***Credit Card Clustering Performance Insights***

**INTRODUCTION:**

Marketing, as well known, is crucial for the growth and sustainability of any business. However, one of the key pain points for any marketing professionals is to know the customers and identify their needs. Tasked by colleagues in the marketing department, your job is to create a model to perform customer grouping.

**STEPS INVOLVED IN DATA CLUSTERING:**

This post aims to help Sales person in the marketing department. As usual, it is split into 5 parts.

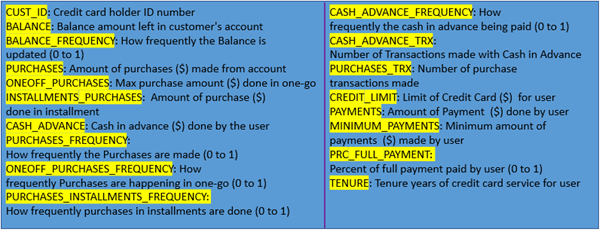
* Problem statement
* Data Review
* Data processing
* k-means clustering
* Conclusion

**PROBLEM STATEMENT:**

Marketing, as well known, is crucial for the growth and sustainability of any business. However, one of the key pain points for any marketing professionals is to know the customers and identify their needs. Tasked by colleagues in the marketing department, your job is to create a model to perform customer grouping.

**DATA REVIEW:**

The dataset contains 17 features and 8,950 records. As illustrated in Figure, it is about customer’s purchase and payment habits, such as how often a customer makes one-off or instalment purchases, or how often they make cash advances, how much payments are made, etc. By inspecting each customer, we can find which type of purchase he/she is keen on, or if he/she prefers cash advance over purchases.



**DATA PROCESSING:**

Generally, to get a deep understanding and preliminary findings on any raw data, we perform exploratory data analysis (EDA). I covered this topic and the general steps in several articles. If you are a newbie on EDA, here we will focus on some steps necessary for the following modelling.

**FILL MISSING VALUES:**

Due to various reasons, missing values commonly appear in raw data. To check how many missing values in each column

There are many approaches to dealing with missing values, including:

* Drop columns with missing values
* Imputation, filling in missing values with some number, such as the mean value along each column.
* An extension to Imputation. Imputation is the standard approach. However, imputed values may be above or below their actual values. To help model make a better prediction, we can add a column to tag the missing value as True or False.

In this case, we will fill the missing values the mean of the column. Specifically,

creditcard\_df.loc[(creditcard\_df[‘MINIMUM\_PAYMENTS’].isnull() == True), ‘MINIMUM\_PAYMENTS’] = creditcard\_df[‘MINIMUM\_PAYMENTS’].mean()creditcard\_df.loc[(creditcard\_df[‘MINIMUM\_PAYMENTS’].isnull() == True), ‘MINIMUM\_PAYMENTS’] = creditcard\_df[‘MINIMUM\_PAYMENTS’].mean()

**DROP FEATURE:**

The dataset contains 17 features. Now we need to consider if all of them is necessary to train the model. In our case, *‘CUST\_ID’* is of no help. So, we will drop it.

creditcard\_df.drop(‘CUST\_ID’, axis = 1, inplace = True)

**SCALE FEATURE:**

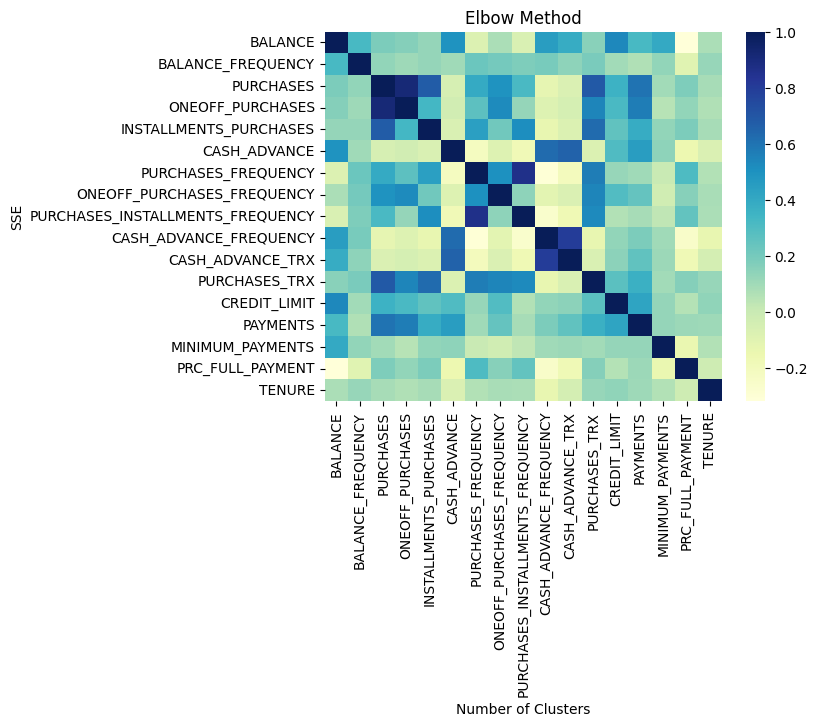
Some features such as *‘PURCHASES’* vary in a wide range, while other features such as *‘PURCHASES\_FREQUENCY’* vary between 0 to 1. We need to scale all features on the same range. Here we use **StandardScaler()** from **sklearn** to remove mean and scale to unit variance.

scaler = StandardScaler()  
creditcard\_df\_scaled = scaler.fit\_transform(creditcard\_df)

**k-MEAN CLUSTERING:**

k-means clustering is an unsupervised machine learning algorithm. According to Wikipedia, it aims to partition the observations into k sets so as to minimize the within-cluster sum of squares (WCSS). WCSS represents the sum of distances of all points to the centroid in a cluster. It starts with a group of randomly initialized centroids and then performs iterative calculations to optimize the position of centroids until the centroids stabilize, or the defined number of iterations is reached.

k-means clustering is an unsupervised machine learning algorithm. According to Wikipedia, it aims to partition the observations into k sets so as to minimize the within-cluster sum of squares (WCSS). WCSS represents the sum of distances of all points to the centroid in a cluster. It starts with a group of randomly initialized centroids and then performs iterative calculations to optimize the position of centroids until the centroids stabilize, or the defined number of iterations is reached.



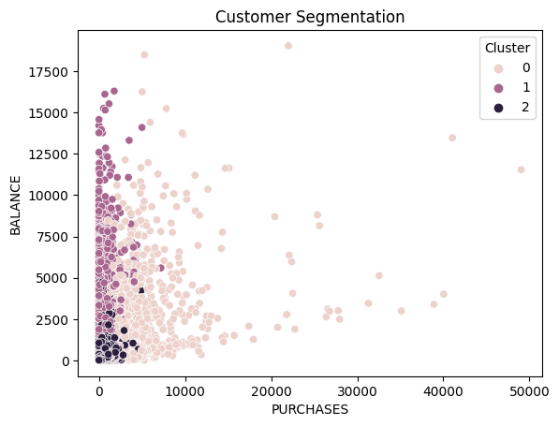
Note we set the number of time that k-means will be run to be 10. This means k-means will run 10 times with different centroid seeds, and the final results will be the best output in terms of WCSS. In addition, we set the maximum number of iterations of k-means for a single run to be 300

**PCA:**

The histogram chart is great to analyse the cluster features, but not so great to visualize the overall cluster distribution. Because we have 17 features, we have to compress the dimensions of freedom to visually observe the cluster distribution. To do such, we use principal component analysis (PCA) to compress features into a 2D space.

For those who are not familiar with PCA, PCA is commonly used for dimensionality reduction by projecting data only onto the first principle components to obtain low-dimension data while preserving or maximizing the variances along the projected direction

So, to implement PCA,



Finally, with 17 features compressed to 2 features, above figure shows the scatter plot of the k-means prediction results. You may notice the Violet cluster is somehow scattered among the Magenta cluster and pink cluster. This may indicate the Violet cluster has feature similarity with the others.

**CONCLUSION:**

We created a k-means model that managed to segment credit card customers into 8 groups. We found some interesting purchases and payment patterns for cluster 0, 2 and 4, and more remains to be explored. We also noticed some overlaps between clusters, indicating there are spaces to improve.