**MEF UNIVERSITY**

**CUSTOMER CLUSTERING WITH MACHINE LEARNING**

**Capstone Project**

**Ömer Faruk Kara**

**ISTANBUL, 2020**

**MEF UNIVERSITY**

**CUSTOMER CLUSTERING WITH MACHINE LEARNING**

**Capstone Project**

**Ömer Faruk Kara**

**Advisor:**

**ISTANBUL, 2020**

**MEF UNIVERSITY**

Name of the project: Customer Clustering with Machine Learning

Name/Last Name of the Student: Ömer Faruk Kara

Date of Thesis Defense: 13/09/2020

14/09/2020

14/09/2020

We hereby state that we have held the graduation examination of \_\_\_\_\_\_\_\_\_\_ and agree that the student has satisfied all requirements.

**THE EXAMINATION COMMITTEE**

Committee Member Signature



# Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University’s ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Name Date Signature

Ömer Faruk Kara 14/09/2020

# EXECUTIVE SUMMARY

CUSTOMER CLUSTERING WITH MACHINE LEARNING

Ömer Faruk Kara

Advisor:

SEPTEMBER 2020

If you are analyzing a company that sells in very different product ranges, you are likely to encounter different types of customers. If you can classify these customers correctly. It can set standard actions while serving them. In this way, it can gain speed and can be improved as standard actions are determined.

**Key Words**: Machine Learning, KMeans Algorithm, Principal Component Analysis (PCA)

# ÖZET

MARKET ALIŞVERİŞÇİ PAZARLAMASI UYGULAMALARI:

SEPET ANALİZİ VE ÖNERİCİ SİSTEM

Ömer Faruk Kara

Tez Danışmanı:

EYLÜL 2020, 17 sayfa

Eğer çok farklı ürün gamlarında satış yapan bir şirketi analiz ediyorsanız, birbirinden farklı tipte müşteriler ile karşılaşmanız olasıdır. Eğer bu müşterileri doğru sınıflandırabilirseniz. Onlara hizmet verirken standart aksiyonlar belirleyebilir. Bu sayede hız kazanabilir ve standart aksiyonlar belirlendiğinden geliştirilebilir.

**Anahtar Kelimeler**: Makina Öğrenmesi, kmeans algoritmasi, PCA

TABLE OF CONTENTS

[Academic Honesty Pledge vi](#_Toc51137303)

[EXECUTIVE SUMMARY vii](#_Toc51137304)

[ÖZET viii](#_Toc51137305)

[1. INTRODUCTION 1](#_Toc51137306)

[Advantages of Customer Segmentation 1](#_Toc51137307)

[2. LITERATURE REVIEW ON CLUSTERING 2](#_Toc51137308)

[What is Kmeans Algorithm? 2](#_Toc51137309)

[What is Silhouette Score on KMeans? 4](#_Toc51137311)

[What is PCA? 5](#_Toc51137312)

[3. THE PROJECT 7](#_Toc51137313)

[Step 0: Company Characteristics and Main Goal 7](#_Toc51137314)

[Step 1: Preparing the data 8](#_Toc51137315)

[Data Set 8](#_Toc51137316)

[Data Preparation 8](#_Toc51137317)

[Step 2: Setting Enviroment 9](#_Toc51137318)

[Step 3: Python Code 10](#_Toc51137319)

[Step 4: Run the code 13](#_Toc51137320)

[Step 5: Results 14](#_Toc51137321)

[APPENDIX 16](#_Toc51137323)

[REFERENCES 17](#_Toc51137324)

# 

.

# INTRODUCTION

In today’s competitive world, it is crucial to understand customer behavior and categorize customers based on their demography and buying behavior. This is a critical aspect of customer segmentation that allows marketers to better tailor their marketing efforts to various audience subsets in terms of promotional, marketing and product development strategies.

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.

In the context of customer segmentation, cluster analysis is the use of a mathematical model to discover groups of similar customers based on finding the smallest variations among customers within each group. These homogeneous groups are known as “customer archetypes” or “personas”.

The goal of cluster analysis in marketing is to accurately segment customers in order to achieve more effective customer marketing via personalization. A common cluster analysis method is a mathematical algorithm known as k-means cluster analysis, sometimes referred to as scientific segmentation. The clusters that result assist in better customer modeling and predictive analytics, and are also are used to target customers with offers and incentives personalized to their wants, needs and preferences.

The process is not based on any predetermined thresholds or rules. Rather, the data itself reveals the customer prototypes that inherently exist within the population of customers.

## Advantages of Customer Segmentation

* Determine appropriate product pricing.
* Develop customized marketing campaigns.
* Design an optimal distribution strategy.
* Choose specific product features for deployment.
* Prioritize new product development efforts.

**Ref: [1] [2] [3]**

# LITERATURE REVIEW ON CLUSTERING

## What is Kmeans Algorithm?

Kmeans algorithm is an iterative algorithm that tries to partition the dataset into Kpre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster’s centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

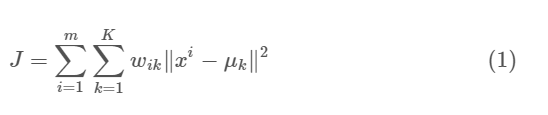
The way kmeans algorithm works is as follows:

1. Specify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.

* Compute the sum of the squared distance between data points and all centroids.
* Assign each data point to the closest cluster (centroid).
* Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

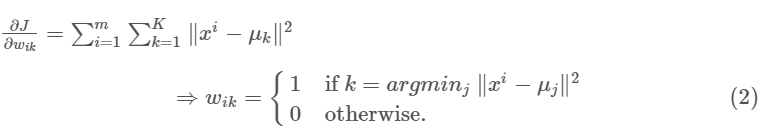
The approach kmeans follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. Below is a break down of how we can solve it mathematically

The objective function is:



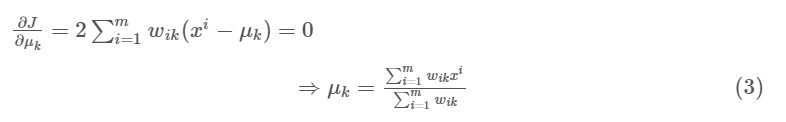
where wik=1 for data point xi if it belongs to cluster k; otherwise, wik=0. Also, μk is the centroid of xi’s cluster.

It’s a minimization problem of two parts. We first minimize J w.r.t. wik and treat μk fixed. Then we minimize J w.r.t. μk and treat wik fixed. Technically speaking, we differentiate J w.r.t. wik first and update cluster assignments (E-step). Then we differentiate J w.r.t. μk and recompute the centroids after the cluster assignments from previous step (M-step). Therefore, E-step is:



In other words, assign the data point xi to the closest cluster judged by its sum of squared distance from cluster’s centroid.

And M-step is:



Which translates to recomputing the centroid of each cluster to reflect the new assignments.

## Ref: [4]

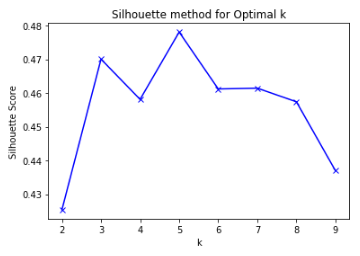
## What is Silhouette Score on KMeans?

This is a better measure to decide the number of clusters to be formulated from the data. It is calculated for each instance and the formula goes like this:

Silhouette Coefficient = (x-y)/ max(x,y)

where, y is the mean intra cluster distance: mean distance to the other instances in the same cluster. x depicts mean nearest cluster distance i.e. mean distance to the instances of the next closest cluster.

The coefficient varies between -1 and 1. A value close to 1 implies that the instance is close to its cluster is a part of the right cluster. Whereas, a value close to -1 means that the value is assigned to the wrong cluster.



Silhouette Method

As per this method k=3 was a local optima, whereas k=5 should be chosen for the number of clusters. This method is better as it makes the decision regarding the optimal number of clusters more meaningful and clear. But this metric is computation expensive as the coefficient is calculated for every instance. Therefore, decision regarding the optimal metric to be chosen for the number of cluster decision is to be made according to the needs of the product.

**Ref: [5]**

## What is Principal Component Analysis PCA?

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

Importantly, the dataset on which PCA technique is to be used must be scaled. The results are also sensitive to the relative scaling. As a layman, it is a method of summarizing data. Imagine some wine bottles on a dining table. Each wine is described by its attributes like colour, strength, age, etc. But redundancy will arise because many of them will measure related properties. So what PCA will do in this case is summarize each wine in the stock with less characteristics.

Intuitively, Principal Component Analysis can supply the user with a lower-dimensional picture, a projection or "shadow" of this object when viewed from its most informative viewpoint.

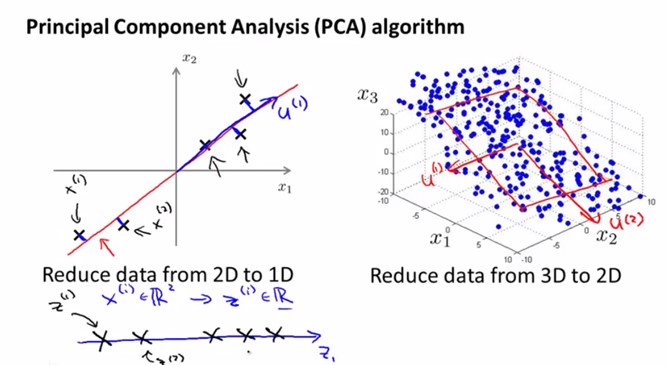


Image Source: Machine Learning Lectures by Prof. Andrew NG at Stanford University

Dimensionality: It is the number of random variables in a dataset or simply the number of features, or rather more simply, the number of columns present in your dataset.

Correlation: It shows how strongly two variable are related to each other. The value of the same ranges for -1 to +1. Positive indicates that when one variable increases, the other increases as well, while negative indicates the other decreases on increasing the former. And the modulus value of indicates the strength of relation.

Orthogonal: Uncorrelated to each other, i.e., correlation between any pair of variables is 0.

Eigenvectors: Eigenvectors and Eigenvalues are in itself a big domain, let’s restrict ourselves to the knowledge of the same which we would require here. So, consider a non-zero vector *v*. It is an eigenvector of a square matrix *A*, if *Av* is a scalar multiple of **v**. Or simply:

*Av = ƛv*

Here, *v* is the eigenvector and *ƛ* is the eigenvalue associated with it.

Covariance Matrix: This matrix consists of the covariances between the pairs of variables. The *(i,j)*th element is the covariance between *i*-th and *j*-th variable.

**Ref: [6]**

# THE PROJECT

## Step 0: Company Characteristics and Main Goal

Our example company is an international company providing hardware and fitting systems and electronic access control systems. It’s customers from the furniture industry, dealers, joiners and cabinet makers, as well as architects, planners and builders.

Our purpose is grouping this companies customers based on their purchases.

But we are not doing a RFM analysis so we don’t want to compare or group them by their purchase size. We want to analyze if there is correlation in their purchased product groups ratio over total purchase.

## Step 1: Preparing the data

## Data Set

Company’s last 12 months invoice data. Columns user are;

CUSTOMER\_NUMBER

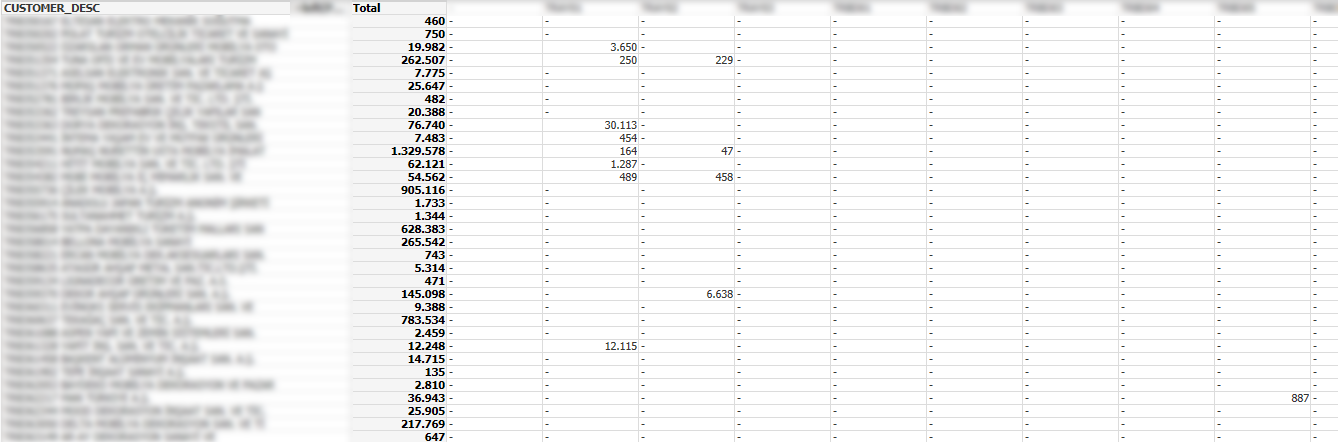
PRODUCT\_GROUP

TOTAL\_INVOCES\_AMOUNT

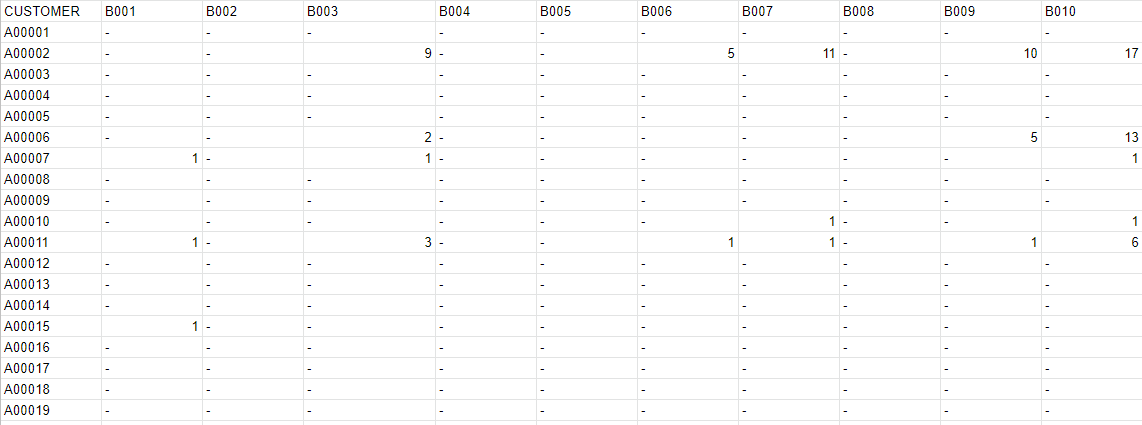
INVOICED\_DATE used for filtering last 12 months.

## Data Preparation

After exporting data from our ERP system, I’m using Qlikview to Pivot data by Customer X Product Group with SUM of Invoiced Amount.



In this analysis we are not interested in Customers purchase capacity so we will use percentage of each product group.



Exported data to CSV to read it easily with pandas framework.

## Step 2: Setting Enviroment

* Install python 3.8

https://www.python.org/downloads/

* Install Pandas

To read our data file we will use pandas library.

pip install pandas

* Install sklearn

We are using KMeans Algorithm for calculate distance between customers. We will use sklearn library for that.

pip install sklearn

* Install matplotlib

For visualization our result, we will use matplotlib library.

pip install matplotlib

## Step 3: Python Code

Data Load and Setting Features of the data

#data load

train = pd.read\_csv("PROJECT\_DATA.csv",";")

#data read

full\_feature\_list=['B001','B002','B003','B004','B005','B006','B007','B008','B009','B010','B011','B012','B013','B014','B015','B016','B017','B018','B019','B020','B021','B022','B023','B024','B025','B026','B027','B028','B029','B030','B031','B032','B033','B034','B035','B036','B037','B038','B039','B040','B041','B042','B043','B044','B045','B046','B047','B048','B049','B050','B051','B052','B053','B054','B055','B056','B057','B058','B059','B060','B061','B062','B063','B064','B065','B066','B067','B068','B069','B070','B071','B072','B073','B074','B075','B076','B077','B078','B079','B080','B081','B082','B083','B084','B085','B086','B087','B088','B089','B090','B091','B092','B093','B094','B095','B096','B097','B098','B099','B100','B101','B102','B103','B104','B105','B106','B107','B108','B109','B110','B111','B112','B113','B114','B115','B116','B117','B118','B119','B120','B121','B122','B123','B124','B125','B126','B127','B128','B129','B130','B131','B132','B133','B134','B135','B136','B137','B138','B139','B140','B141','B142','B143','B144','B145','B146','B147','B148','B149','B150','B151','B152','B153','B154','B155','B156','B157','B158','B159','B160','B161','B162','B163','B164','B165','B166','B167','B168','B169','B170','B171','B172','B173','B174','B175','B176','B177','B178','B179','B180','B181','B182','B183','B184','B185','B186','B187','B188','B189','B190','B191','B192','B193','B194','B195','B196','B197','B198','B199','B200','B201','B202','B203','B204','B205','B206','B207','B208','B209','B210','B211','B212','B213','B214','B215','B216','B217','B218','B219','B220','B221','B222','B223','B224','B225','B226','B227','B228','B229','B230','B231','B232','B233','B234','B235','B236','B237','B238','B239','B240','B241','B242','B243','B244','B245','B246','B247','B248','B249','B250','B251','B252']

selected\_feature\_list = ['B116', 'B098', 'B094', 'B081', 'B121', 'B126', 'B141', 'B159', 'B185', 'B195', 'B196', 'B201', 'B209', 'B211']

feature\_list = full\_feature\_list

a1 = train['CUSTOMER'].values

D=[]

for feature in feature\_list:

    for f in train[feature].values:

        D.append(f)

plt.clf()

D=np.array(D)

customer\_size=len(a1)

feature\_size=len(D)//customer\_size

X = D.reshape((customer\_size,feature\_size))

We don’t have a fixed cluster size and want to determine it. So we will run our KMeans algorithm for cluster size 2 to our defined cluster range and we will store it’s silhouette score to our array. We will store the results in our filesystem.

def write\_file(fn, str):

    with open(now\_str+'/'+fn, "w") as text\_file:

        print(str, file=text\_file)

silhouetteX= []

silhouetteY= []

file\_name='\_featuresize'+str(feature\_size)+'\_'+str(datetime.datetime.now())

file\_name='\_featuresize'+str(feature\_size)

temp\_str=""

#kmeans

x0=datetime.datetime.now()

now\_str=x0.strftime("%Y%m%d%H%M%S")

os.mkdir(now\_str)

cluster\_range=100

for cs in range(cluster\_range)-1:

    x=datetime.datetime.now()

    cluster\_size=cs+2

    kmeans = KMeans(n\_clusters=cluster\_size).fit(X)

    score = silhouette\_score (X, kmeans.fit\_predict(X), metric='euclidean')

    silhouetteX.append(cluster\_size)

    silhouetteY.append(score)

    export=""

    for i in range(len(a1)):

        export+=str(a1[i])+";"+str(kmeans.labels\_[i])+"\n"

    write\_file('export'+file\_name+'\_'+str(cluster\_size)+'.csv', export)

    temp\_str+="Total time = {} | Calculation time= {} | For n\_clusters = {} | silhouette score is {})".format((datetime.datetime.now()-x0),(datetime.datetime.now()-x), cluster\_size, score)+'\n'

We are using Principal Component Analysis (PCA) to show result in two-dimensional graph. Code will store graph images on file system.

#PCA

    pca = PCA(n\_components=2)

    principalComponents = pca.fit\_transform(X)

    principalDf = pd.DataFrame(data = principalComponents

             , columns = ['PC1', 'PC2'])

#    print(principalDf.head())

#    print(principalDf['PC1'].values)

    plt.clf()

    plt.scatter(principalDf['PC1'].values,principalDf['PC2'].values, c=kmeans.labels\_, cmap='rainbow')

    plt.savefig(now\_str+'/PCA\_'+file\_name+'\_'+str(cluster\_size)+'.png')

#/PCA

Logging and store silhouette score graph.

    print("Total time = {} | Calculation time= {} | For n\_clusters = {} | silhouette score is {})".format((datetime.datetime.now()-x0),(datetime.datetime.now()-x), cluster\_size, score))

plt.clf()

#plt.scatter(silhouetteX, silhouetteY)

plt.plot(silhouetteX, silhouetteY, linewidth=3)

#plt.show()

plt.savefig(now\_str+'/silhouette\_'+file\_name+'\_'+str(cluster\_range)+'.png')

write\_file('log\_'+file\_name+'.txt', temp\_str)

## Step 4: Run the code

It takes total 9 minutes 24 seconds. It takes longer when the cluster size increases.

For example for cluster size 4 it takes around 1 seconds but it takes around 9 seconds for 100 clusters. Interesting result is maximum time spent (10 sec.) while calculating for cluster size 88.

Total time = 0:00:01.049263 | Calculation time= 0:00:01.047054 | For n\_clusters = 2 | silhouette score is 0.8226367402696589)

……………………

……………………

Total time = 0:00:13.667354 | Calculation time= 0:00:02.012510 | For n\_clusters = 10 | silhouette score is 0.7822199519312913)

……………………

……………………

Total time = 0:04:34.803758 | Calculation time= 0:00:06.729990 | For n\_clusters = 67 | silhouette score is 0.693718271801207)

……………………

……………………

Total time = 0:07:18.179784 | Calculation time= 0:00:09.550510 | For n\_clusters = 87 | silhouette score is 0.684521772689901)

Total time = 0:07:28.398423 | Calculation time= 0:00:10.211659 | For n\_clusters = 88 | silhouette score is 0.6406413361498463)

Total time = 0:07:36.780997 | Calculation time= 0:00:08.381577 | For n\_clusters = 89 | silhouette score is 0.6144718888012429)

……………………

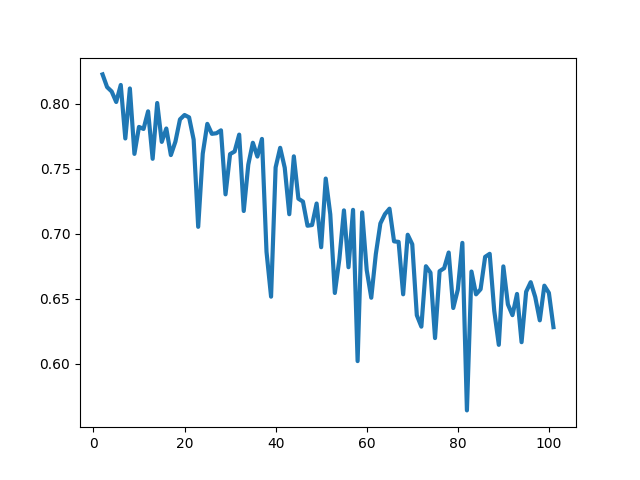
……………………

Total time = 0:09:15.077126 | Calculation time= 0:00:09.337510 | For n\_clusters = 100 | silhouette score is 0.6544412866824009)

Total time = 0:09:24.651828 | Calculation time= 0:00:09.573768 | For n\_clusters = 101 | silhouette score is 0.6280913862214829)

## Step 5: Results

**Silhouette score by cluster size**



Silhouette score is decreasing by cluster size It gives an idea that maximum cluster size must be 20 for our data.

**PCA by cluster size.**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| cluster size 2 |  | cluster size 6 |
|  |  |  |
|  |  |  |
| cluster size 10 |  | cluster size 14 |

# After analyzing the PCA results we can see that large amount of customers are grouped in 2 main groups. Increasing cluster sizes only create customer groups with few outlier customers.

# 

# APPENDIX

The codes and result files are stored in public GitHub repository.

<https://github.com/omerfkara/Capstone>

# REFERENCES

[1] <https://towardsdatascience.com/clustering-algorithms-for-customer-segmentation-af637c6830ac>

[2] <https://towardsdatascience.com/customer-segmentation-using-k-means-clustering-d33964f238c3>

[3] <https://www.optimove.com/resources/learning-center/customer-segmentation-via-cluster-analysis#:~:text=The%20clusters%20that%20result%20assist,their%20wants%2C%20needs%20and%20preferences.&text=Rather%2C%20the%20data%20itself%20reveals,within%20the%20population%20of%20customers>.

[4] <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>

[5] <https://medium.com/@jyotiyadav99111/selecting-optimal-number-of-clusters-in-kmeans-algorithm-silhouette-score-c0d9ebb11308>

[6] <https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial>