Group 1

PREPARED BY:

Jose German, Mohammed Araf, Abdela Mossa, Luiz Tavares

CSML-1000

York University

Group Project

PREPARED FOR:

Hashmat Rohian

Instructor

CSML-1000

York University

Table of Contents

Executive Summary

DFBanker, a leading financial institution, has engaged 4Friends ML Inc. to address key challenges related to the effective utilization of credit card transaction data. With a dataset encompassing behavioral variables of 9000 active credit card holders, our objective was to leverage clustering analytics using Python as the primary programming language.

Through meticulous data preprocessing, including imputation and cleaning, we prepared the dataset for modeling and training. We evaluated various clustering algorithms, with KMeans emerging as the most favorable option, yielding significant insights into customer segments and transaction patterns.

Furthermore, our team developed a user-friendly application deployed on the Hugging Face platform, allowing DFBanker to predict optimal marketing strategies for individual customers based on cluster analysis. This application serves as a valuable tool for enhancing customer engagement and driving business growth.

Additionally, we provided detailed marketing strategy recommendations tailored to distinct customer segments identified through clustering. These recommendations aim to increase sales revenue and customer satisfaction by leveraging insights derived from the clustering model.

In conclusion, our partnership with DFBanker marks a significant milestone in leveraging data analytics to address business challenges. By integrating clustering analytics, we've empowered DFBanker to tailor marketing strategies, and enhance overall customer experience in the competitive credit card industry. As we continue our collaboration, we remain committed to driving positive outcomes and shaping the future of data-driven decision-making in finance.

Introduction

In response to the challenges faced by DFBanker in effectively utilizing data from credit card transactions, 4Friends ML Inc. is stepping in as a strategic partner to provide tailored solutions. Drawing on our expertise in analytics and machine learning, we will integrate clustering analytics into DFBanker's platform to address their specific needs.

Our team at 4Friends ML Inc. will begin by delving into the dataset containing behavioral variables of 9000 active credit card holders. Through advanced clustering techniques, we will uncover distinct groups or clusters of transactions, allowing DFBanker to gain valuable insights into customer segments and transaction patterns.

Our solutions will be focused on tailoring marketing strategies, and enhancing customer experience. From crafting Customer Experience Enhancement Strategies, our deliverables will equip DFBanker with actionable recommendations based on data-driven insights.

By partnering with 4Friends ML Inc., DFBanker will have access to cutting-edge analytics capabilities and strategic guidance, positioning them for success in the ever-evolving financial landscape.

Data Dictionary

Following is the Data Dictionary for Credit Card dataset:

DATA DICTIONARY	
Data Element Name	**Data Element Definition**
CUST_ID	Identification of Credit Card holder (Categorical)
BALANCE	Balance amount left in their account to make purchases
BALANCE_FREQUENCY	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
INSTALLMENTS_PURCHASES	Amount of purchase done in installment
CASH_ADVANCE	Cash in advance given by the user
PURCHASES_FREQUENCY	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
CREDIT_LIMIT	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

Exploratory Data Analysis:

Programming Language:

Python has been selected as the primary programming language for this project based on the expertise and experience of our existing project team.

Data Source:

The Customer Credit Card data used in this project has been sourced from an external provider, available at https://www.kaggle.com.

Data Dictionary:

We've leveraged the data dictionary available in the above section to interpret the dataset better.

Understanding the data:

- It was found that only CUST_ID is a categorical variable and all other variables are numeric.
- Following 2 variables have some null values and data preprocessing is required to replace these null values.
    - MINIMUM_PAYMENTS (3.50% Null values)
    - CREDIT_LIMIT (0.02 % Null values)

Data Preprocessing:

- Data Imputation: Mean values were calculated for below 2 numeric variables and it was used to replace the corresponding null values.
    - MINIMUM_PAYMENTS
    - CREDIT_LIMIT
    -

- Data Cleaning: It was found that following numeric variables do not have much variation and hence dropped to minimize the # columns within the dataset so that model performance can be improved.

    - CUST_ID (not required to train or test the model)
    - ONEOFF_PURCHASES_FREQUENCY (high correlation with ONEOFF_PURCHASES)
    - PURCHASES_INSTALLMENTS_FREQUENCY (High correlation with PURCHASES_INSTALLMENTS)
    - 'CASH_ADVANCE_FREQUENCY' (High Correlation with CASH_ADVANCE)
    - 'PRC_FULL_PAYMENT'] (Not relevant for Model development)

    -

Modeling and Training

We have decided to use the following 2 approaches for the model development & training & then, compare both of them and choose the best one.

1. Use PyCaret package for Model Development & Training
2. Use Kmeans algorithm for Model Development & Training

(1) <u>Using PyCaret package:</u>

Investigation of Different Clustering Algorithms

1. Using default settings of Pycaret

Clus. Algos	Evaluation Metrics					
	Silhoute e	Calinski-Har abasz	Davies-B ouldin	Homogeneit y	Rand Index	Complet- ness
Kmeans	0.3968	2675.4	1.3211	0	0	0
AP	0.1844	693.4	0.9937	0	0	0
Meanshft	0.4270	152.3	0.6082	0	0	0
Spectral	0	0	0	0	0	0
Agglom.	0.3815	2215.9	1.4904	0	0	0
DBSCAN	0	0	0	0	0	0
Optics	-0.5428	6.7718	1.3894	0	0	0
Birthc	0.3815	2215.9	1.4904	0	0	0

Group 1

Prior to applying any preprocessing techniques except the default settings, we observed that among the eight clustering algorithms, the KMeans clustering yielded the most favorable results compared to other clustering algorithms, which resulted in a higher Silhoutee value and Calinski-Harabasz values. Since we didn't use ground truth labels for comparison, we also obtained a result of 0 for the Homogeneity, Rand Index, and Completeness metrics, which rely on such comparisons to assess clustering performance. Upon analyzing the KMeans model, we also found that 63.73% of the dataset samples were assigned to Cluster-2, while 28.73% were assigned to Cluster-0, 0.32% to Cluster-1, and the remaining 7.2% to Cluster-3.

2. **After applying custom pre-processing techniques**

In order to enhance the performance of the proposed clustering models, we conducted preprocessing by dropping certain features as outlined in the preprocessing section. Additionally, guided by our exploratory data analysis (EDA) experiments, we reduced the number of clusters to 3 and applied min-max normalization to scale feature values within the range of 0 to 1, and also addressed missing values in the process. The results obtained after we applied these techniques are illustrated below.

As it is also observed from the tables, all six models, except for MeanShift and Agglomerative clustering, have shown improvements in performance when evaluated using the Silhouette metric. The Optics and DBSCAN models have benefited the most, with their performance increasing from -0.5428 to 0.4346 and from 0 to 0.5335, respectively.

Group 1

Clus. Algos	Evaluation Metrics					
	Silhoute e	Calinski-Har abasz	Davies-B ouldin	Homogeneit y	Rand Index	Complet- ness
Kmeans	0.4523	5621.1	0.9864	0	0	0
AP	0.1897	1195.1	1.3301	0	0	0
Meanshft	0.3946	634.6	1.2488	0	0	0
Spectral	0.4369	5601.8	1.2092	0	0	0
Agglom.	0.3578	4188.6	1.2753	0	0	0
DBSCAN	0.5335	58.2	1.4026	0	0	0
Optics	0.4346	14.7	1.3220	0	0	0
Birtch	0.4348	7423.7	0.9552	0	0	0

(2) Model Development using KMeans algorithm:

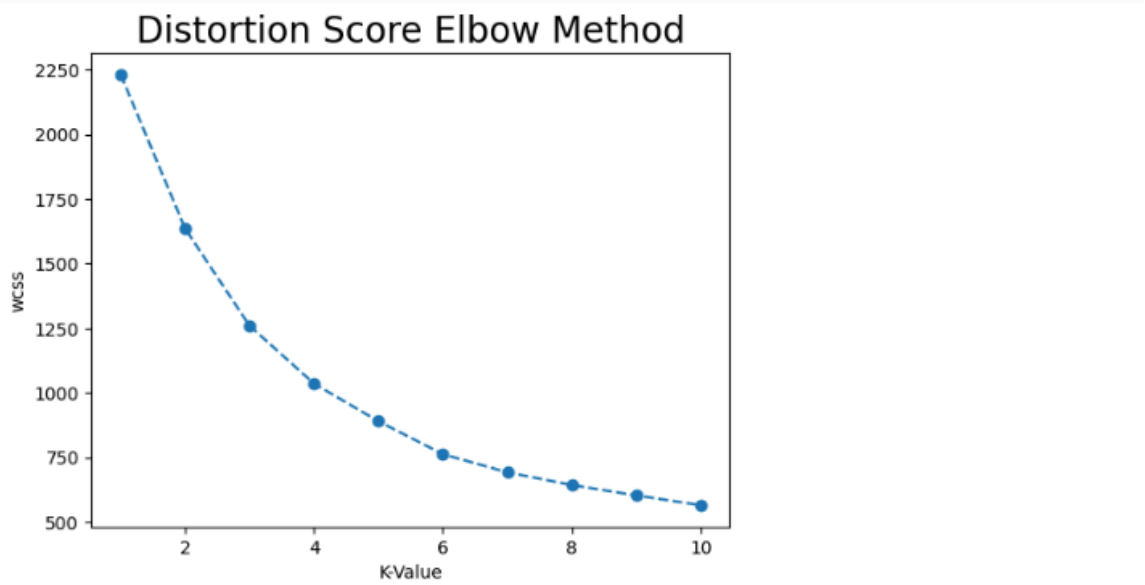
TSNE: We have used t-SNE to reduce the dimension of data into 2 so that we can plot the features in a two-dimensional plot. t-SNE (t-distributed Stochastic Neighbor Embedding) is an unsupervised non-linear dimensionality reduction technique for data exploration and visualizing high-dimensional data.

KMeans: KMeans algorithms have been chosen from the SKLearn python package. The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares (see below). This algorithm requires the number of clusters to be specified. It scales well to large numbers of samples and has been used across a large range of application areas in many different fields.

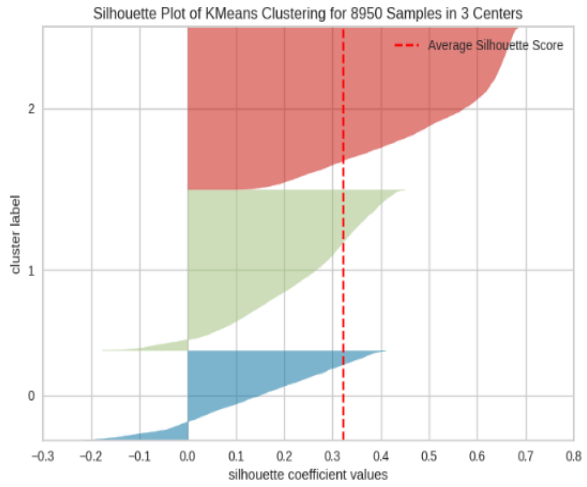
Group 1

Distortion Score (Elbow Method)

Distortion is taking into account ONLY the tightness of the cluster. This is measured by average of the squared distances between each point in a cluster and the cluster center. The closer all the points in a cluster are to the center of that same cluster, the lower the distortion is.



Silhouette also takes into account the distances between points of one cluster and NEAREST cluster center. Meaning that in order to have a good silhouette score, clusters generally need to be tighter and farther from each other.



<Axes: title={'center': 'Silhouette Plot of KMeans Clustering for 8950 Samples in 3 Centers'}, xlabel='silhouette coefficient values', ylabel='cluster label'>

It helped us to determine the optimal value for K=3 i.e customers can be divided into 3 clusters.

App Development

Deployment platform: Hugging Face (https://huggingface.co)

Hugging Face serves as a machine learning and data science platform, offering users the ability to construct, implement, and train machine learning models. As a community-

centric platform, it provides the necessary infrastructure for hosting ML and AI models and applications.

Based on our business problem, our application predicts the correct marketing strategy to each customer based on the parameters given. All of the work done in prior phases such as model training has allowed us to utilize existing credit card user data provided by our customer to segment and learn from the clusters or groups. With this information we can correctly target additional marketing to those specific groups which would likely take advantage of the offers and therefore increasing activity on credit cards which translates to more business.

The application's codebase is entirely written in Python and makes use of various libraries, including Gradio, for constructing the user interface. Gradio, an open-source Python library, enables the creation of web applications tailored for machine learning models.

Applications code-behind (python):

app.py > cc_predict

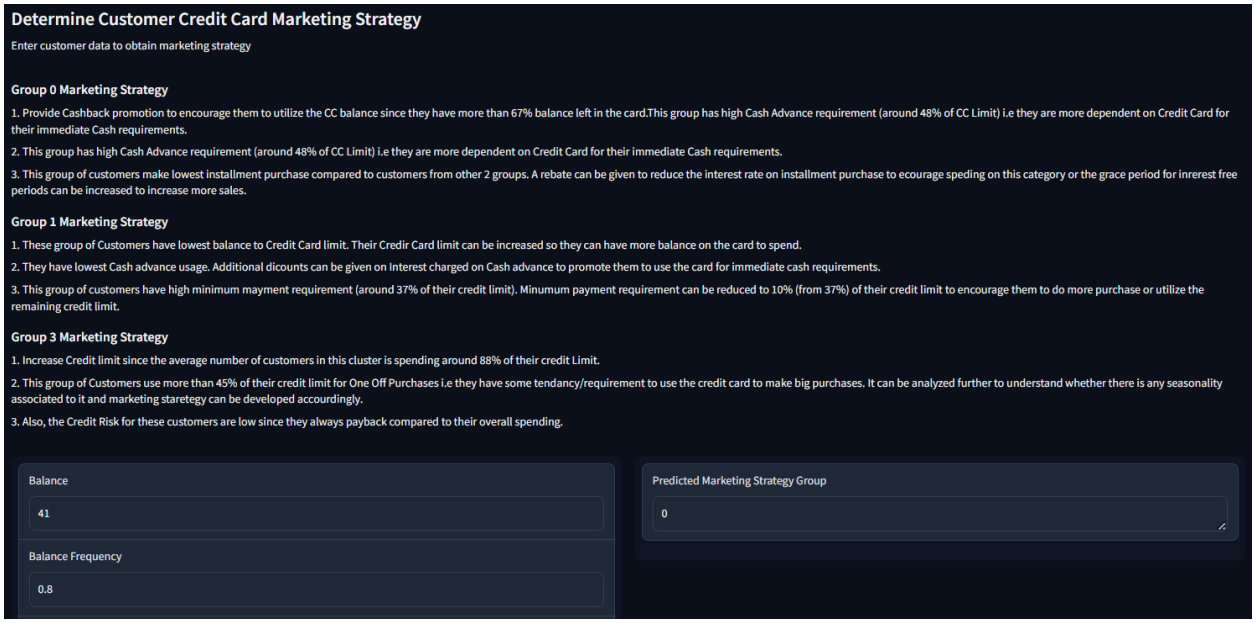
```python
# Imports
import numpy as np
import pandas as pd
import pickle
import gradio as gr
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import warnings
warnings.filterwarnings('ignore')

# Load saved model using pickle
with open('cc_model.pkl', 'rb') as file1:
    model1 = pickle.load(file1)

# Define prediction function
def cc_predict(balance,balance_frequency,purchases,oneoff_purchases,installments_purchases,
               cash_advance,purchases_frequency,oneoff_purchases_frequency,purchases_installments_frequency,cash_advance_frequency,
               cash_advance_trx,purchases_trx,credit_limit,payments,minimum_payments,
               prc_full_payment,tenure):
    input_data = pd.DataFrame({'BALANCE': [balance],'BALANCE_FREQUENCY': [balance_frequency],'PURCHASES': [purchases],'ONEOFF_PURCHAS
                               'CASH_ADVANCE': [cash_advance],'PURCHASES_FREQUENCY': [purchases_frequency],'ONEOFF_PURCHASES_FREQUENC
                               'CASH_ADVANCE_TRX': [cash_advance_trx],'PURCHASES_TRX': [purchases_trx],'CREDIT_LIMIT': [credit_limit]
                               'PRC_FULL_PAYMENT': [prc_full_payment],'TENURE': [tenure]})
    #input_data = [[40.900749,0.818182,95.40,0.00,95.4,0.000000,0.166667,0.000000,0.083333,0.000000,0,2,1000.0,201.802084,139.509787,

    predicted_output = model1.predict(input_data)

    return predicted_output[0]
```

Once the model went through the normal training process it was serialized into a pickle file in order to use in our online application. The trained model and application can also be used as an API to directly interact with existing systems to get marketing strategy:

Use of serialized pickle model in app code-behind:

```
# Load saved model using pickle
with open('cc_model.pkl', 'rb') as file1:
    model1 = pickle.load(file1)
```

Application interface:



Application URL:

https://huggingface.co/spaces/jos-ger/creditcardcluster

Marketing Strategies and Recommendations:

Based on the K-Means Clustering model that our analytics team developed, we are able to divide our consumer base into three distinct segments. We would like to propose following strategies to increase Sales Revenue and to increase Customer Satisfaction based on our findings from each of these customer segments. We are confident that it will significantly improve our client's marketing plans for the upcoming fiscal quarter.

Customer Segment	# of Customers (%)	Proposed Marketing Strategies
Cluster 0	3530 (39.4%)	• These groups of customers have the lowest balance in their credit cards. Their Credit Card limit is recommended to be increased so they can have more balance on the card to spend. • They have lowest Cash advance usage. Additional discounts can be given on Interest charged on Cash advance to promote them to use the card for immediate cash requirements. • This group of customers have a high minimum payment requirement (around 37% of their credit limit). Minimum payment requirements can be reduced to 10% (from 37%) of their credit limit to encourage them to do more purchases or utilize the remaining credit limit.
Cluster 1	3495 (39.05 %)	• Provide more "*Cashback*" promotion to encourage them to utilize the CC balance since they have more than 67% balance left in the card. • This group has high Cash Advance requirements (around 48% of CC Limit) i.e they are more dependent on Credit Cards for their immediate Cash requirements. • This group of customers make the lowest installment purchase compared to customers from other 2 groups. A rebate can be given to reduce the interest rate on installment purchase to encourage spending on this category or the grace period for interest free periods can be increased to increase more sales

Cluster 2	1925 (21.5 %)	• Increase Credit limit since the average number of customers in this cluster is spending around 88% of their credit Limit. • This group of Customers use more than 45% of their credit limit for "One Off Purchases" i.e they have some tendency/requirement to use the credit card to make big purchases. It can be analyzed further to understand whether there is any seasonality associated with it and marketing strategy can be developed accordingly. • Also, the Credit Risk for these customers are low since they always payback compared to their overall spending.

Conclusion

In conclusion, our partnership with DFBanker marks a significant step forward in solving their data-related challenges. By integrating clustering analytics, we've helped DFBanker tailor marketing strategies, and enhance customer experience.

With the insights provided by our solutions, DFBanker is now better equipped to make informed decisions and drive business growth. The tools developed, including the Clustering Analysis report, will support DFBanker in navigating the competitive credit card industry.

Looking ahead, we're committed to continuing our support for DFBanker and ensuring their success in leveraging data for business optimization. Together, we're shaping the future of data-driven decision-making in finance and driving positive outcomes for DFBanker and its customers.