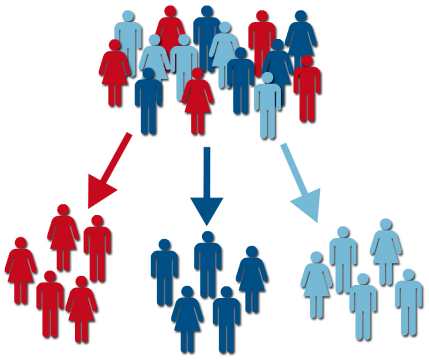
Customer Segmentation

A Data Mining Approach

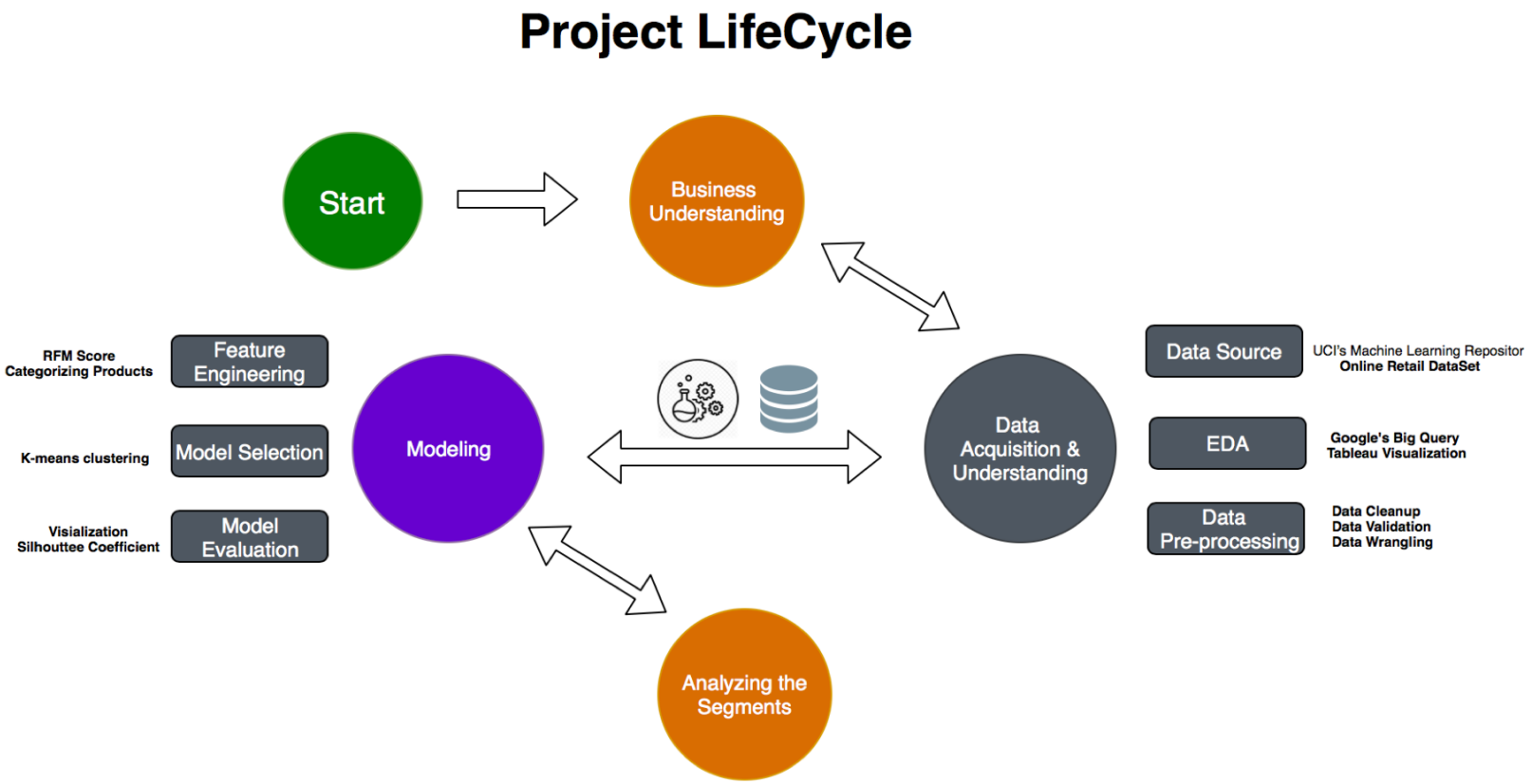
By:
Deepansh Parab

Introduction

- It seems like Customer Segmentation is fundamental to every Business
- It allows business to provide targeted product marketing
- E.g. The 2012 Obama re-election campaign
- Retail Industry is no different
- **Lets see how we can use RFM and clustering analysis for customer segmentation**



Project Flow



Business Understanding

- To be on par with the competition, an UK based online retail company wants to understand their customers buying preferences in order to:
  - pinpoint their marketing strategy
  - deepen customer loyalty

- To achieve the above goals we need to analyze the companies transactional dataset to create segments based on the customers purchase history

Data Understanding

- The dataset is is extracted from UCI's Machine Learning Repository uploaded by **Dr. Daqing Chen**

  Link : https://data.world/uci/online-retail

- The dataset contains transactions occurring between 01/12/2010 and 09/12/2011
- No. of Observation : 541,909
- No. of features: 8
- No. of Transactions: 22,190
- No. of Customers: 4,372
- No. of Products: 3,684

Data Understanding

Features	Description
InvoiceNo	6-digit integral number uniquely assigned to each transaction
StockCode	5-digit integral number uniquely assigned to each
Description	Product name
Quantity	The quantities of each product (item) per transaction
InvoiceDate	The day and time when each transaction was generated
UnitPrice	Product price per unit in sterling £
CustomerID	5-digit integral number uniquely assigned to each customer
Country	The name of the country where each customer resides

Data Exploration

Checking Missing values:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
column type	object	object	object	int64	datetime64[ns]	float64	float64	object
null values (nb)	0	0	1454	0	0	0	135080	0
null values (%)	0	0	0.268311	0	0	0	24.9267	0

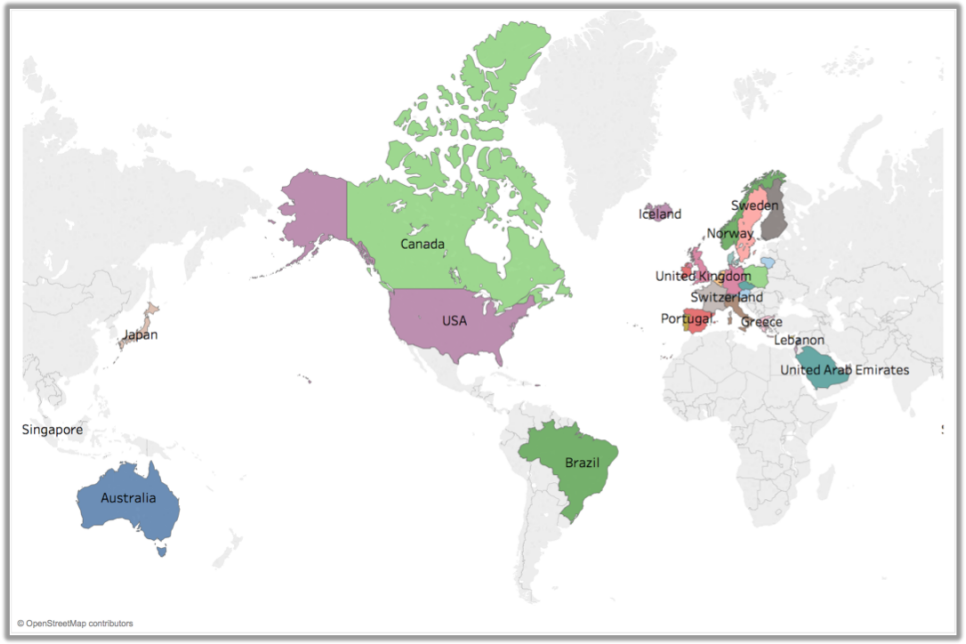
Checking Duplicate values:

```
Duplicate Entries: 5225
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
565	536412	21448	12 DAISY PEGS IN WOOD BOX	2	2010-12-01 11:49:00	1.65	17920.0	United Kingdom
601	536412	21448	12 DAISY PEGS IN WOOD BOX	2	2010-12-01 11:49:00	1.65	17920.0	United Kingdom
604	536412	21448	12 DAISY PEGS IN WOOD BOX	2	2010-12-01 11:49:00	1.65	17920.0	United Kingdom

Feature Exploration

Country:



- **The company covers 37 different countries**
- **Covering all major countries in Europe**
- **Largest customer base is United Kingdom with 3,950 customers which is 90% of the dataset**

Feature Exploration

Cancelled Invoice Number:

	CustomerID	InvoiceNo	Number of products
0	12346.0	541431	1
1	12346.0	C541433	1
2	12747.0	537215	7
3	12747.0	538537	8

- The prefix **C** for the **InvoiceNo** variable: this indicates transactions that have been cancelled
- existence of users who only came once and only purchased one product (e.g. **CustomerID: 12346**) and then cancelled that product
- This customers are not contributing to the companies revenue as they are returning the products that they have previously purchased

Feature Exploration

Negative Quantity:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
61624	C541433	23166	MEDIUM CERAMIC TOP STORAGE JAR	-74215	2011-01-18 10:17:00	1.04	12346.0	United Kingdom
61619	541431	23166	MEDIUM CERAMIC TOP STORAGE JAR	74215	2011-01-18 10:01:00	1.04	12346.0	United Kingdom

```
CustomerID         14527
Quantity              -1
StockCode              D
Description     Discount
UnitPrice           27.5
Name: 141, dtype: object

'---------------> Discounted Products also contribute to Negative Quantity'

154 CustomerID                                  15311
Quantity                                       -1
StockCode                                  35004C
Description    SET OF 3 COLOURED  FLYING DUCKS
UnitPrice                                    4.65
Name: 154, dtype: object
--------------------> HYPOTHESIS NOT FULFILLED
```

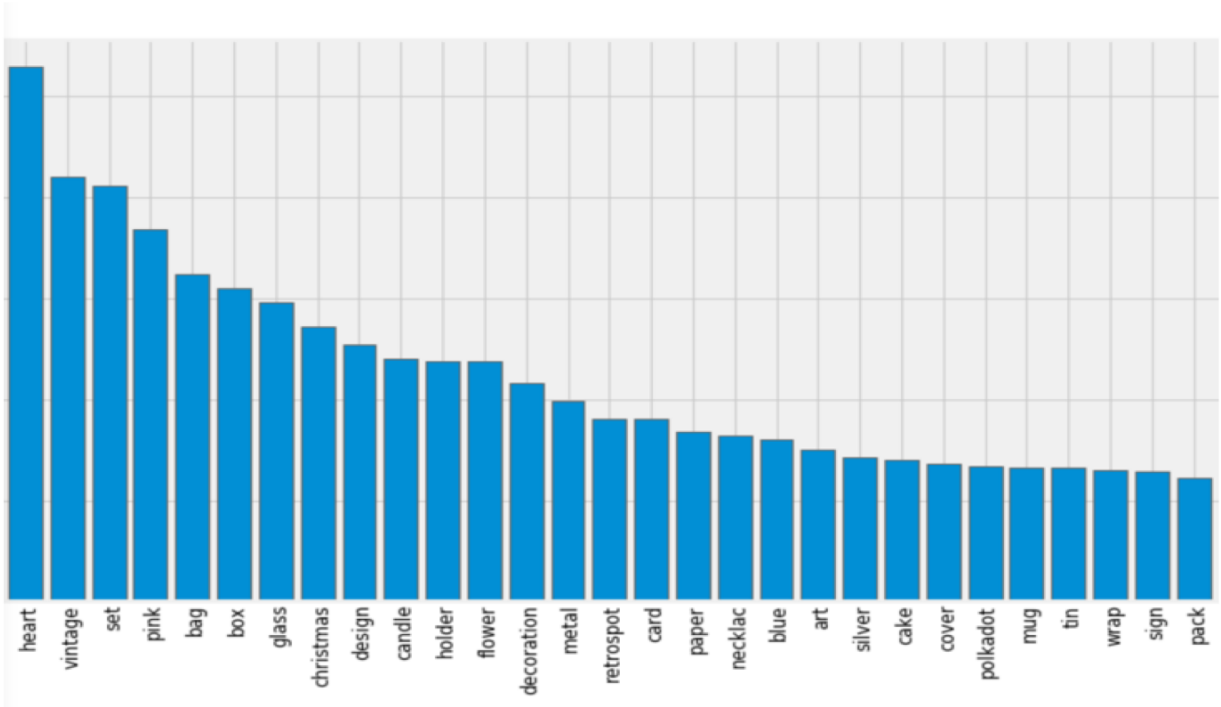
Feature Exploration

Special Stock Code(Expense to company):

```
D               -> Discount
M               -> Manual
BANK CHARGES    -> Bank Charges
POST            -> POSTAGE
C2              -> CARRIAGE
PADS            -> PADS TO MATCH ALL CUSHIONS
DOT             -> DOTCOM POSTAGE
CRUK            -> CRUK Commission
```

Feature Exploration

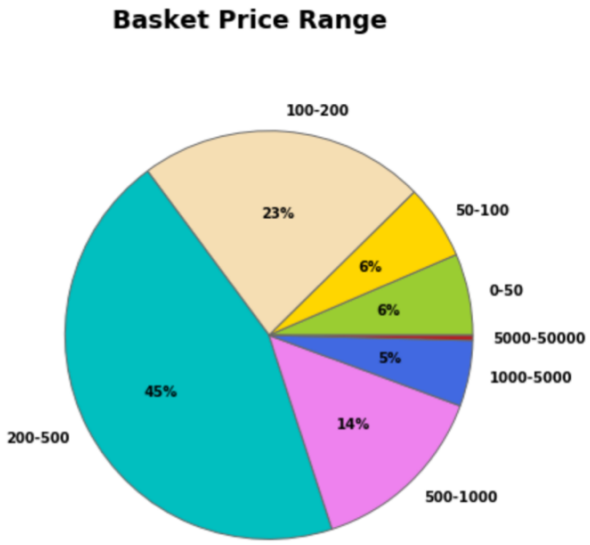
Most Frequent Words in Description:



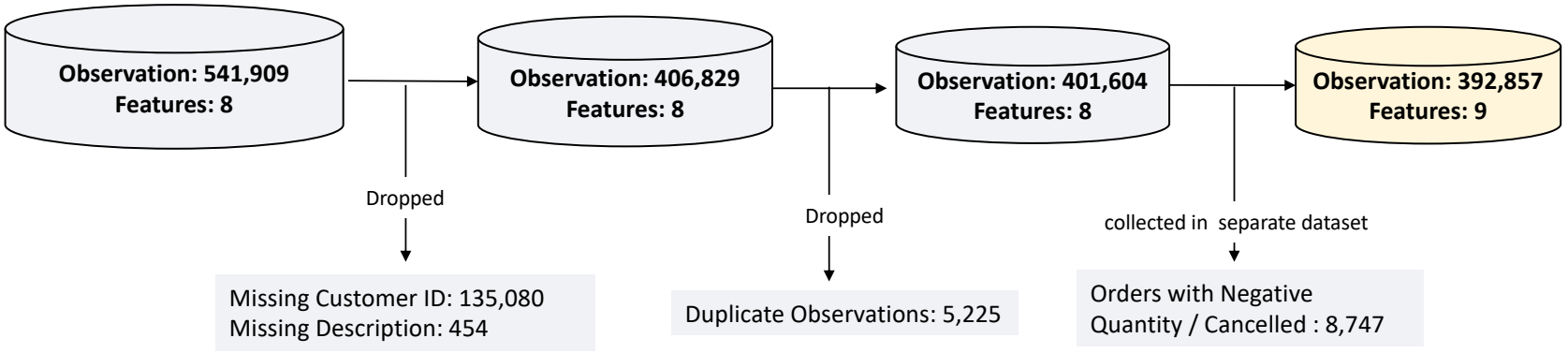
Feature Exploration

Basket Price:

	CustomerID	InvoiceNo	Basket_Price	InvoiceDate
10	12747.0	577104	312.73	2011-11-17 17:13:00
9	12747.0	569397	675.38	2011-10-04 08:26:00
8	12747.0	563949	301.70	2011-08-22 10:38:00
7	12747.0	558265	376.30	2011-06-28 10:06:00



Data Pre-Processing



After the data pre-processing we are left with 392,857 observations and 9 features for our analysis

Feature Engineering

- **Categories for Products:**
  - One - hot encoded data to generate bag of words
  - Included price-range (**UnitPrice**) to generate a balanced group
  - Performed **text – clustering** to group products into the **five** categories

- **RFM Score Calculation:**
  - Calculated Basket Price
  - Calculated Recency, Frequency and Monetary Scores for each customer

Categories for Products:

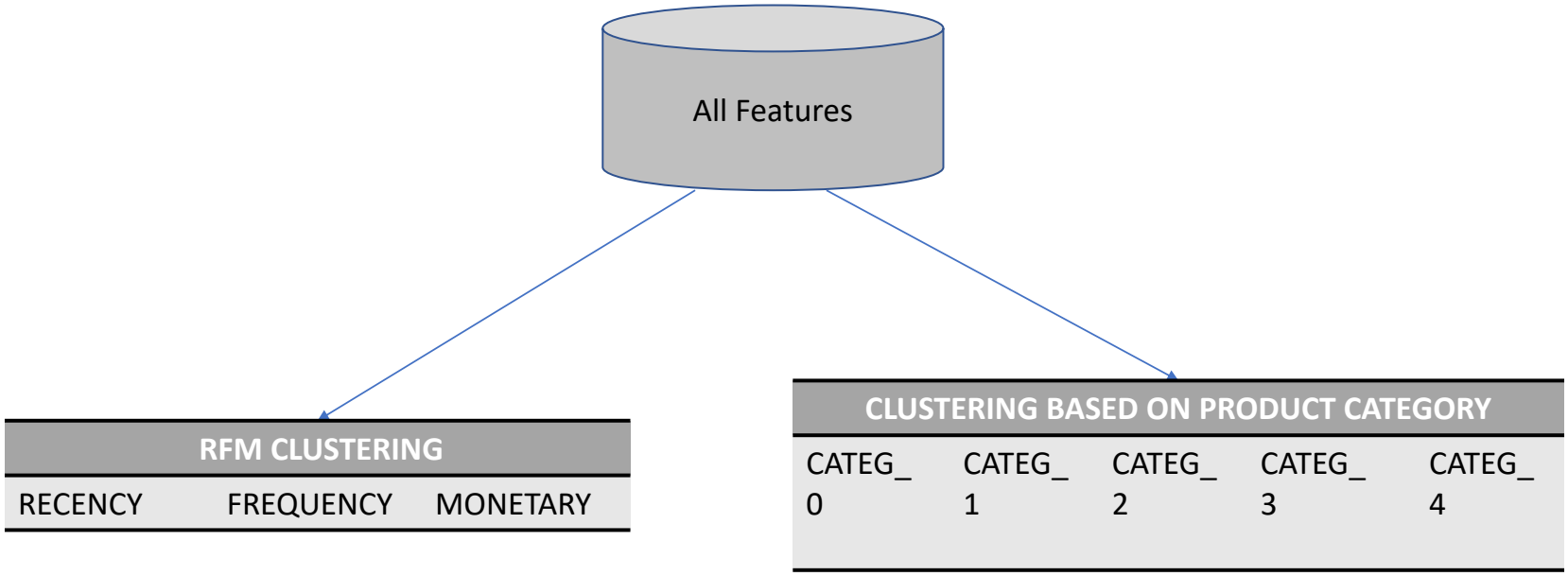
Data Modeling:

K-means Clustering:

- **What is k-means?**
  k-**means** is clustering algorithm that aims to partition **n** observations into **k** clusters such that each observation belongs to the cluster with the nearest **mean**, serving as a prototype of the cluster

- **Why k-means?**
  As our feature variables are numerical and our goal is unsupervised to find out some sort of structure in the customers, I used k-means clustering

Feature Selection for Modeling:



RFM Clustering:

Summary of RFM Data Set:

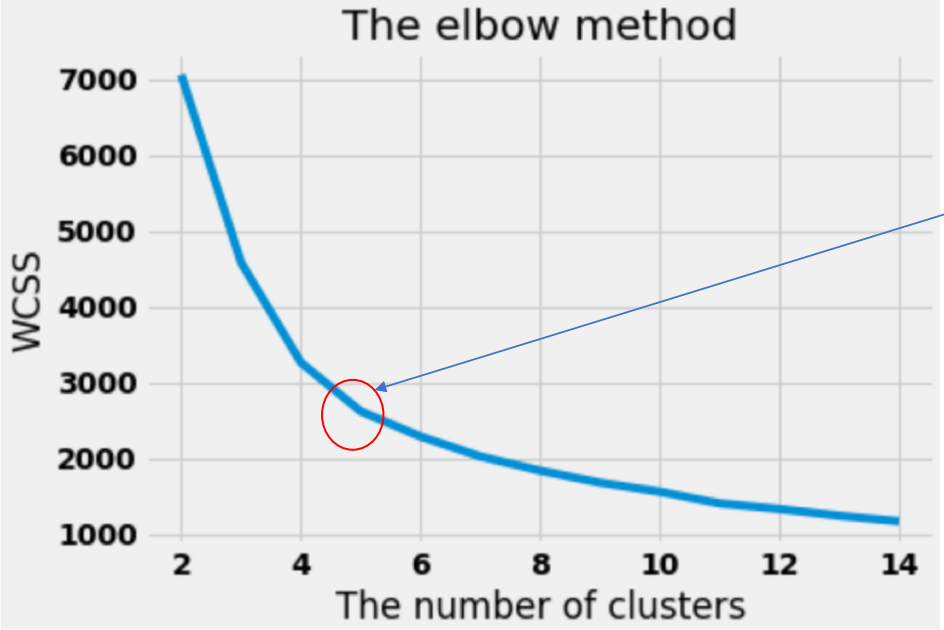
	Recency	Frequency	Monetary
count	3912.000000	3912.000000	3912.000000
mean	91.876534	4.227505	1747.337910
std	99.695909	7.134453	6730.162064
min	0.000000	1.000000	2.900000
25%	17.000000	1.000000	292.007500
50%	50.000000	2.000000	635.070000
75%	143.000000	5.000000	1536.315000
max	373.000000	205.000000	259657.300000

Sqrt. Transformation

	Recency	Frequency	Monetary
count	3912.000000	3912.000000	3912.000000
mean	8.150115	1.793623	32.072545
std	5.045659	1.005326	26.811816
min	0.000000	1.000000	1.702939
25%	4.123106	1.000000	17.088227
50%	7.071068	1.414214	25.200595
75%	11.958261	2.236068	39.195853
max	19.313208	14.317821	509.565796

Data Model:

Selecting value for K



We can see that the sum of squared errors decreases significantly
At K = 5

Modeling:

K-means Model:

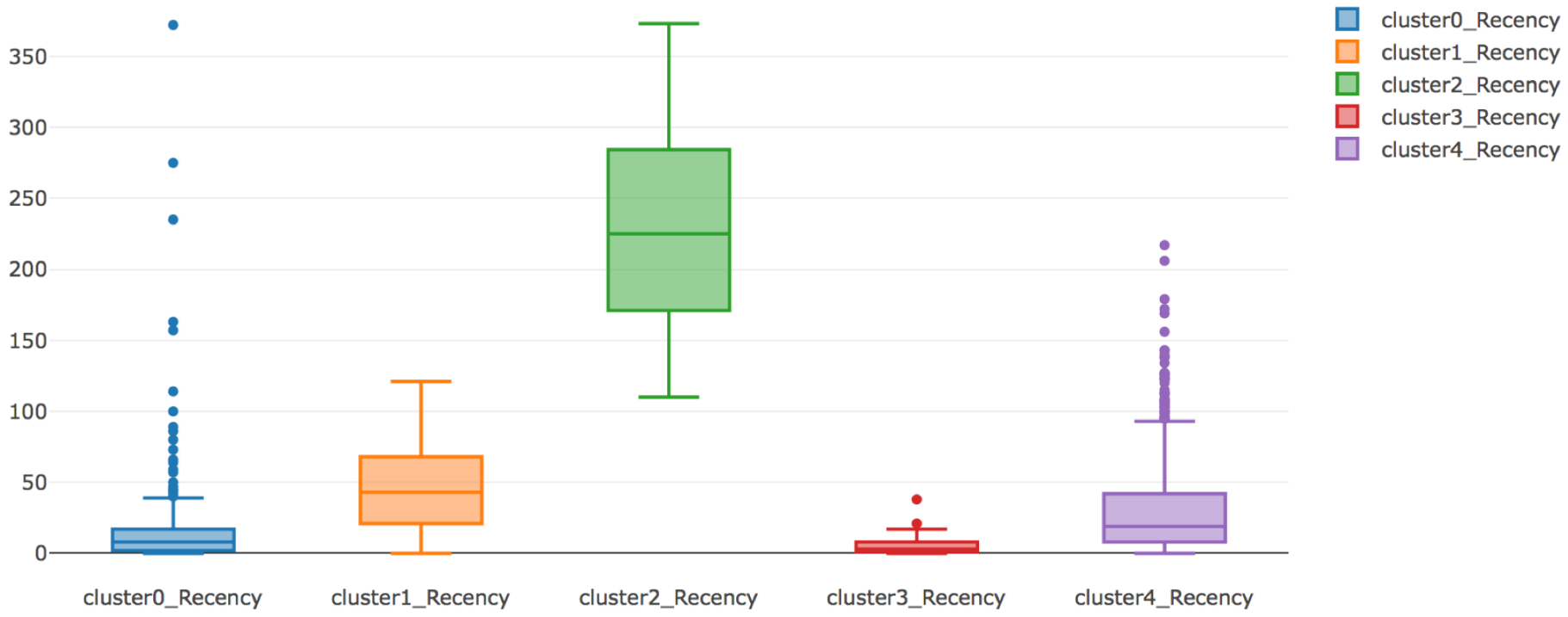
```
n_clusters = 5
kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=100)
kmeans.fit(scaled_matrix)
```

Size of each cluster:

	4	1	0	3	2
nb. of customers	1499	1101	1001	281	30

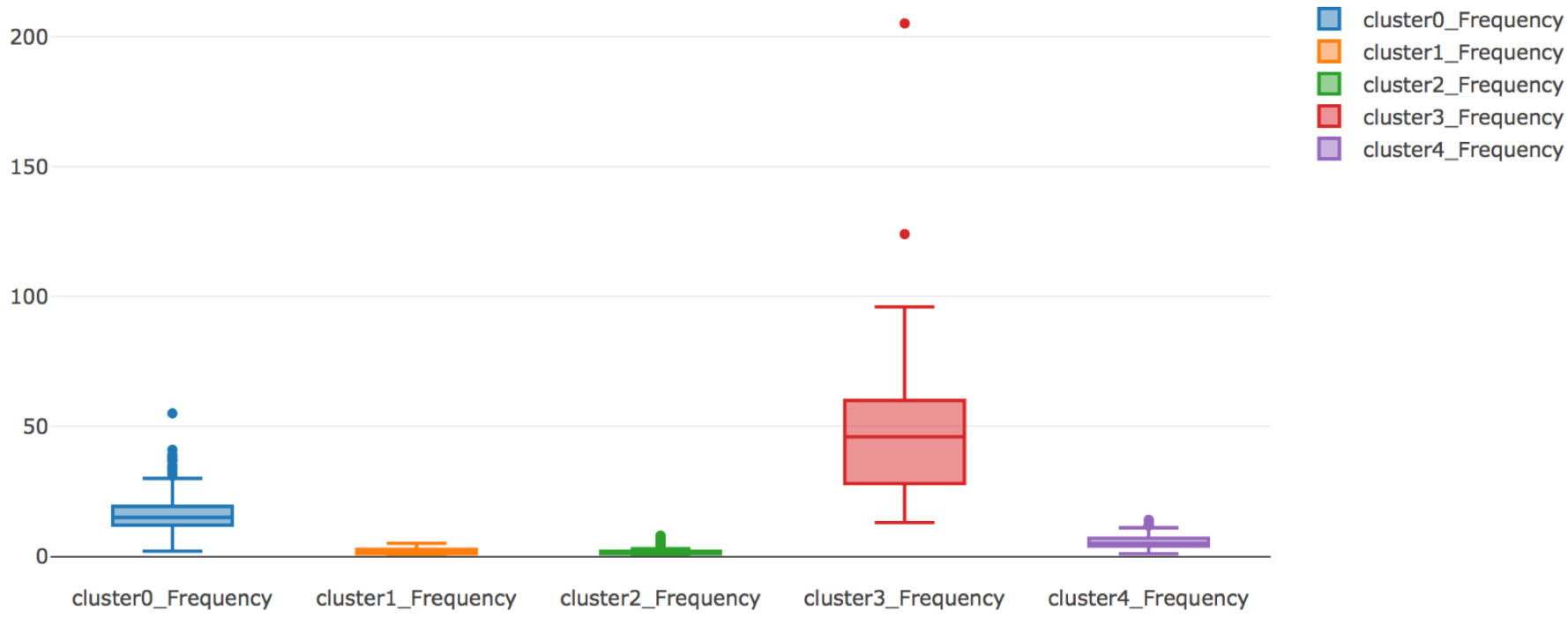
Model Evaluation:

Recency values for the clusters:



Model Evaluation:

Frequency values for the clusters:



Model Evaluation:

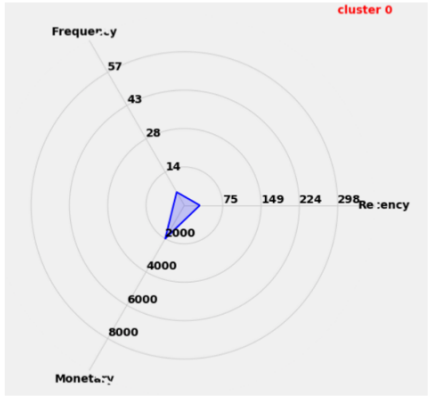
Monetary values for the clusters:

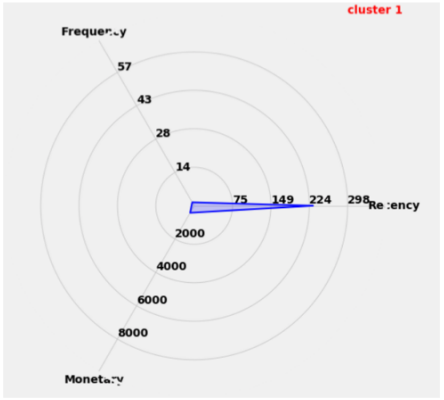
Model Evaluation:

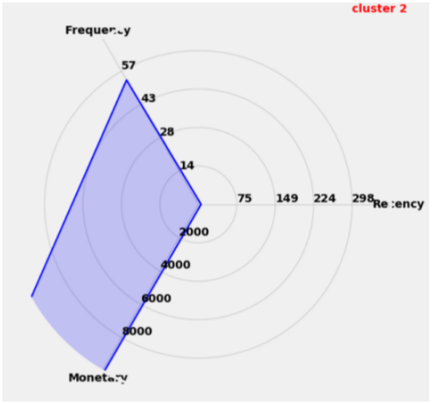
Analyzing Clusters

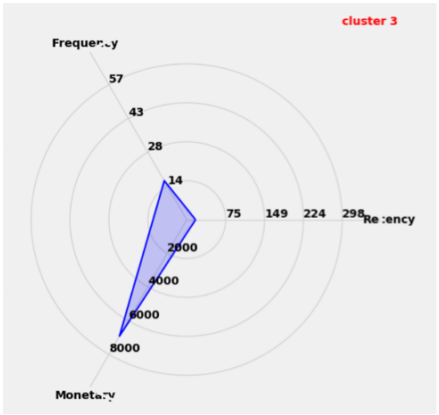
cluster	Recency	Frequency	Monetary	First Purchase	size
1.0	231.806540	1.544959	422.360092	262.998183	1101
4.0	46.038025	1.893929	526.933918	137.575717	1499
0.0	30.181818	5.768232	1996.501329	271.465534	1001
3.0	17.113879	16.483986	6915.607117	337.313167	281
2.0	5.666667	53.066667	54630.335667	353.766667	30

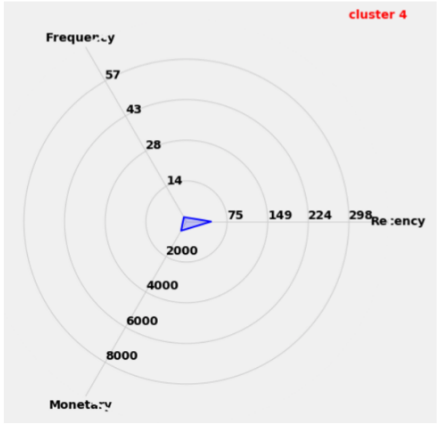
Model Evaluation:











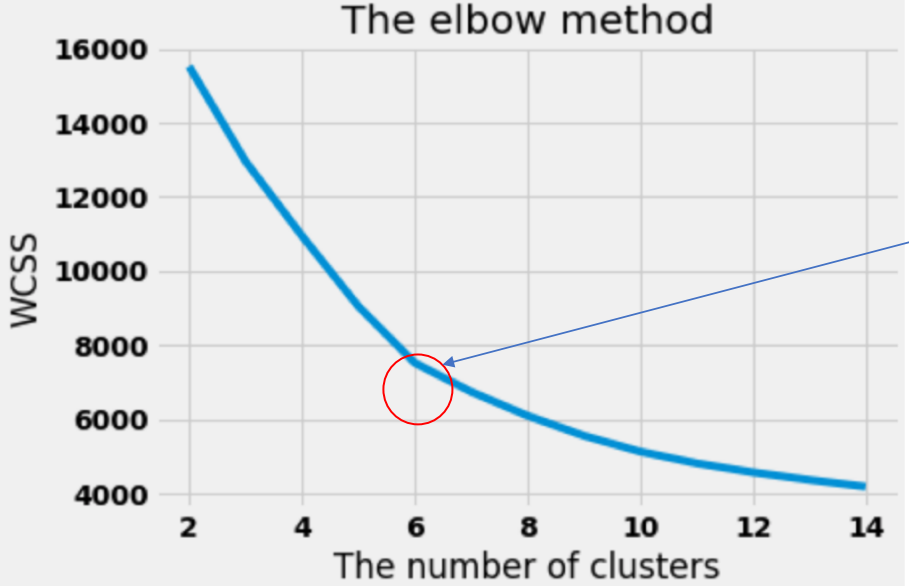
Clustering based on Product Category:

Summary of Product Data Set:

	categ_0	categ_1	categ_2	categ_3	categ_4
count	3912.000000	3912.000000	3912.000000	3912.000000	3912.000000
mean	14.271384	16.619778	21.828790	22.113450	25.173641
std	15.362811	14.764759	16.994285	20.303235	17.012616
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.286077	6.636503	11.368240	7.311255	14.202991
50%	10.507669	14.349445	19.395527	17.705969	23.559591
75%	18.841870	22.737526	28.970929	30.900551	32.993435
max	100.000000	100.000000	100.000000	100.000000	100.000000

Data Model:

Selecting value for K



We can see that the sum of squared errors decreases significantly
At K = 6

Size of each cluster:

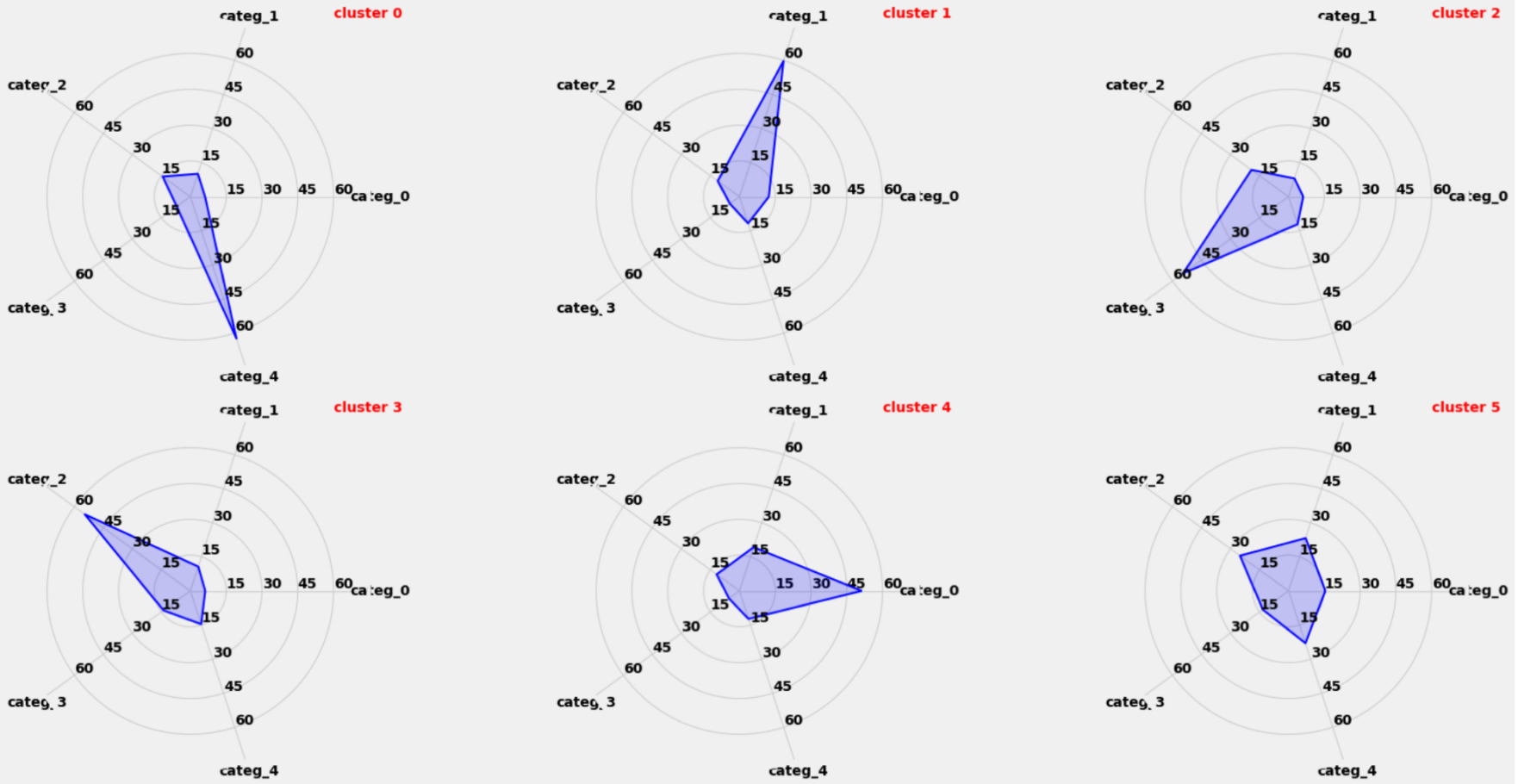
	5	3	1	4	0	2
nb. of customers	1882	576	451	383	323	297

Model Evaluation:

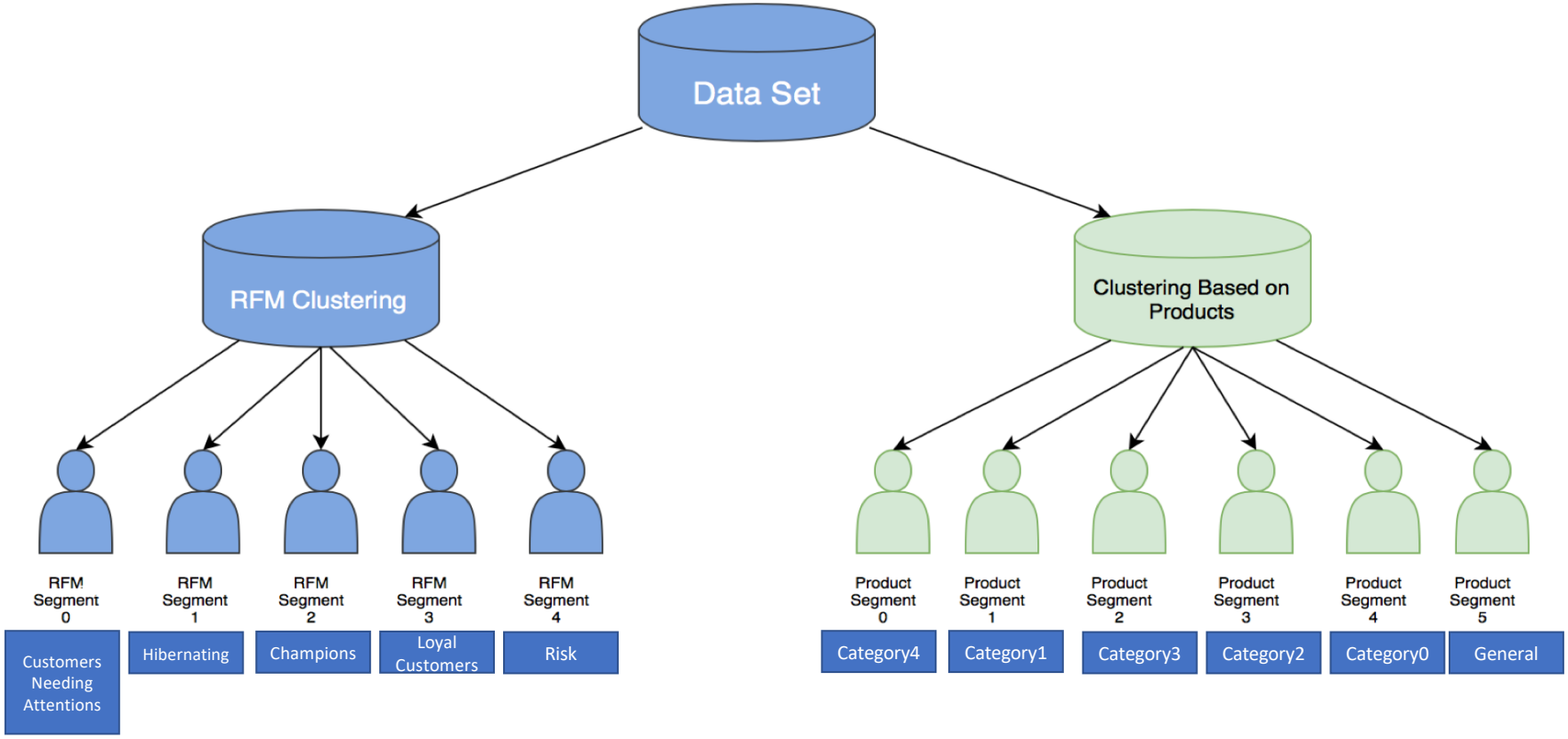
Analyzing Clusters

cluster	Monetary	categ_0	categ_1	categ_2	categ_3	categ_4	size
2.0	740.521279	6.185162	8.028510	19.124808	54.485001	12.176519	297
3.0	1057.384203	6.229217	10.686778	54.608761	13.860865	14.617590	576
1.0	1113.633370	12.320237	59.731109	11.188104	5.009677	11.768375	451
0.0	1924.048390	6.153872	10.104399	14.349232	7.025059	62.367438	323
4.0	1977.558253	51.284830	19.225287	11.754328	5.448736	12.301553	383
5.0	2192.070506	15.441699	23.244465	25.059524	13.309472	22.951303	1882

Model Evaluation:



Analysis of Segments:



Analysis of Segments:

Customer Segment	Activity	Actionable Tip
Champions	Bought recently, buy often and spend the most!	Reward them. Can be early adopters for new products. Will promote your brand.
Loyal Customers	Spend good money with us often. Responsive to promotions.	Upsell higher value products. Ask for reviews. Engage them.
Hibernating	Last purchase was long back, low spenders and low number of orders	Offer other relevant products and special discounts. Recreate brand value
Customers Needing Attention	Above average recency, frequency and monetary values. May not have bought very recently though.	Make limited time offers, Recommend based on past purchases. Reactivate them.
Lost	Lowest recency, frequency and monetary scores.	Revive interest with reach out campaign, ignore otherwise.

Appendix:

- http://github.com/deepansh27/CustomerSegmentation
- https://en.wikipedia.org/wiki/Radar_chart
- http://archive.ics.uci.edu/ml/datasets/Online+Retail
- silhouette analysis on KMeans clustering — scikit-learn 0.19.1 documentation
- KMeans — scikit-learn 0.19.1 documentation
- Suggestions from BBBY's Data Science Team members (Annajium, Roberto, Jonas)
- https://www.putler.com/rfm-analysis/
- Professor Rong Lou
- one hot encoding python sklearn – YouTube