CREDIT CARD USERS SEGMENTATION USING CLUSTERING TECHNIQUES

A PROJECT REPORT

in partial fulfilment for the award of the degree of

MASTERS OF BUSINESS ADMINISTRATION
in
BUSINESS ANALYTICS

Submitted by
DAMINI THANDELE
FS/MBA-BA/1807

EXTERNAL SUPERVISOR
NAME: Mr. Sachin Singh
Designation: Analytics Manager

INTERNAL SUPERVISOR
NAME: Dr. V. B. Gupta
Designation: Head of Department

SCHOOL OF DATA SCIENCE AND FORECASTING
(UNIVERSITY TEACHING DEPARTMENT)

DEVI AHILYA VISHWAVIDYALAYA
Indore (M.P)

July, 2020

SCHOOL OF DATA SCIENCE AND FORECASTING DEVI AHILYA VISHWAVIDYALAYA INDORE (M.P)

STATEMENT OF ORIGINALITY

In accordance with the requirements for the Degree of Masters of Business Administration in BUSINESS ANALYTICS, in SCHOOL OF DATA SCIENCE AND FORECASTING, I present this report entitled CREDIT CARD USERS SEGMENTATION USING CLUSTERING TECHNIQUES.

This report is completed under the Supervision of:

Mr. Sachin Singh
Designation & Affiliation:
Analytics Manager, CRG Solutions Pvt. Ltd.

Dr. V. B. Gupta
Designation & Affiliation:
Head of Department, SDSF

I declare that the work presented in the report is my own work except as acknowledged in the text and footnotes, and that to my knowledge this material has not been submitted either in whole or in part, for any other degree at this University or at any other such Institution.

Damini Thandele
Name and Signature of the Student
Date:

SCHOOL OF DATA SCIENCE AND FORECASTING DEVI AHILYA VISHWAVIDYALAYA INDORE (M.P)

RECOMMENDATION

This dissertation entitled CREDIT CARD USERS SEGMENTATION USING CLUSTERING TECHNIQUES submitted by DAMINI THANDELE towards the partial fulfilment of Degree of Master of Business Administration in Business Analytics of Devi Ahilya Vishwavidyalaya, Indore is a satisfactory account of his/her project work and is recommended for the award of degree.

Mr. Sachin Singh
External Supervisor

Dr. V. B. Gupta
Internal Supervisor

Dr. V. B. Gupta
Head of Dept.

SCHOOL OF DATA SCIENCE AND FORECASTING
DEVI AHILYA VISHWAVIDYALAYA
INDORE (M.P)

CERTIFICATE

This is to certify that the dissertation entitled "Credit Card Users Segmentation Using Clustering Techniques" submitted by -- Damini Thandele --- is approved for the award of Masters of Business Administration in Business Analytics.

Dr. V. B. Gupta
INTERAL EXAMINER
DATE:

EXTERNAL EXAMINER
DATE:

ACKNOWLEDGEMENT

This project report titled “Credit Card Users Segmentation Using Clustering Techniques” is the result of my 6 months internship at CRG Solutions Pvt Ltd, Mumbai. I am immensely grateful for receiving the opportunity of working on this project and wish to present it as my final semester Major Project for the Degree of Masters of Business Administration in Business Analytics in School of Data Science and Forecasting, DAVV, Indore.

I would like to express my sincere gratitude to the Head of Department (SDSF), who is also my Internal Supervisor, Dr. V. B. Gupta, and to my External Supervisor, Mr. Sachin Singh, for providing their invaluable guidance, comments and suggestions throughout the course of the project. I would also like to thank all the faculty members at SDSF as well, for providing me the knowledge which served as a major contributor towards the progress of the project.

Lastly, I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance that has been helpful in various phases of the completion of the project.

Damini Thandele

Name of the Student with Signature

CERTIFICATE
OF INTERNSHIP

CRG SOLUTIONS

THIS CERTIFICATE IS PRESENTED TO

Damini Thandele

For Successfully Completing the internship at CRG Solutions

From 2nd January'2020 to 31st March'2020

27th August'2020

Date

Himanshu Sethi

Business Head- Services

TABLE OF CONTENTS

ABSTRACT

Customer segmentation is key to customer retention, and it's well known in the marketing biz that it's cheaper to hang on to a current customer than it is to bring in a new one. Customer segmentation also makes marketing cost-effective by targeting relevant audience. Since credit card companies usually have a high marketing budget, they must follow a personalized approach. The benefits of this approach are many, one of those is the fact that highly predictable customer groups make it easier to forecast card profitability.

In the project I've used an unsupervised learning algorithm – clustering, in order to form the segments (or clusters) of customers that show similar behavioral properties. I started off with basic exploratory data analysis, then performed some data preprocessing in order to gain further insights in the data by deriving some KPIs. I also handled the missing data through imputation. Due to the large number of features, in order to deal with the 'Curse of Dimensionality', I used a feature extraction technique known as Principal Component Analysis. After choosing the optimal number of components (with the help of a scree plot), I moved on to perform the clustering.

For the purpose of clustering I used three different algorithms – K-means, K-medoids and DBSCAN. Finally, I displayed the output of all the different clustering algorithms with the help of 2D scatter plots, cross tables and bar graphs in an attempt to try and distinguish the clusters from one another. In the end, I compared the algorithms on an objective level using Silhouette Score and chose the one that got the largest score.

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

It's well documented that about 65% to 75% of new products fail or miss their revenue target? With much research in the issue it's found that it's because the companies fail to understand what their customers truly want and instead follow a one-size-fits-all approach. It's high time for companies to get to know their customers really well. Now, for the purpose of our project, the bank can potentially have millions of customers. Does it make sense to look at the details of each customer separately and then make a decision? Certainly not! It is a manual process and will take a huge amount of time. So what can the bank do? One option is to segment its customers into different groups. (Sharma G. , 2018)

The purpose of the project is to segment credit card users into different groups in order to better serve the marketing efforts, like giving credit card offers, by targeting the group of users with a particular marketing campaign who are more likely to respond to it. Customer segmentation also improves customer service and assists in customer loyalty and retention. We'll first start by some basic data exploration, then to some advanced data analysis. Once our data preparation is done, we will use a feature extraction method to reduce the dimensionality of the data. Finally, we will build our clustering models using distance-based methods such as k-means, k-medoids and DBSCAN.

At the end of the project we'll see that on the basis of a measure called Silhouette Score, we can say that DBSCAN performed the best among all the different clustering algorithms and that the customers can be divided into 4 segments, or clusters, with their distinctly defining characteristics.

CREDIT CARD MARKET

Credit card is an easy-to-use instrument that allows an individual to take credit from a banking institute as and when required. It provides a revolving line of credit which helps a user to take credit multiple times from the institute. The debt is repaid periodically and can be borrowed again once it is repaid. (Revolving credit, 2020)

Key Terminologies

1. **Balance** – It is the amount that is due to be paid to the credit card company. It is based on purchases, interest charges, and any additional fees.
2. **Minimum Payment** – It is the minimum amount of payment that is required from the user to help avoid a 'late payment fee'.
3. **Grace Period** – It is also known as the 'zero-interest period' which is usually 3 weeks long. It's the amount of time you have to pay your balance in full without incurring a finance charge.
4. **Statement Date** – It is the date when your bill for the last billing cycle arrives.
5. **Due Date** – It is the date when your bill payment is due. It is set at the end of the grace period.
6. **Annual Percentage Rate (APR)** – It is the interest charged for a whole year.
7. **Daily Periodic Rate (DPR)** – Credit card issuers calculate interest on a daily basis. DPR is the APR divided by 365.
8. **Billing Cycle** – It is the time frame between two bills raised and usually ranges from 28 to 31 days.
9. **Average Daily Balance** – It is the sum of the daily balances held by a credit card user over a billing cycle divided by the number of days in the cycle. It is used in the calculation of interest.
10. **Cash Advance** – It is the service provided by credit card issuers which allows a user to withdraw a sum of cash from an ATM or a bank. It accrues an additional 'Cash Advance Fee'.
11. **Credit Limit** – It is the maximum amount of credit that a credit card issuer extends to a user.

How Credit Cards Work

A credit card user makes a series of purchases in a particular billing cycle and gets the statement of their balance on the statement date which is to be paid by the end of the grace period. If the user manages to pay the entire due balance before the end of the grace period, they will not be required to pay a single penny in interest for getting the debt. (Irby, 2020)

In the balance statement, the user also receives information about the minimum amount due for the given billing cycle. The minimum due amount is generally about 5 percent of the total amount of the bill. Paying up the minimum amount due can help the user avoid paying the late payment fee.

But paying only the minimum amount does not mean that the interest on the outstanding bill amount will be waived off. Depending on the kind of credit card the user has, they may be charged anywhere from 3 to 4 percent per month on their credit card bill which remains unpaid at the end of the credit free period. Interest is levied on the entire outstanding amount until they make the complete payment of their credit card bill. (Maji, 2020)

Also, if the user doesn't even pay the full minimum due amount, any unpaid minimum due amount from previous bills also get added to their current minimum due amount.

Lastly, other fees such as the cash advance fee are also charged on the credit card.

Interest Calculation

$$Interest = Average\ Daily\ Balance\ X\ DPR\ X\ No.\ of\ Days\ in\ Billing\ Cycle$$

(Credit Card Interest Rates, 2020)

Let's say the APR is 23.99%.

Therefore, in order to calculate the DPR, we must divide the APR by 365. Hence, the DPR is 23.99 % / 365 = 0.065726 %.

Now, let's take for example a simple scenario wherein there's a carry over balance of $700 at the beginning of a billing cycle, and no other purchase is made in the cycle. In this case,

$$\text{Average Daily Balance} = (\$700 \text{ X } 31) / 31 = \$700$$

Therefore,

$$\text{Interest} = \$700 \text{ X } 0.00065726 \text{ X } 31 = \mathbf{\$14.26}$$

But in real life the scenario is quite different. Let's take another example.

The table below displays the details of transactions for a given period cycle starting at 01/07/2019 and ending at 31/07/2019. Note that there is a carried over (unpaid) amount at the starting of the billing cycle (from the previous billing cycle). The due date of the previous billing cycle is 26/07/2019.

Let's calculate the interest.

Date	**Transaction**	**Transaction Amt**	**Cumm. Amt**
01/07/2019	$700.00	Carry over	$700.00
06//07/2019	$69.45	Grocery	$769.45
12/07/2019	$21.05	Gas	$790.50
17/07/2019	$20.51	Restaurant	$811.01
21/07/2019	$41.00	Gift purchase	$852.01
26/07/2019	-$300.00	Bill payment	$552.01

Table 1: Example Transaction Data

In order to calculate the Average Daily Balance, we need to calculate the sum of daily balances for the entire billing cycle. The user kept a steady balance of $700.00 for 5 days. Therefore, the sum of the balance for the 5 days is,

$$\$700.00 \ X \ 5 \ = \ \$3500.00$$

Then he purchased grocery items worth $69.45, taking his balance to $769.45. This balance remains steady for 6 days, and so on. So,

$$\$769.45 \ X \ 6 \ = \ \$4616.70$$

$$\$790.50 \ X \ 5 \ = \ \$3952.50$$

$811.01 X 4 = $3244.04

$852.01 X 5 = $4260.05

$552.01 X 6 = $3312.06

Adding up all the daily balances, we get,

$3500.00 + $4616.70 + $3952.50 + $3244.04 + $4260.05 + $3312.06 + = $22885.35

And thus,

Average Daily Balance = $22885.35 / 31 = $738.24

Now we can calculate the interest for the given billing cycle.

Interest = $738.24 X. 0.00065726 X 31 = **$15.04**

EXPLORING DATA

The data contains 18 columns and 8950 rows. Let's see the first 7 rows.

CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES
C10001	40.900749	0.818182	95.4	0	95.4
C10002	3202.46742	0.909091	0	0	0
C10003	2495.14886	1	773.17	773.17	0

C10004	1666.67054	0.636364	1499	1499	0
C10005	817.714335	1	16	16	0
C10006	1809.82875	1	1333.28	0	1333.28
C10007	627.260806	1	7091.01	6402.63	688.38

CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX
0	0.166667	0	0.083333	0	0
6442.94548	0	0	0	0.25	4
0	1	1	0	0	0
205.788017	0.083333	0.083333	0	0.083333	1
0	0.083333	0.083333	0	0	0
0	0.666667	0	0.583333	0	0
0	1	1	1	0	0

PURCHASES_TRX	CREDIT_LIMIT	PAYMENTS	MINIMUM_PAYMENTS	PRC_FULL_PAYMENT	TENURE
2	1000	201.802084	139.509787	0	12
0	7000	4103.0326	1072.34022	0.222222	12
12	7500	622.066742	627.284787	0	12
1	7500	0		0	12
1	1200	678.334763	244.791237	0	12
8	1800	1400.05777	2407.24604	0	12
64	13500	6354.31433	198.065894	1	12

Table 2: Dataset

Data Dictionary

Let's take a quick look at the meaning of the variables present in the data.

Feature	Description
CUST_ID	Credit card holder's ID
BALANCE	Monthly average balance
BALANCE_FREQUENCY	How frequently the balance is updating, or, rate of credit card usage
PURCHASES	Total purchase amount
ONEOFF_PURCHASES	Total amount of one-off purchases
INSTALLMENTS_PURCHASES	Total amount of instalment purchases
CASH_ADVANCE	Total cash-advance (withdrawal of cash from ATM) amount
PURCHASES_FREQUENCY	Frequency of purchases (PURCHASES_TRX/TENURE)
ONEOFF_PURCHASES_FREQUENCY	Frequency of one-off-purchases
PURCHASES_INSTALLMENTS_FREQUENCY	Frequency of instalment purchases
CASH_ADVANCE_FREQUENCY	Cash-Advance frequency
CASH_ADVANCE_TRX	Number of transactions made of 'Cash in Advance'
PURCHASES_TRX	Number of purchase transactions made
CREDIT_LIMIT	Credit limit
PAYMENTS	Total payments (due amount paid by the customer to decrease their statement balance) in the period
MINIMUM_PAYMENTS	Total minimum payments that were due in the period
PRC_FULL_PAYMENT	Percentage of months with full payment of the due statement balance
TENURE	Number of months as a customer

Table 3: Data Dictionary

Descriptive Statistics

Now let's take a look at the count, mean, standard deviation and 5-number summary of each of the variables.

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE
count	8950.0	8950.0	8950.0	8950.0	8950.0	8950.0
mean	1564.5	0.9	1003.2	592.4	411.1	978.9
std	2081.5	0.2	2136.6	1659.9	904.3	2097.2
min	0.0	0.0	0.0	0.0	0.0	0.0
25%	128.3	0.9	39.6	0.0	0.0	0.0
50%	873.4	1.0	361.3	38.0	89.0	0.0
75%	2054.1	1.0	1110.1	577.4	468.6	1113.8
max	19043.1	1.0	49039.6	40761.3	22500.0	47137.2

	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX
count	8950.0	8950.0	8950.0	8950.0	8950.0	8950.0
mean	0.5	0.2	0.4	0.1	3.2	14.7
std	0.4	0.3	0.4	0.2	6.8	24.9
min	0.0	0.0	0.0	0.0	0.0	0.0
25%	0.1	0.0	0.0	0.0	0.0	1.0
50%	0.5	0.1	0.2	0.0	0.0	7.0
75%	0.9	0.3	0.8	0.2	4.0	17.0
max	1.0	1.0	1.0	1.5	123.0	358.0

	CREDIT _LIMIT	PAYMENTS	MINIMUM_PAY MENTS	PRC_FULL_PAY MENT	TENURE
count	8949.0	8950.0	8637.0	8950.0	8950.0
mean	4494.4	1733.1	864.2	0.2	11.5
std	3638.8	2895.1	2372.4	0.3	1.3
min	50.0	0.0	0.0	0.0	6.0
25%	1600.0	383.3	169.1	0.0	12.0
50%	3000.0	856.9	312.3	0.0	12.0
75%	6500.0	1901.1	825.5	0.1	12.0
max	30000.0	50721.5	76406.2	1.0	12.0

Table 4: Descriptive Statistics

FEATURE ENGINEERING

Here, we will perform some mutation tasks in order to derive a few key features.

Deriving KPIs & Other Useful Features

1. **Monthly Average Purchase and Monthly Average Cash Advance Amount**

   These show the average amount of purchase and cash advance transactions made by the user in a month.

   - *Monthly average purchase = total amount of purchases / total tenure*
   - *Monthly average cash advance = total cash advance / total tenure*

2. Purchase Type

There are four types of purchases made by a user. These are -

- *ONEOFF* - users who only make one-off purchases
- *INSTALLMENTS* - users who only make installments purchases
- *BOTH* - users who make both types of purchases
- *NONE* - users who do not make either of the types of purchases (they either make cash advance transactions or are inactive)

Here's the frequency table –

Both	2774
Installments	2260
None	2042
One-off	1874

Table 5: Purchase Types Count

Note: – NONE purchase type include cash advance transactions.

3. Average Amount Per Purchase and Cash Advance Transaction

These show the average amount spent on a single purchase and cash advance transaction by the user.

- *Average amount per purchase transaction = total amount of purchases / total purchase transaction*
- *Average amount cash advance transaction = total cash advance / total cash advance transaction*

4. **Limit Usage (or Balance to Credit Limit Ratio)**

   It tells how much debt a user is carrying or how much credit they are using from their existing limit. (Kagan, 2020)

   - *Credit utilization ratio = balance / credit limit*

5. **Payments to Minimum Payments Ratio**

   It tells us how much more than the minimum payment a user is paying.

   - *Payments to Minimum Payments Ratio = Payments made by the user / Minimum Payments due to the user*

Using the Derived KPIs to Gain Insights into the Customer Profiles

Let's try to gain some insight on the customer profiles using visualizations.

1. **Distribution of Purchase Types**

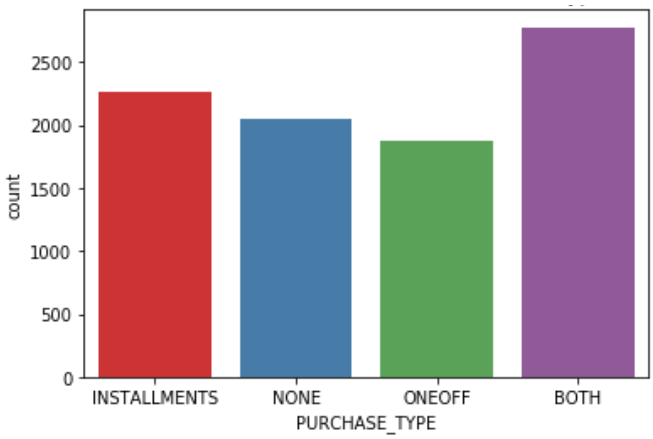

Figure 1: Distribution of Users of Different Purchase Types

We can see that the majority of credit card users prefer to make purchases in both one-off and installment payments.

2. Credit Utilization by Different Purchase Types

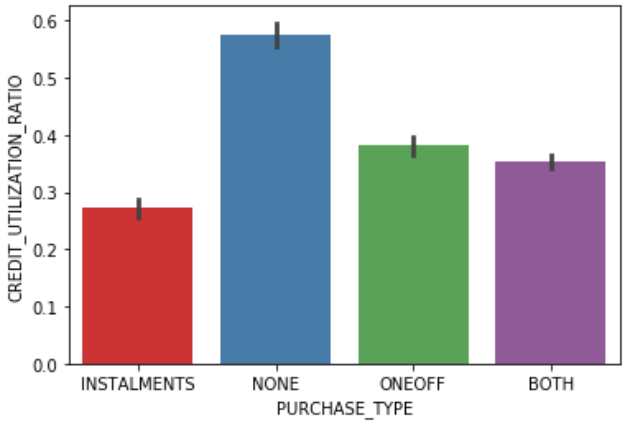

Figure 2: Credit Utilization by Different Purchase Types

We can see that credit card users, in the 'None' category, have the highest credit utilization ratio, that is, users that only use their credit cards to make cash advance transactions use their credit limit the most.

3. Total Purchases by Different Purchase Types

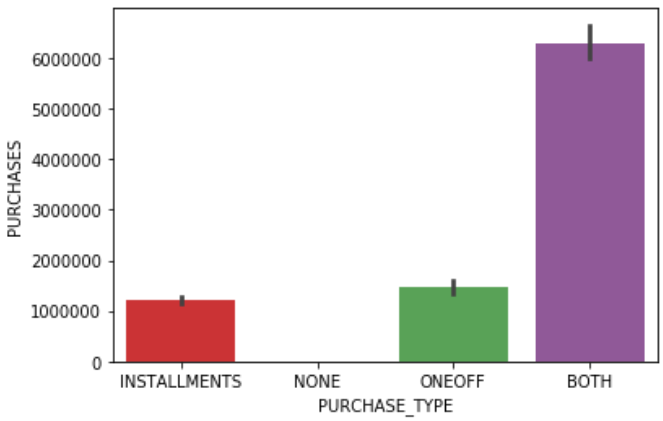

Figure 3: Total Purchases by Different Purchase Types

Here, we can argue that our most active credit card users are those that make use of both – one off and installments payments services.

DATA PREPROCESSING

Let's perform some basic data preprocessing such as handling any missing values in the data.

Handling Missing Data

Let's have a look at the count and percentage of observations that are missing in each variable.

	miss_count	miss_%
CUST_ID	0	0.000000
BALANCE	0	0.000000
BALANCE_FREQUENCY	0	0.000000
PURCHASES	0	0.000000
ONEOFF_PURCHASES	0	0.000000
INSTALLMENTS_PURCHASES	0	0.000000
CASH_ADVANCE	0	0.000000
PURCHASES_FREQUENCY	0	0.000000
ONEOFF_PURCHASES_FREQUENCY	0	0.000000
PURCHASES_INSTALLMENTS_FREQUENCY	0	0.000000
CASH_ADVANCE_FREQUENCY	0	0.000000
CASH_ADVANCE_TRX	0	0.000000
PURCHASES_TRX	0	0.000000
CREDIT_LIMIT	1	0.011177
PAYMENTS	0	0.000000
MINIMUM_PAYMENTS	313	3.498379
PRC_FULL_PAYMENT	0	0.000000
TENURE	0	0.000000
AVG_MONTH_PURCHASE	0	0.000000
AVG_MONTH_CASH_ADVANCE	0	0.000000
PURCHASE_TYPE	0	0.000000
AVG_AMT_PURCHASE_TRX	0	0.000000
AVG_AMT_CASH_ADVANCE_TRX	0	0.000000
CREDIT_UTLIZATION_RATIO	1	0.011177
PAY_MIN_PAY_RATIO	313	3.498379

Table 6: Missing Data

We can see that only 4 columns have missing values and they form less than 30% of the total values present in the column. Therefore, we can impute them (against dropping the entire column).

For the purpose of imputation, we'll follow a 4-step procedure, which is as follows.

STEP 1 - We will delete some known values from the dataframe

STEP 2 - Then we will use different imputing techniques

STEP 3 - We will compare all imputation results of different techniques with the actual values

STEP 4 - We will choose the imputation technique whose results are closer to the actual values

Imputing using Mean

Actual Values - 2407.246035, 156.644197, 11142.93224, 153.957216

Imputed Values - 863.149816, 863.149816, 863.149816, 863.149816

Since the values appear significantly far apart, we will use a different imputation method.

Imputing using Median

Actual Values - 2407.246035, 156.644197, 11142.93224, 153.957216

Imputed Values - 312.4522915, 312.4522915, 312.4522915, 312.4522915

Since the values are still significantly far apart, we will use a different imputation method.

Imputing using KNN Imputation

KNN refers to the K Nearest Neighbors algorithm, which is a classification algorithm. It works by selecting the k nearest neighbors to an observation whose class is to be predicted. Once the neighbors are identified using a distance formula such as Euclidean distance, it uses a voting criterion to assign class to the given observation.

But, KNN can also be used as a regressor, if instead of voting, we take the average values of the neighbors. This is how we are going to estimate the missing values.

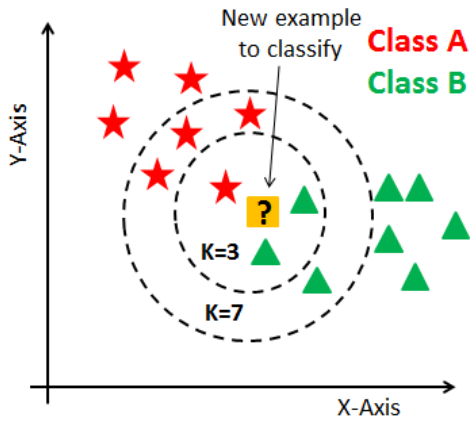

Figure 4: KNN Imputation (Navlani, 2018)

Limitations of KNN Algorithm

1. Since KNN algorithm uses mean value for estimation, it is highly sensitive to outliers.
2. Since it's a distance-based algorithm, feature-scaling is required.

Note: - Since we will proceed with k-means clustering afterwards, which is also distance-based and is sensitive to outliers, we can remove the outliers and perform feature scaling here.

Removing Outliers

Outliers are values that 'lie outside' the other values. The following formulas were used to remove the extreme outliers: -

$$IQR = Q3 - Q1$$

$$lower\ outer\ fence = Q1 - 3*IQR$$

$$upper\ outer\ fence = Q3 + 3*IQR$$

The observations where any given feature's values fell outside the fences, were dropped from the data.

After removing the outliers the dataset was reduced to **4426** observations.

Scaling Variables

A feature has two components – magnitude and unit. But machine learning algorithms are not aware of the unit element behind a number. That is, it treats features having different units just the same, which is problematic, since they are like apples and oranges. In terms of distance-based algorithms, the algorithm wrongly assigns more weightage to the features that have higher magnitude values for finding out which points are closer to a certain other point.

The solution, hence, is to bring the values in a fixed range or scale. This is called feature scaling. (Feature scaling, 2020)

Before scaling, we will first check the normality of the data using Jarque-Bera test in order to decide a suitable scaling technique.

Jarque-Bera Test

Since, the size of the data is greater that 2000, we are using this particular test out of many to check for the normality of data.

The test matches the skewness and kurtosis of data to see if it matches a normal distribution. A normal distribution has a skew of zero (i.e. it's perfectly symmetrical around the mean) and a kurtosis of three.

The formula for the Jarque-Bera test statistic (usually shortened to just **JB test statistic**) is:

$$JB = \frac{N}{6}\left(S^2 + \frac{(K-3)^2}{4}\right)$$

where,

S = sample skewness,

K = sample kurtosis, &

N = sample size. (Jarque–Bera test, 2020)

The null and alternate hypothesis in the test are: -

H_0: the data is normally distributed.
H_a: the data does not come from a normal distribution.

If the *P*-value is less than (or equal to) α, then the null hypothesis is rejected in favour of the alternative hypothesis. And, if the *P*-value is greater than α, then the null hypothesis is not rejected.

After conducting the test, we find that none of our variables are normally distributed.

Hence, in this case, we will proceed with Min-Max Normalization of the variables.

Min-Max Normalization

Min-max normalization is an operation which rescales a set of data. In this approach, the data is scaled to a fixed range - usually 0 to 1.

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

After performing the scaling, all our features are now having values which are only between 0 and 1.

Imputing Values

Actual Values - 110.77257639, 94.55678257, 330.25530328, 290.95634019

Imputed Values - 88.38863089, 279.0490857, 190.01063702, 168.88502209

Note that the scaled values are multiplied by 100 for the purpose of better comparison. Since the values seem close enough, we'll use the KNN imputed values.

FEATURE EXTRACTION USING PCA

Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the older ones). Let us first understand the need for feature extraction or feature selection.

Curse of Dimensionality

It means that as the number of features or dimensions grow, the amount of data we need to generalize (more the redundancy, better the generalization) accurately grows exponentially.

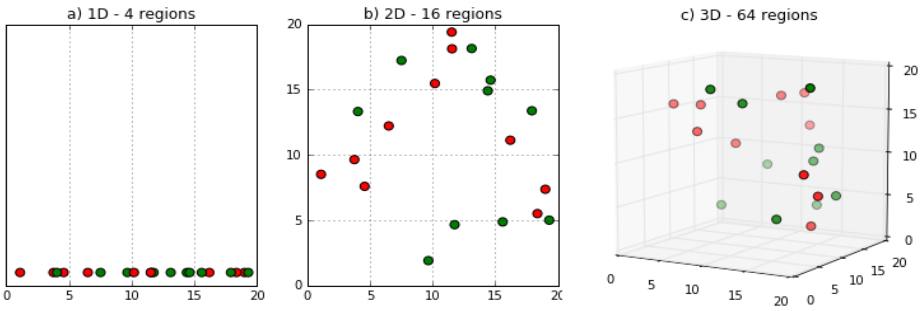

Figure 5: Curse of Dimensionality (deepai, n.d.)

The exponential growth in dimension space causes high sparsity in the data set and unnecessarily increases storage space and processing time for the particular modelling algorithm. (Pore, 2017)

The idea is that observed features or observed dimensionality obscure the true or intrinsic dimensionality of the data.

Principal Component Analysis

Principal Component Analysis is a dimensionality reduction method that performs feature extraction by constructing a new set of dimensions which are linear combinations of the original. It does so in a way that preserves as much structure in the data as possible.

Original dimensions: $X_1\ X_2\ X_3\ \ldots.\ X_d$

New dimensions: $E_1\ E_2\ E_3\ \ldots.\ E_m = f\ (X_1\ X_2\ X_3\ \ldots.\ X_d)$

1st PC: in the direction of the greatest variability in the data

2nd PC: perpendicular to 1st, greatest variability of what's left, and so on

The line that explains the most variability, most preserves the distance between data points, which is important especially when working with distance-based algorithms. (Brems, 2017)

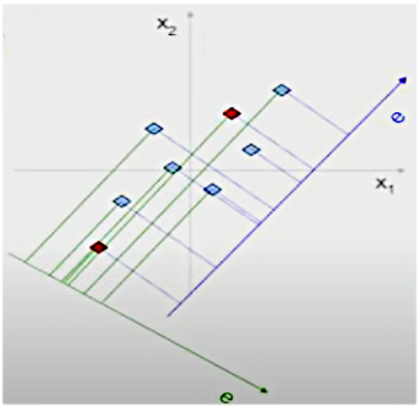

Figure 6: Eigenvector Preserving Most Variability

If you look at the red points, their projections on the blue line keep their distance the same while their projections on the green line get on top of each other. It means that the blue line is better as it preserves the structure of the data.

PCA Algebra

Let's have a look at the calculations involved behind the creation of principal components.

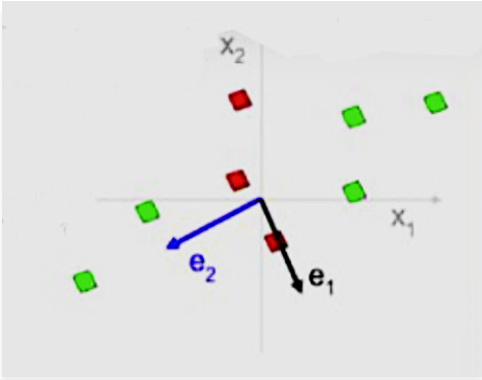

Figure 7: Perpendicular Eigenvectors

1. **Center the data**

   Subtract mean from each attribute. (Dubey, The Mathematics Behind Principal Component Analysis, 2018)

$$X_{i,\,a} = X_{i,\,a} - \prod_a$$

   Where,

   i: i^{th} observation

   a: a^{th} feature

   $\prod_a$: Mean of a^{th} feature

2. **Compute covariance matrix**

   The formulae for variance and covariance are as follows: -

$$Cov_{b,\,a} = \frac{1}{n}\sum_{i=1}^{n} X_{i,b}\, X_{i,a}$$

$$Var_a = \frac{1}{n}\sum_{i=1}^{n} X_{i,a}^2$$

Hence the Covariance Matrix is written as: -

$$\Sigma = \begin{bmatrix} var_1 & cov_{1,2} \\ cov_{2,1} & var_2 \end{bmatrix}$$

Where,

Σ : Covariance matrix

Note: - d number of features will generate a d X d covariance matrix.

3. Calculating eigenvalues and their corresponding eigenvectors

$$det\,(\Sigma - \lambda\, I) = 0$$

It will produce an equation with root d. Solving this equation will give us the required eigenvalues.

$$\Sigma e_i = \lambda_i\, e_i$$

Where,

Σ : Covariance matrix

e: i^{th} eigenvector of length d

λ: i^{th} eigenvalue

Note: - For a d X d covariance matrix there will be d eigenvectors and their corresponding d eigenvalues.

Divide the eigenvector by its Euclidean distance in order to get a unit length vector.

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

These eigenvectors are our new principal components.

4. **Project data points in the new dimensions**

Original data point: X = {X_1 X_2 X_3 …. X_d}

New data point: X' = {X'_1 X'_2 X'_3 …. X'_m}

$$X'_i = (X_i - \Pi_i)^T e_i$$

Selecting Optimal Number of Components

Let's see how we can select the optimal number of PCs for our dataset.

Variance explained by a principal component

The amount of variance explained by each of the component is given by -

$$V = \frac{1}{n}\sum_{i=1}^{n}(X'_{ij} - \Pi)^2$$

Scree Plot

A scree plot shows how much variation each PC captures from the data. The y axis is the amount of variation explained by principal component. An ideal curve should be steep, then bends at an "elbow" — this is the cutting-off point — and after that flattens out. These are the number of components that are sufficient to significantly describe the data.

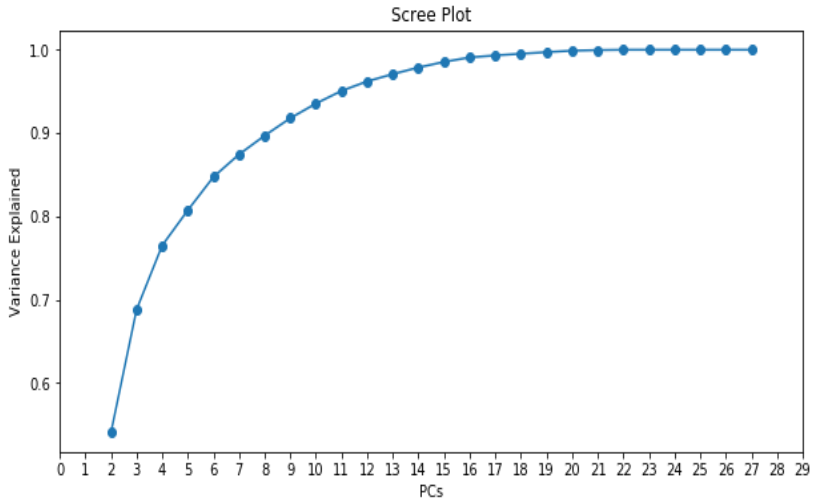

Figure 8: Scree Plot for Finding Optimal No. of Components

From the above chart we can see that after the 13th component, the variance tends to increase very slowly. Therefore, 13 will be our optimal no. of components which explain around 97% variance in the data.

APPLYING CLUSTERING ALGORITHMS

Clustering is an unsupervised learning algorithm which is used to group similar observations together in order to find meaningful structure, explanatory underlying processes, and groupings present in the set of observations.

For the purpose of the project I have used three different clustering algorithms, which are –

1. K-means
2. K-medoids
3. DBSCAN

and then compared them on basis of their Silhouette Scores. Then I chose the algorithm with the greatest score to perform the groupings. Let's see them one by one.

K-means

K-means clustering is one of the simplest and most commonly used clustering algorithms. It tries to find cluster centres that are representative of certain regions of the data. (Sharma P. , The Most Comprehensive Guide to K-Means Clustering You'll Ever Need, 2019)

The algorithm alternates between two steps: assigning each data point to the closest cluster centre, and then setting each cluster centre as the mean of the data points that are assigned to it.

The algorithm is finished when the assignment of instances to clusters no longer changes.

Finding Optimal Number of Clusters (k)

The goal here isn't just to make clusters, but to make good, meaningful clusters. Quality clustering is when the data points within a cluster are close together, and afar from other clusters. That is, we need to minimize the intra-cluster distance and maximize the inter-cluster distance.

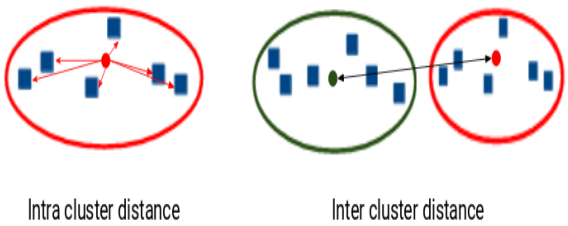

Figure 9: Inter and Intra Cluster Distances (Sharma P. , The Most Comprehensive Guide to K-Means Clustering, n.d.)

The correct choice of k is often ambiguous. Increasing k without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster (i.e., when k equals the number of data points, n). Intuitively then, the optimal choice of k will strike a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster. There are several methods for making this decision, one of them is the elbow method.

Elbow Method

An elbow graph is a line plot between the number of clusters chosen on the x-axis and the inertia on the y-axis. (Elbow method (clustering), 2020)

Inertia tells how far away the points within a cluster are. Therefore, a small value of inertia is aimed for. It is the sum of squared distances of samples to their closest cluster centre or it can be stated as the within-cluster Sum of Squared Errors (SSE). The range of inertia's value starts from zero and goes up.

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(j)} \right\|_2^2$$

The intuition is that increasing the number of clusters will naturally decrease the intra-cluster distances within the clusters but that at some point this decrease is marginal, and the 'elbow' reflects it. This point tells us the actual number of clusters in our data.

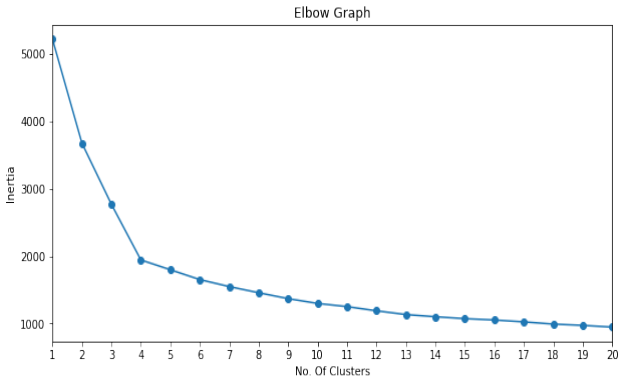

Figure 10: Elbow Graph (K-means)

From the above graph, we can see that the optimum number of clusters is 4.

Output

The following result in the form of a crosstab was observed –

Cluster label	BOTH	INSTALMENTS	NONE	ONEOFF
0	0	1076	0	0
1	0	0	1107	0
2	1354	0	0	0
3	0	0	0	889

Table 7: K-means Clusters

This is the 2D visualization of the clusters taking only the first 2 principal components –

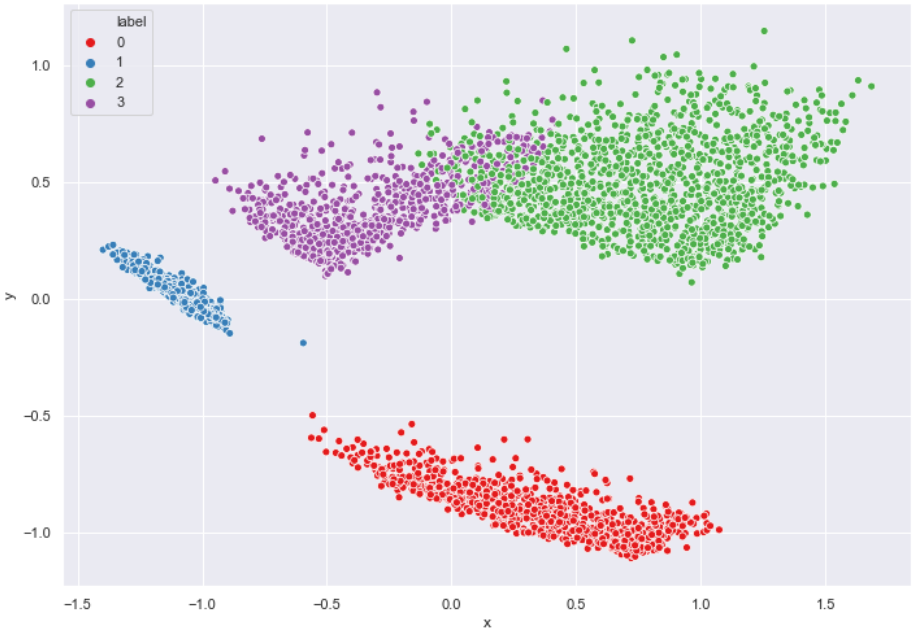

Figure 11: K-means Visualization

One assumption of variable importance in clustering is that if the average value of a variable ordered by clusters differs significantly among each other, that variable is likely important in creating the clusters. Let's have a look at the top 10 variables that have the highest inter-cluster variance in their mean values –

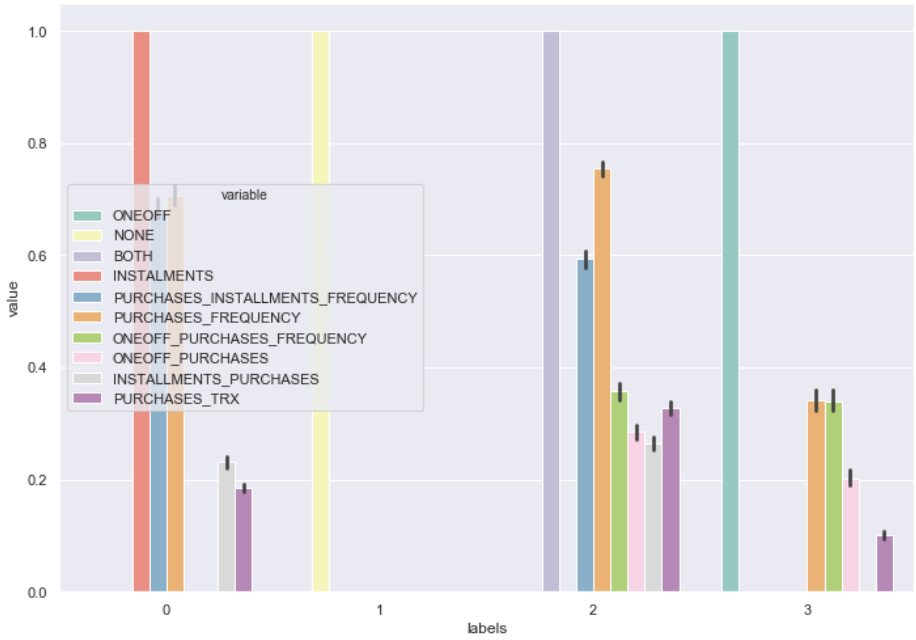

Figure 12: K-means Top 10 Variables

These are the variables that are most important in defining the clusters. (Grootendorst, 2019)

K-medoids

The k-medoids or partitioning around medoids (PAM) algorithm is similar to the k-means algorithm as both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize the distance between points labelled to be in a cluster and a point designated as the centre of that cluster. In contrast to the k-means algorithm, k-medoids chooses actual data points as centres. (k-medoids, 2020)

In order to choose the optimum number of clusters the elbow method was used with number of clusters on the x-axis and inertia on the y-axis. This is the result –

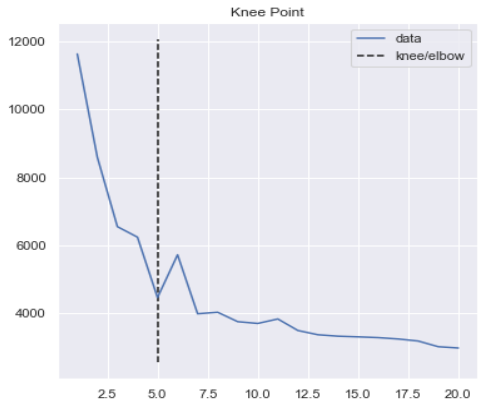

Figure 13: Elbow Graph (K-medoids)

From the above graph, we can see that the optimum number of clusters is 5.

Output

The following result in the form of a crosstab was observed –

Cluster label	BOTH	INSTALMENTS	NONE	ONEOFF
0	1597	0	0	0
1	0	2146	0	0
2	0	0	0	1781
3	1144	0	0	0
4	0	0	1965	0

Table 8: K-medoids Clusters

This is the 2D visualization of the clusters taking only the first 2 principal components -

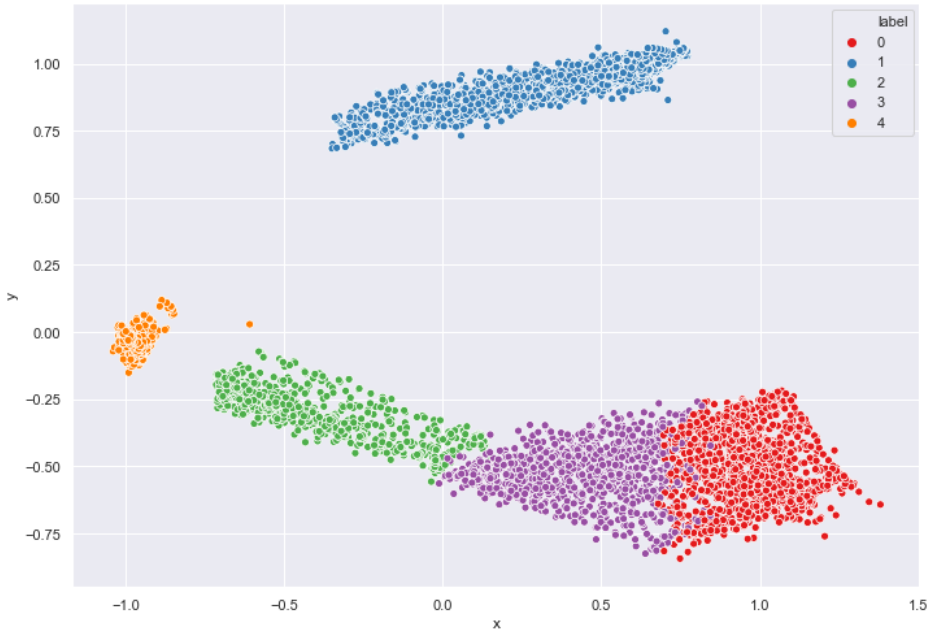

Figure 14: K-medoids Visualization

Let's have a look at the top 10 variables that have the highest inter-cluster variance in their mean values –

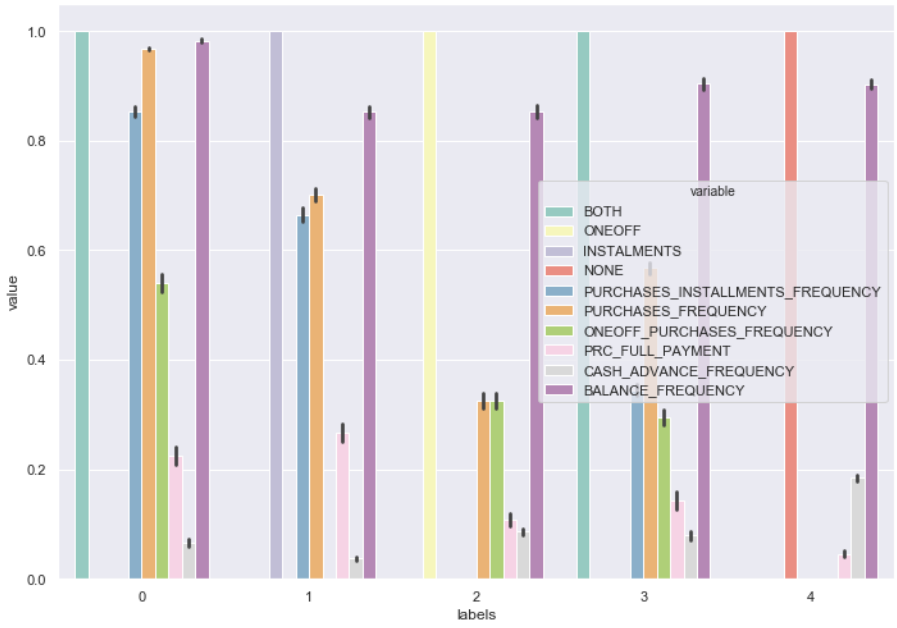

Figure 15: K-medoids Top 10 Variables

DBSCAN

DBSCAN or Density-based Spatial Clustering of Applications with Noise is yet another clustering algorithm other than k-medoids which is robust to outliers.

It divides the data points into three categories – core points, neighbour points, and outliers. It does that on the basis of two parameters – MinPoints and Eps. MinPoints is the number of minimum points (including the point itself) that a data point need in its surrounding area (or neighbourhood) for it to be classified as a core point. It can also be seen as the minimum number of points required to form a dense region. Surrounding area is defined by Eps, wherein Eps is the length of the radius drawn by taking the data point as the centre. The points that fall in this area are the neighbour points. A point which is neither a core point or a neighbour point is classified as an outlier.

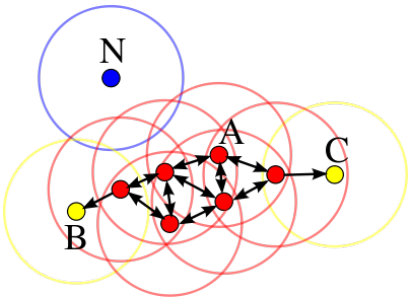

Figure 16: DBSCAN Clustering (DBSCAN, n.d.)

Parameter Tuning

MinPoints - As a rule of thumb, $minPts = 2 \cdot dim$ can be used. Since we have 10 PCs, our MinPoints will be 20. (DBSCAN, 2020)

Eps - If the eps value chosen is too small, a large part of the data will not be clustered, they will be considered outliers because don't satisfy the number of points to create a dense region. On the other hand, if the value that was chosen is too high, clusters will merge and the majority of objects will be in the same cluster. The eps should be chosen based on the distance of the dataset (we can use a k-distance graph to find it), but in general small eps values are preferable.

(Sitanggang, 2015)

K-distance Graph

A k-distance graph is made by plotting the distance to the k = minPts-1 nearest neighbour ordered from the largest to the smallest value. Good value of ε is where this plot shows an "elbow".

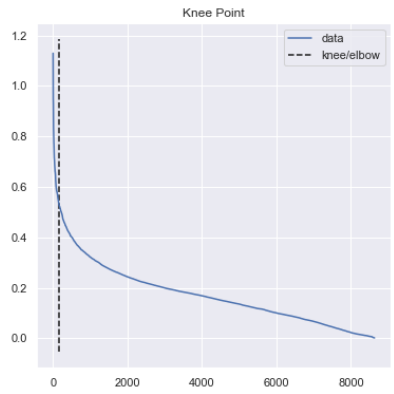

Figure 17: K-distance Graph

The elbow point lies at eps = 0.5257.

Output

The following result in the form of a crosstab was observed –

Cluster label	BOTH	INSTALMENTS	NONE	ONEOFF
-1	15	0	4	6
0	0	2146	0	0
1	0	0	1961	0
2	0	0	0	1775
3	2726	0	0	0

Table 9: DBSCAN Clusters

Note: - -1 refers to outliers and it was not taken while calculating the Silhouette Score because it's not a cluster in itself.

This is the 2D visualization of the clusters taking only the first 2 principal components –

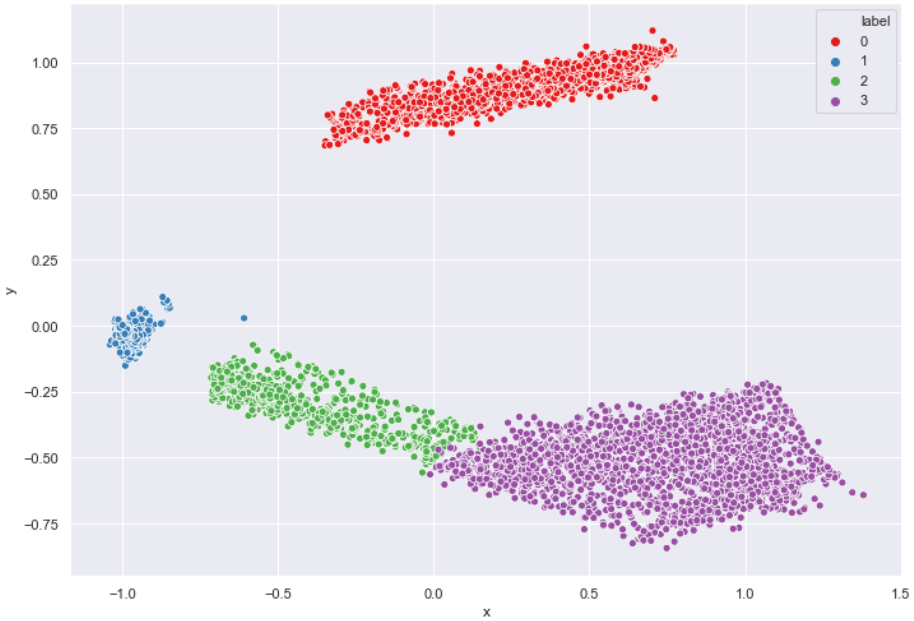

Figure 18: DBSCAN Visualization

Let's have a look at the top 10 variables that have the highest inter-cluster variance in their mean values –

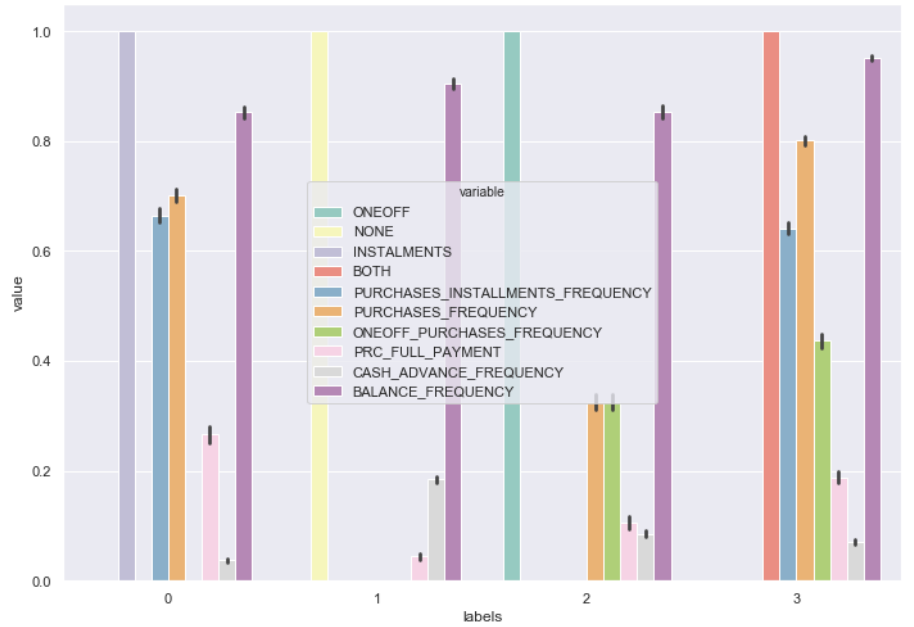

Figure 19: DBSCAN Top 10 Variables

CONCLUSION

We have seen earlier the outputs that were given by three different clustering algorithms, and now it's time to decide which model should be chosen for deployment at this final stage of the project.

There are both advantages and drawbacks of each of the algorithms that I've used in the project. For example, K-means may be easier to implement and understand, but it is sensitive to outliers, and because of that we had to remove a large portion of our data. K-medoids, on the other hand, is robust to outliers but is not suitable for clustering non-spherical (arbitrary shaped) groups of objects, which is also the case with k-means. Also, both k-means and k-medoids may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly. Lastly, DBSCAN is both robust to outliers and can produce non-spherical clusters, but is highly sensitive to parameter tuning. These are the reasons why it is difficult to decide which one to use.

Therefore, I used a metric called the 'Silhouette Coefficient' to do an objective analysis of the performances of the algorithms. The silhouette score computes the compactness of a cluster. It is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). 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 neighbouring clusters. (Sebastian Raschka, 2017)

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

where, y is the mean intra cluster distance i.e., mean distance to the other instances in the same cluster and x depicts mean nearest cluster distance i.e., mean distance to the instances of the next closest cluster. We calculate the silhouette coefficient for each sample or observation in the dataset and finally, compute its mean. In order for the silhouette coefficient to be closer to 1, x, or the mean inter-cluster distance, must be greater than y, the intra-cluster distance.

Let's have a look at the silhouette coefficients of our three algorithms –

K-means	0.454
K-medoids	0.474
DBSCAN	0.549

Table 10: Silhouette Scores

Since, DBSCAN has the greatest value of the silhouette coefficient, it is the one that we will choose for deployment. The outputs of this algorithm are our final results.

REFERENCES

DBSCAN. (n.d.). Retrieved from wikipedia.org/: https://en.wikipedia.org/wiki/DBSCAN#Parameter_estimation

Wikipedia. (2020). *DBSCAN*. Retrieved from wikipedia: https://en.wikipedia.org/wiki/DBSCAN#Parameter_estimation

Dubey, A. (2018, 12 21). *The Mathematics Behind Principal Component Analysis*. Retrieved from towardsdatascience: https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643

Sharma, G. (2018, 9 22). *4 Types of Customer Segmentation All Marketers Should Know*. Retrieved from business2community: https://www.business2community.com/customer-experience/4-types-of-customer-segmentation-all-marketers-should-know-02120397

Revolving credit. (2020, 7 16). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Revolving_credit

Credit Card Interest Rates. (2020, 6 30). Retrieved from paisabazaar: https://www.paisabazaar.com/credit-card/interest-rates/

Irby, L. (2020, 3 25). *Credit Card Grace Period Explained*. Retrieved from thebalance: https://www.thebalance.com/credit-card-grace-period-explained-960699

Maji, P. (2020, 2 24). *Credit Card Payment: What happens if you pay only the minimum amount due?* Retrieved from financialexpress: https://www.financialexpress.com/money/credit-card-payment-what-happens-if-you-pay-only-the-minimum-amount-due/1877081/

Kagan, J. (2020, 2 9). *Credit Utilization Ratio*. Retrieved from investopedia: https://www.investopedia.com/terms/c/credit-utilization-rate.asp

Feature scaling. (2020, 7 16). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Feature_scaling

Jarque–Bera test. (2020, 7 16). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Jarque–Bera_test

Pore, P. (2017, 4). *Must-Know: What is the curse of dimensionality?* Retrieved from kdnuggets: https://www.kdnuggets.com/2017/04/must-know-curse-dimensionality.html

Brems, M. (2017, 4 18). *A One-Stop Shop for Principal Component Analysis*. Retrieved from towardsdatascience: https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c

Dubey, A. (2018, 12 21). *The Mathematics Behind Principal Component Analysis*. Retrieved from towardsdatascience: https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643

Sharma, P. (2019, 8 19). *The Most Comprehensive Guide to K-Means Clustering You'll Ever Need*. Retrieved from analyticsvidhya: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

Elbow method (clustering). (2020, 7 16). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Elbow_method_(clustering)

Grootendorst, M. (2019, 7 31). *Cluster Analysis: Create, Visualize and Interpret Customer Segments*. Retrieved from towardsdatascience: https://towardsdatascience.com/cluster-analysis-create-visualize-and-interpret-customer-segments-474e55d00ebb

k-medoids. (2020, 7 16). Retrieved from wikipedia: https://en.wikipedia.org/wiki/K-medoids

DBSCAN. (2020, 7 16). Retrieved from wikipedia: https://en.wikipedia.org/wiki/DBSCAN

Sebastian Raschka, V. M. (2017). Quantifying the quality of clustering via silhouette plots. In V. M. Sebastian Raschka, *Python Machine Learning* (p. 358). Pckt Publishing.

Sitanggang, N. R. (2015). *Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra.* Bogor, Indonesia: IOP Publishing Ltd.

Navlani, A. (2018). Retrieved from https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

deepai. (n.d.). Retrieved from https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality.

Sharma, P. (n.d.). *The Most Comprehensive Guide to K-Means Clustering*. Retrieved from analyticsvidhya: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

DBSCAN. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/DBSCAN

(n.d.). Retrieved from https://en.wikipedia.org/wiki/Revolving_credit

(n.d.). Retrieved from https://en.wikipedia.org/wiki/DBSCAN#Parameter_estimation