# UNSUPERVISED MACHINE LEARNING: CLUSTERING

Khobie Maseko
20/12/2021

# Project Background and Objective

- This data was obtained from a dataset by a Kaggle user named ‘arjunbhasin2013’ titled **C C GENERAL**. The main objective of this project is to cluster the data on this dataset and see which clustering algorithm is best for this purpose.
- The data from this dataset will be used to develop a customer segmentation to define marketing strategy. It summarizes the usage behavior of about 9000 active credit card holders.
- This dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months.
- Dateset source: https://www.kaggle.com/arjunbhasin2013/ccdata

# Data Exploration Plan

The following are the preliminary steps that we will follow to determine which clustering method is the the best one for our dataset :

1. Data Overview
2. Data Cleaning and Feature Engineering:
3. Clustering Method 1: K-Means
4. Clustering Method 2: Hierarchical Agglomerative Clustering
5. Key Findings and Insights
6. Recommendations
7. Conclusion

# Data Overview - Data Attributes

Following is the Data Dictionary for Credit Card dataset :-

CUST*ID* : Identification of Credit Card holder (Categorical)
BALANCE : Balance amount left in their account to make purchases (
BALANCEFREQUENCY : How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated)
PURCHASES : Amount of purchases made from account
ONEOFF*PURCHASES* : Maximum purchase amount done in one-go
INSTALLMENTSPURCHASES : Amount of purchase done in installment
CASH*ADVANCE* : Cash in advance given by the user
PURCHASESFREQUENCY : How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
ONEOFFPURCHASESFREQUENCY : How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased)
PURCHASESINSTALLMENTSFREQUENCY : How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done)
CASHADVANCEFREQUENCY : How frequently the cash in advance being paid
CASHADVANCETRX : Number of Transactions made with "Cash in Advanced"
PURCHASES*TRX* : Numbe of purchase transactions made
CREDITLIMIT : Limit of Credit Card for user
PAYMENTS : Amount of Payment done by user
MINIMUM_PAYMENTS : Minimum amount of payments made by user
PRCFULLPAYMENT : Percent of full payment paid by user
TENURE : Tenure of credit card service for user

# Data Overview - Data Description

```
data.shape
```

```
(8950, 18)
```

```
data.dtypes
```

```
CUST_ID                             object
BALANCE                            float64
BALANCE_FREQUENCY                  float64
PURCHASES                          float64
ONEOFF_PURCHASES                   float64
INSTALLMENTS_PURCHASES             float64
CASH_ADVANCE                       float64
PURCHASES_FREQUENCY                float64
ONEOFF_PURCHASES_FREQUENCY         float64
PURCHASES_INSTALLMENTS_FREQUENCY   float64
CASH_ADVANCE_FREQUENCY             float64
CASH_ADVANCE_TRX                     int64
PURCHASES_TRX                        int64
CREDIT_LIMIT                       float64
PAYMENTS                           float64
MINIMUM_PAYMENTS                   float64
PRC_FULL_PAYMENT                   float64
TENURE                               int64
dtype: object
```

To the left are the descriptive statistics of the dataset.

There are 18 columns and 8950 rows in this dataset.

It also has 3 types of data: int64, float64 and object.

For purposes of this project, we will exclude the object column “CUST_ID” as it is irrelevant to us moving forward.

# Data Cleaning and Feature Engineering

skew_columns

| | |
|---|---|
| MINIMUM_PAYMENTS | 13.622797 |
| ONEOFF_PURCHASES | 10.045083 |
| PURCHASES | 8.144269 |
| INSTALLMENTS_PURCHASES | 7.299120 |
| PAYMENTS | 5.907620 |
| CASH_ADVANCE_TRX | 5.721298 |
| CASH_ADVANCE | 5.166609 |
| PURCHASES_TRX | 4.630655 |
| BALANCE | 2.393386 |
| PRC_FULL_PAYMENT | 1.942820 |
| CASH_ADVANCE_FREQUENCY | 1.828686 |
| ONEOFF_PURCHASES_FREQUENCY | 1.535613 |
| CREDIT_LIMIT | 1.522464 |

dtype: float64

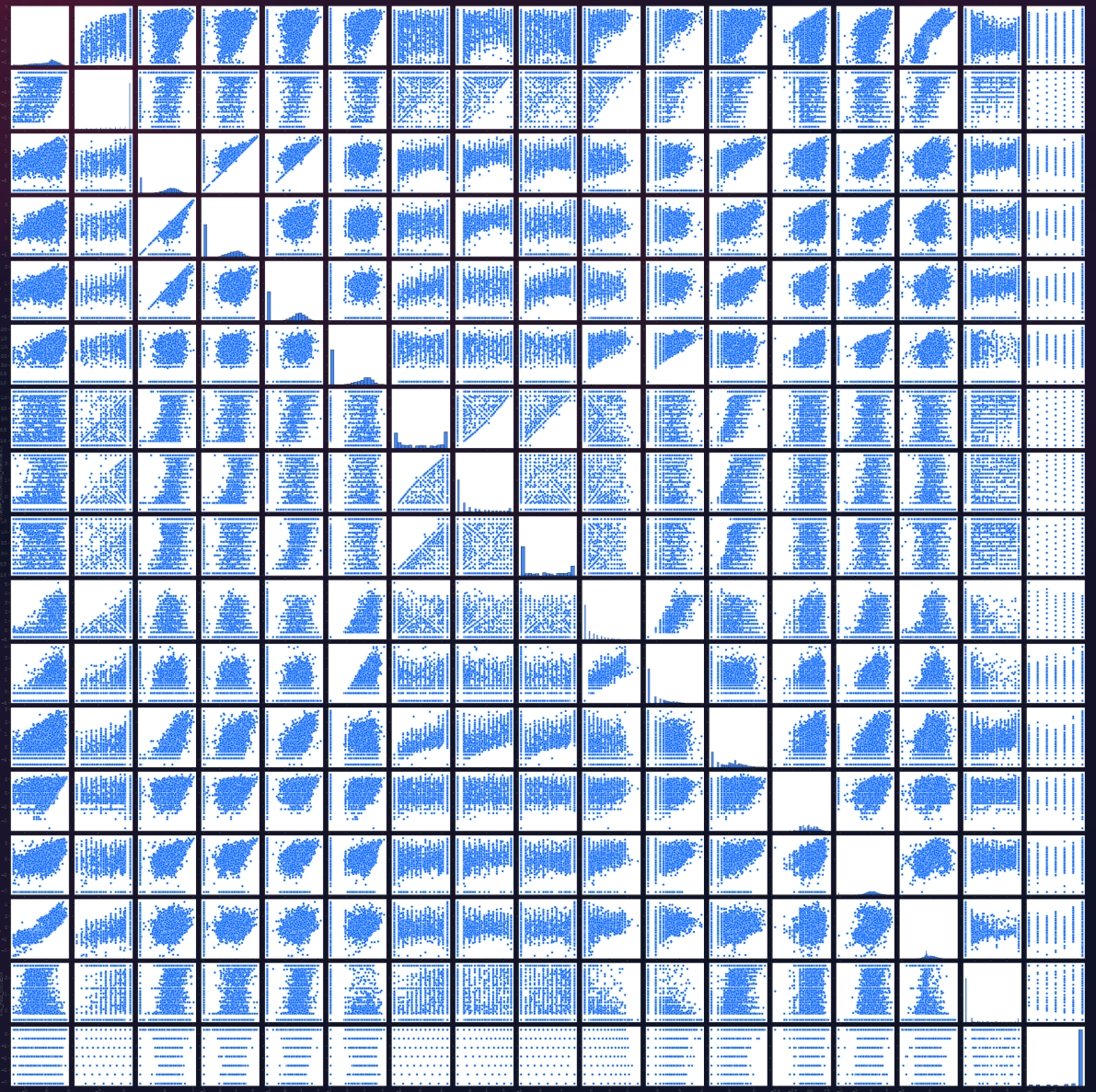
| | |
|---|---|
| BALANCE | CREDIT_LIMIT |
| BALANCE_FREQUENCY | BALANCE |
| PURCHASES | ONEOFF_PURCHASES |
| ONEOFF_PURCHASES | PURCHASES |
| INSTALLMENTS_PURCHASES | PURCHASES |
| CASH_ADVANCE | CASH_ADVANCE_TRX |
| PURCHASES_FREQUENCY | PURCHASES_INSTALLMENTS_FREQUENCY |
| ONEOFF_PURCHASES_FREQUENCY | PURCHASES_TRX |
| PURCHASES_INSTALLMENTS_FREQUENCY | PURCHASES_FREQUENCY |
| CASH_ADVANCE_FREQUENCY | CASH_ADVANCE_TRX |
| CASH_ADVANCE_TRX | CASH_ADVANCE_FREQUENCY |
| PURCHASES_TRX | PURCHASES |
| CREDIT_LIMIT | BALANCE |
| PAYMENTS | PURCHASES |
| MINIMUM_PAYMENTS | BALANCE |
| PRC_FULL_PAYMENT | BALANCE |
| TENURE | CREDIT_LIMIT |

dtype: object

We:

- examined the correlation and skew of all of the variables (except for CUST_ID);
- performed appropriate feature transformations and/or scaling, and
- replaced any NaN values that might have been in the dataset with a “0”.

# Data Cleaning and Feature Engineering

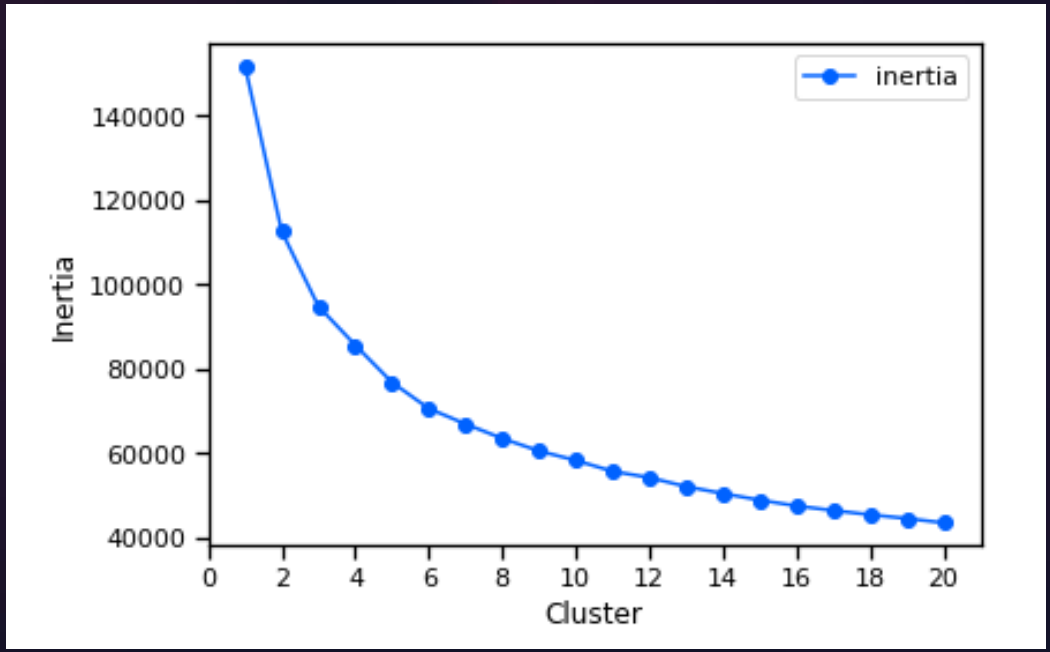


We made a pairplot of the scaled and transformed data.

The pairplot showed us that the scaled and transformed features did not correlate with each other for the most part.

We noted, however, that some features such as “ONCEOFF_PURCHASES” and “PURCHASES” did correlate with each other.

# Clustering Method 1: K-Means

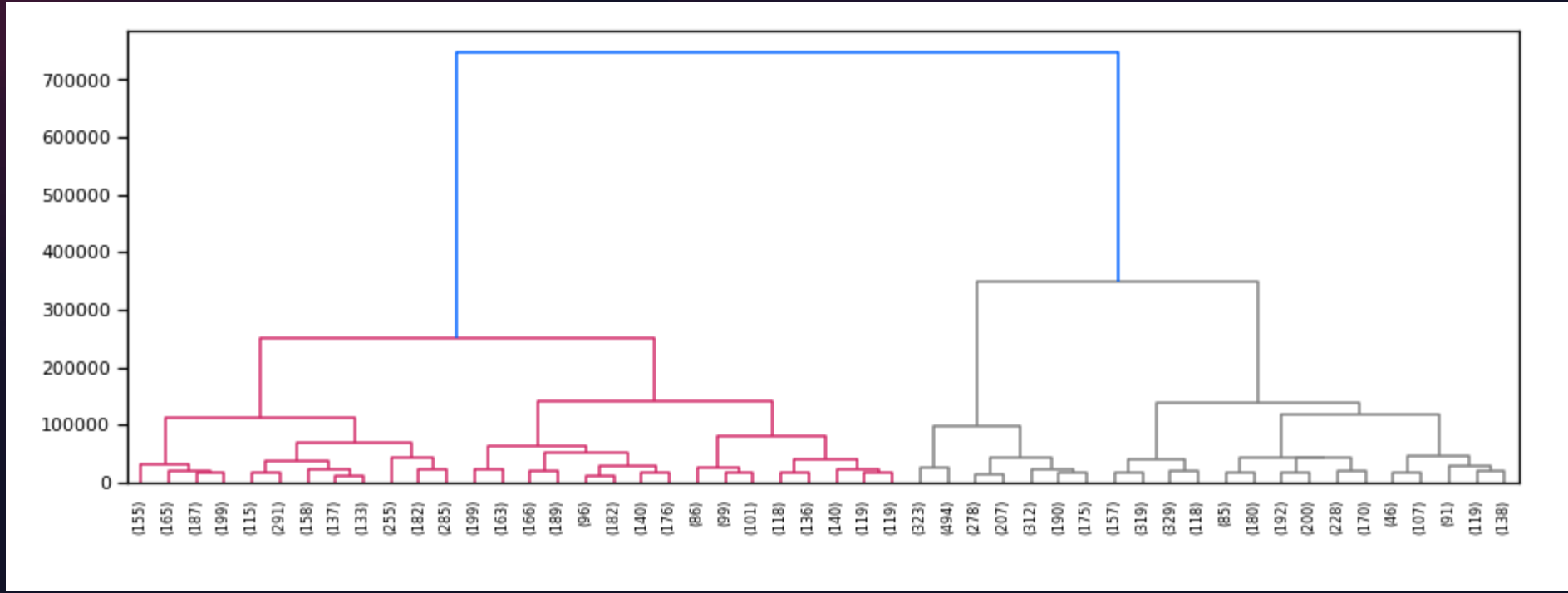



We first fitted a K-means clustering model with 2 clusters.

We then fitted K-Means models with cluster values ranging from 1 to 20 to determine which K-Means value we should use.

We determined that **5** was the best k value (inflection point) by calculating inertia.

# Clustering Method 2: Hierarchical Agglomerative Clustering




We used the Agglomerative Clustering method on our dataset. We then compared the results between this method and the K-Means method. We noted that although the cluster numbers were not identical, the clusters themselves were very consistent.We plotted the above dendrogram which was created from the agglomerative clustering method.

# Key Findings and Insights

| kmeans | TENURE | number |
|---|---|---|
| 0 | -4.122768 | 7 |
| | -3.375526 | 7 |
| | -2.628285 | 12 |
| | -1.881044 | 10 |
| | -1.133803 | 24 |
| | -0.386562 | 55 |
| | 0.360680 | 1777 |
| 1 | -4.122768 | 60 |
| | -3.375526 | 62 |
| | -2.628285 | 59 |
| | -1.881044 | 57 |
| | -1.133803 | 74 |
| | -0.386562 | 140 |
| | 0.360680 | 1873 |
| 2 | -4.122768 | 48 |
| | -3.375526 | 39 |
| | -2.628285 | 54 |
| | -1.881044 | 36 |
| | -1.133803 | 57 |
| | -0.386562 | 59 |
| | 0.360680 | 1175 |
| 3 | -4.122768 | 32 |
| | -3.375526 | 39 |
| | -2.628285 | 34 |
| | -1.881044 | 31 |
| | -1.133803 | 37 |
| | -0.386562 | 52 |
| | 0.360680 | 1188 |
| 4 | -4.122768 | 57 |
| | -3.375526 | 43 |
| | -2.628285 | 37 |
| | -1.881044 | 41 |
| | -1.133803 | 44 |
| | -0.386562 | 59 |
| | 0.360680 | 1571 |

These are the results of running the K-Means method with 5 clusters. We can see that the clusters between the two methods are relatively similar.

| TENURE | agglom | number |
|---|---|---|
| -4.122768 | 0 | 95 |
| | 1 | 109 |
| -3.375526 | 0 | 74 |
| | 1 | 116 |
| -2.628285 | 0 | 85 |
| | 1 | 111 |
| -1.881044 | 0 | 84 |
| | 1 | 91 |
| -1.133803 | 0 | 127 |
| | 1 | 109 |
| -0.386562 | 0 | 204 |
| | 1 | 161 |
| 0.360680 | 0 | 5513 |
| | 1 | 2071 |

These are the results of running the Hierarchical Agglomerative method.

We note that this method has 7 clusters.

This means that the clusters would be smaller than the ones using K-Means and that would be better for developing a customer segmentation to define marketing strategy.

# Recommendations

I would recommend obtaining more data in the form of features to give us more insight for more precise clustering.

# Conclusion

We may conclude that the Hierarchical Agglomerative Cluster model is the method best suited to this dataset. The clusters are smaller, making them easier to manage.

# Jupyter Notebook

The Jupyter Notebook for this project can be viewed at:

https://github.com/KhobieMaseko/IBM-Machine-Learning-Professional-Certificate-/blob/ec9d12b7798020ae8a79502cb09eb670c14a7b1f/Project%204%20Unsupervised%20Machine%20Learning1.ipynb