', 'M'})

Most Interesting Findings…

antecedents	consequents	antecedent support	consequent support	support	confidence	lift
frozenset({'YesSubscr', 'Male'})	frozenset({'YesDisco'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr', 'Male'})	frozenset({'YesPromo'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr', 'Male'})	frozenset({'YesDisco', 'YesPromo'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr'})	frozenset({'Male'})	0.27	0.68	0.27	1.0	1.4705882352941175
frozenset({'YesSubscr'})	frozenset({'YesDisco'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr'})	frozenset({'YesPromo'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr'})	frozenset({'YesDisco', 'Male'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr'})	frozenset({'YesPromo', 'Male'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr'})	frozenset({'YesDisco', 'YesPromo'})	0.27	0.43	0.27	1.0	2.3255813953488373
frozenset({'YesSubscr'})	frozenset({'YesDisco', 'Male', 'YesPromo'})	0.27	0.43	0.27	1.0	2.3255813953488373

In this dataset, every single Subscriber is male, uses discounts, and promotion codes!

Most Interesting Findings…

antecedents	consequents	antecedent support	consequent support	support	confidence	lift
frozenset({'Female', 'NoSubscr'})	frozenset({'NoDisco'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female', 'NoSubscr'})	frozenset({'NoPromo'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female', 'NoSubscr'})	frozenset({'NoPromo', 'NoDisco'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female'})	frozenset({'NoDisco'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female'})	frozenset({'NoPromo'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female'})	frozenset({'NoSubscr'})	0.32	0.73	0.32	1.0	1.36986301369863
frozenset({'Female'})	frozenset({'NoPromo', 'NoDisco'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female'})	frozenset({'NoDisco', 'NoSubscr'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female'})	frozenset({'NoPromo', 'NoSubscr'})	0.32	0.57	0.32	1.0	1.7543859649122808
frozenset({'Female'})	frozenset({'NoPromo', 'NoDisco', 'NoSubscr'})	0.32	0.57	0.32	1.0	1.7543859649122808

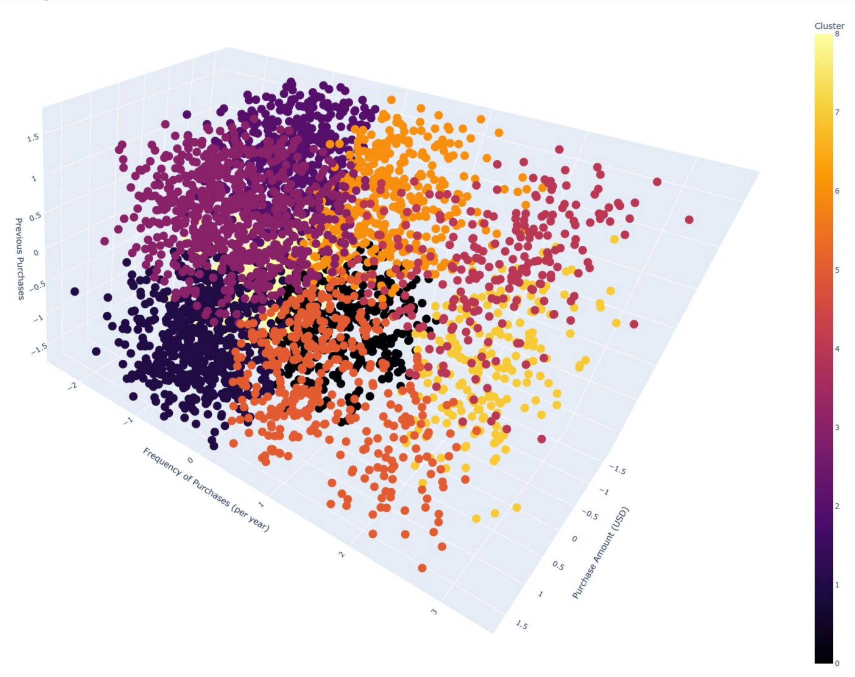
Thus, every female doesn't use discounts, promo codes, and isn't subscribed!

Predicting Most Valuable Customers

antecedents	consequents	antecedent support	consequent support	support	confident	lift
frozenset({'3-4', 'NoDisco', 'NoSubscr'})	frozenset({'NoPromo'})	0.2248939179632249	0.5657708628005658	0.224893917963	1.0	1.7675
frozenset({'20-39'})	frozenset({'NoSubscr'})	0.3620933521923621	0.7114568599717115	0.268741159830	0.7421875	1.0431939612326044
frozenset({'20-39'})	frozenset({'Male'})	0.3620933521923621	0.6916548797736917	0.256011315417	0.70703125	1.0222312755623721
frozenset({'20-39'})	frozenset({'NoDisco'})	0.3620933521923621	0.5657708628005658	0.209335219236	0.578125	1.0218359375
frozenset({'20-39'})	frozenset({'NoPromo'})	0.3620933521923621	0.5657708628005658	0.209335219236	0.578125	1.0218359375
frozenset({'20-39'})	frozenset({'NoPromo', 'NoDisco'})	0.3620933521923621	0.5657708628005658	0.209335219236	0.578125	1.0218359375
frozenset({'20-39'})	frozenset({'NoDisco', 'NoSubscr'})	0.3620933521923621	0.5657708628005658	0.209335219236	0.578125	1.0218359375
frozenset({'20-39'})	frozenset({'NoPromo', 'NoSubscr'})	0.3620933521923621	0.5657708628005658	0.209335219236	0.578125	1.0218359375
frozenset({'20-39'})	frozenset({'NoPromo', 'NoSubscr', 'NoDisco'})	0.3620933521923621	0.5657708628005658	0.209335219236	0.578125	1.0218359375
frozenset({'20-39', 'NoSubscr'})	frozenset({'NoDisco'})	0.26874115983026875	0.5657708628005658	0.209335219236	0.778947368421	1.376789473684210
frozenset({'20-39', 'NoSubscr'})	frozenset({'NoPromo'})	0.26874115983026875	0.5657708628005658	0.209335219236	0.778947368421	1.376789473684210
frozenset({'20-39', 'NoSubscr'})	frozenset({'NoPromo', 'NoDisco'})	0.26874115983026875	0.5657708628005658	0.209335219236	0.778947368421	1.376789473684210
frozenset({'20-39', 'NoSubscr', 'NoDisco'})	frozenset({'NoPromo'})	0.20933521923620935	0.5657708628005658	0.209335219236	1.0	1.7675

Assuming this businesses aims to get people on their subscription model, their target customers would be 100% males.
The valuable cluster dictates that the age range of 20-39 provides the highest value to this business, but the majority of them aren't subscribed, with generally high confidence scores of (0.57-0.78).
Males in this age range should be an advertising priority.
Offering them discounts and promo codes could turn them into subscribers.

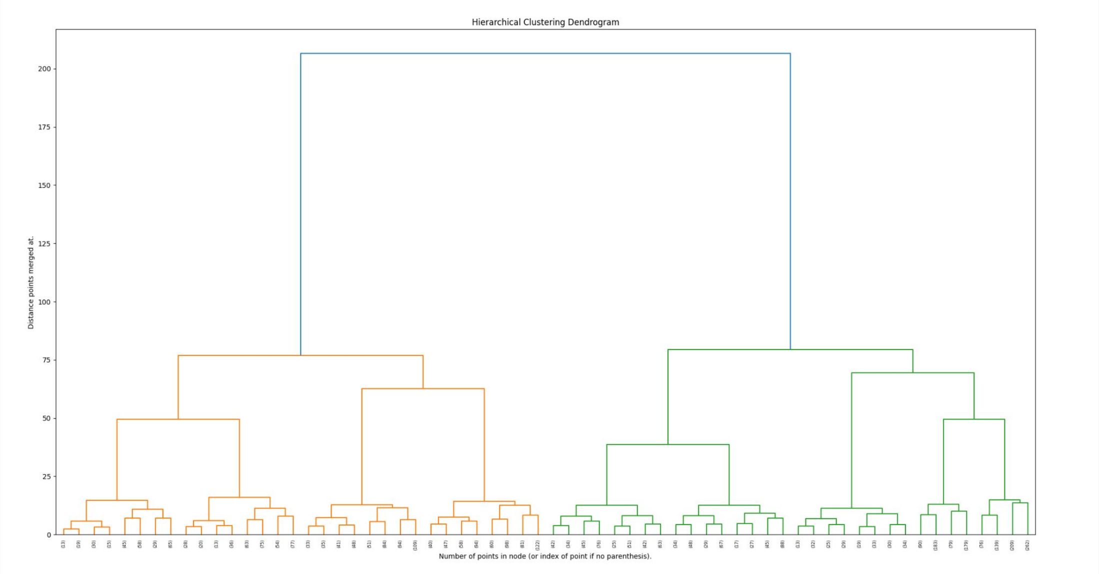
Data Visualization



We chose a high-contrast color ramp to maximize visibility between clusters.

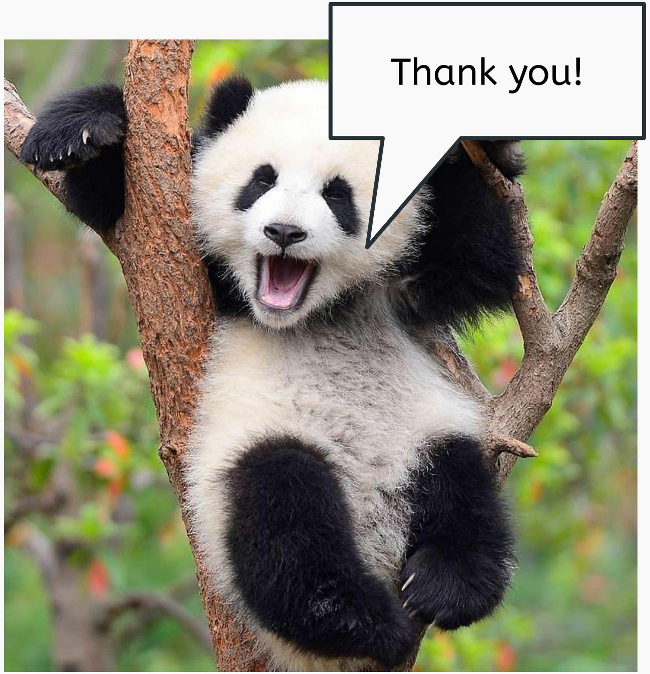
Cluster #4 (coral) had the best results.

Data Visualization



We also utilized the agglomerative clustering method. This dendrogram shows the division of the data, at different indexes

Pandas was invaluable



This project would have been nearly impossible without pandas. Every time we hit an impossible roadblock, pandas would have a library function to get us un-stuck, the most notable of which was dataframe.values.tolist(), which let us keep the apriori algorithm as we intended in our pitch.