

Proyecto de Clusterización

Addison Amin Reyes Cedano, 2021-2026

El objetivo de este proyecto es desarrollar un sistema que estime el valor de un dato usando un modelo de Machine Learning no supervisado para clusterización. Para mi proyecto tome la decisión de usar el [dataset](#) de tarjetas de créditos de un banco en donde el dataset tiene como columnas balance de la tarjeta, ID de la tarjeta, frecuencia con la que cambia el balance, compras hechas, total de compras de la tarjeta, la compra más cara, cantidad de veces con la que se paga la tarjeta de crédito, límite de la tarjeta, etc... Lo primero que hice fue importar librerías y leer el archivo “.csv”.

▼ Librerías y Dataset

Dataset: <https://www.kaggle.com/datasets/arjunbasin2013/ccdata>

```
[1] 1 from sklearn.cluster import AgglomerativeClustering
2 from yellowbrick.cluster import KElbowVisualizer
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.mixture import GaussianMixture
5 from sklearn.decomposition import PCA
6 from sklearn.cluster import MeanShift
7 from sklearn.cluster import DBSCAN
8 from sklearn.cluster import KMeans
9 from sklearn.cluster import OPTICS
10 import matplotlib.pyplot as plt
11 from matplotlib import colors
12 import seaborn as sns
13 import pandas as pd
14 import numpy as np
```

```
[2] 1 df = pd.read_csv('/content/CC_GENERAL.csv')
2 df.head(5)
```

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	
0	C10001	40.900749	0.818182	95.40	0.00	95.4	0.000000	0.166667	0.000000	0.000000	0.0
1	C10002	3202.467416	0.909091	0.00	0.00	0.0	6442.945483	0.000000	0.000000	0.000000	0.0
2	C10003	2495.148862	1.000000	773.17	773.17	0.0	0.000000	1.000000	1.000000	1.000000	0.0
3	C10004	1666.670542	0.636364	1499.00	1499.00	0.0	205.788017	0.083333	0.083333	0.083333	0.0
4	C10005	817.714335	1.000000	16.00	16.00	0.0	0.000000	0.083333	0.083333	0.083333	0.0

Una vez tenido el dataset en un dataframe empecé el proceso de ingeniería de características, elimine las filas y columnas nulas, pase todas las columnas a valores numéricos y por ultimo revise que todos los datos estuvieran correctamente limpios.

Ingeniería de características

0s 1 df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CUST_ID          8950 non-null   object  
 1   BALANCE           8950 non-null   float64 
 2   BALANCE_FREQUENCY 8950 non-null   float64 
 3   PURCHASES         8950 non-null   float64 
 4   ONEOFF_PURCHASES 8950 non-null   float64 
 5   INSTALLMENTS_PURCHASES 8950 non-null   float64 
 6   CASH_ADVANCE      8950 non-null   float64 
 7   PURCHASES_FREQUENCY 8950 non-null   float64 
 8   ONEOFF_PURCHASES_FREQUENCY 8950 non-null   float64 
 9   PURCHASES_INSTALLMENTS_FREQUENCY 8950 non-null   float64 
 10  CASH_ADVANCE_FREQUENCY 8950 non-null   float64 
 11  CASH_ADVANCE_TRX 8950 non-null   int64   
 12  PURCHASES_TRX    8950 non-null   int64   
 13  CREDIT_LIMIT     8949 non-null   float64 
 14  PAYMENTS          8950 non-null   float64 
 15  MINIMUM_PAYMENTS 8637 non-null   float64 
 16  PRC_FULL_PAYMENT 8950 non-null   float64 
 17  TENURE            8950 non-null   int64   

dtypes: float64(14), int64(3), object(1)
memory usage: 1.2+ MB
```

0s 1 df.isna().any()

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

0s [9] 1 df.head(5)

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	
0	10001	40.900749	0.818182	95.40	0.00		95.40	0.000000	0.166667	0.000000	0.0
1	10002	3202.467416	0.909091	0.00	0.00		0.00	6442.945483	0.000000	0.000000	0.0
2	10003	2495.148862	1.000000	773.17	773.17		0.00	0.000000	1.000000	1.000000	0.0
4	10005	817.714335	1.000000	16.00	16.00		0.00	0.000000	0.083333	0.083333	0.0
5	10006	1809.828751	1.000000	1333.28	0.00		1333.28	0.000000	0.666667	0.000000	0.5

0s 1 df.isna().any()

```
CUST_ID           False
BALANCE           False
BALANCE_FREQUENCY False
PURCHASES         False
ONEOFF_PURCHASES False
INSTALLMENTS_PURCHASES False
CASH_ADVANCE      False
PURCHASES_FREQUENCY False
ONEOFF_PURCHASES_FREQUENCY False
PURCHASES_INSTALLMENTS_FREQUENCY False
CASH_ADVANCE_FREQUENCY False
CASH_ADVANCE_TRX False
PURCHASES_TRX    False
CREDIT_LIMIT     True
PAYMENTS          False
MINIMUM_PAYMENTS True
PRC_FULL_PAYMENT False
TENURE            False
dtype: bool
```

0s [5] 1 df = df.dropna()

```
[7] 1 df['CUST_ID']
0   C10001
1   C10002
2   C10003
4   C10005
5   C10006
...
8943  C19184
8945  C19186
8947  C19188
8948  C19189
8949  C19190
Name: CUST_ID, Length: 8636, dtype: object
```

```
0s 1 cust_id = []
2 for i in df['CUST_ID']:
3   i = int(i[1:])
4   cust_id.append(i)
5
6 df['CUST_ID'] = cust_id
```

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

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.replace.html>

```

1 df.info()
2
3 <class 'pandas.core.frame.DataFrame'>
4 Int64Index: 8636 entries, 0 to 8949
5 Data columns (total 18 columns):
6 #   Column           Non-Null Count  Dtype  
7 --  -- 
8 0   CUST_ID          8636 non-null    int64  
9 1   BALANCE          8636 non-null    float64
10 2   BALANCE_FREQUENCY 8636 non-null    float64
11 3   PURCHASES        8636 non-null    float64
12 4   ONEOFF_PURCHASES 8636 non-null    float64
13 5   INSTALLMENTS_PURCHASES 8636 non-null    float64
14 6   CASH_ADVANCE     8636 non-null    float64
15 7   PURCHASES_FREQUENCY 8636 non-null    float64
16 8   ONEOFF_PURCHASES_FREQUENCY 8636 non-null    float64
17 9   PURCHASES_INSTALLMENTS_FREQUENCY 8636 non-null    float64
18 10  CASH_ADVANCE_FREQUENCY 8636 non-null    float64
19 11  CASH_ADVANCE_TRX 8636 non-null    int64  
20 12  PURCHASES_TRX   8636 non-null    int64  
21 13  CREDIT_LIMIT    8636 non-null    float64
22 14  PAYMENTS        8636 non-null    float64
23 15  MINIMUM_PAYMENTS 8636 non-null    float64
24 16  PRC_FULL_PAYMENT 8636 non-null    float64
25 17  TENURE          8636 non-null    int64  
26 dtypes: float64(14), int64(4)
27 memory usage: 1.3 MB

```



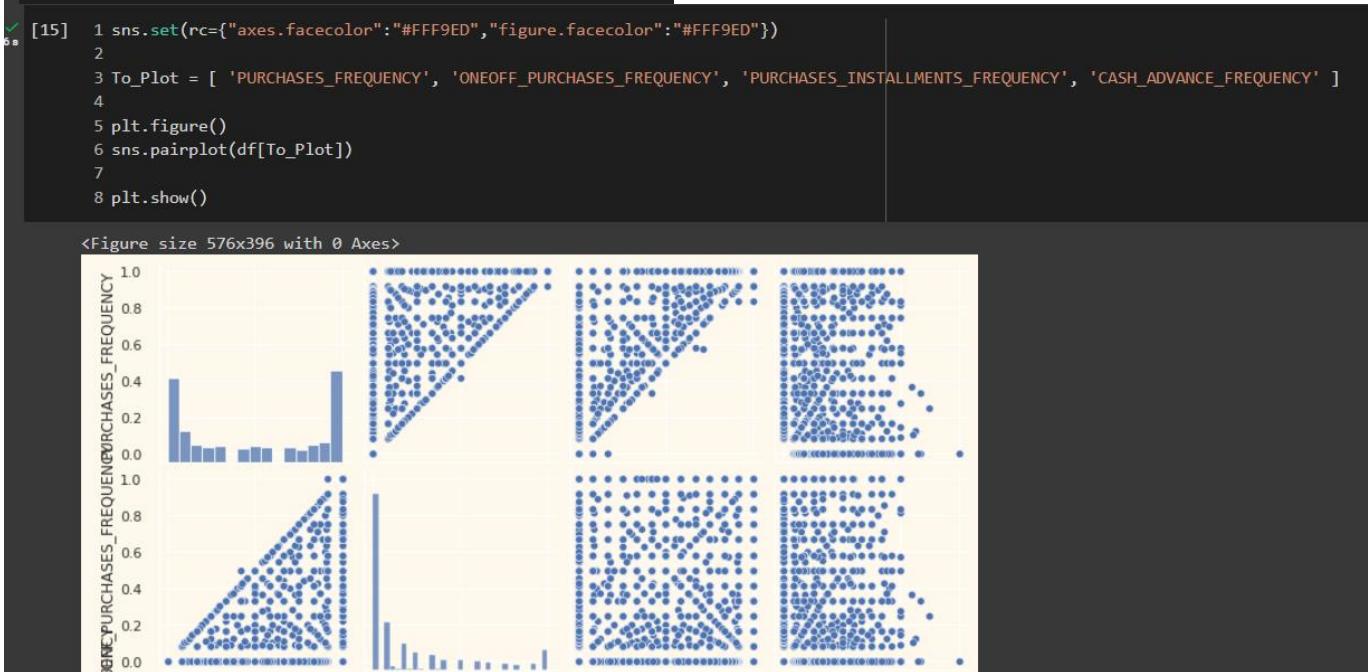
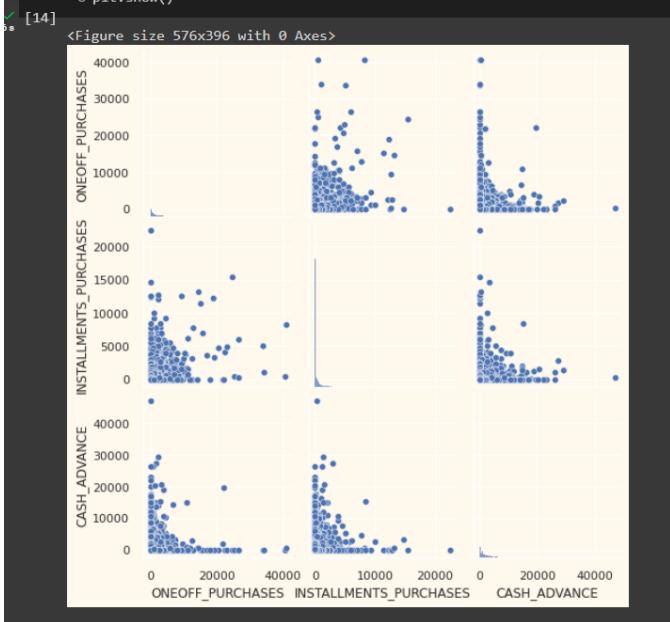
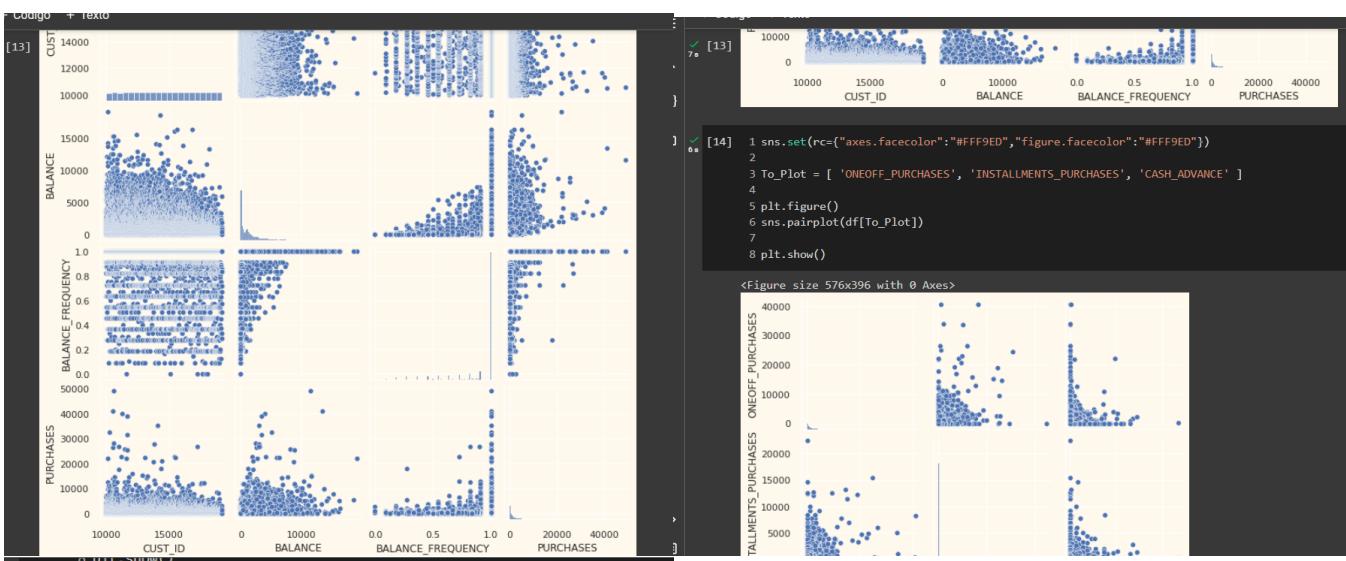
```

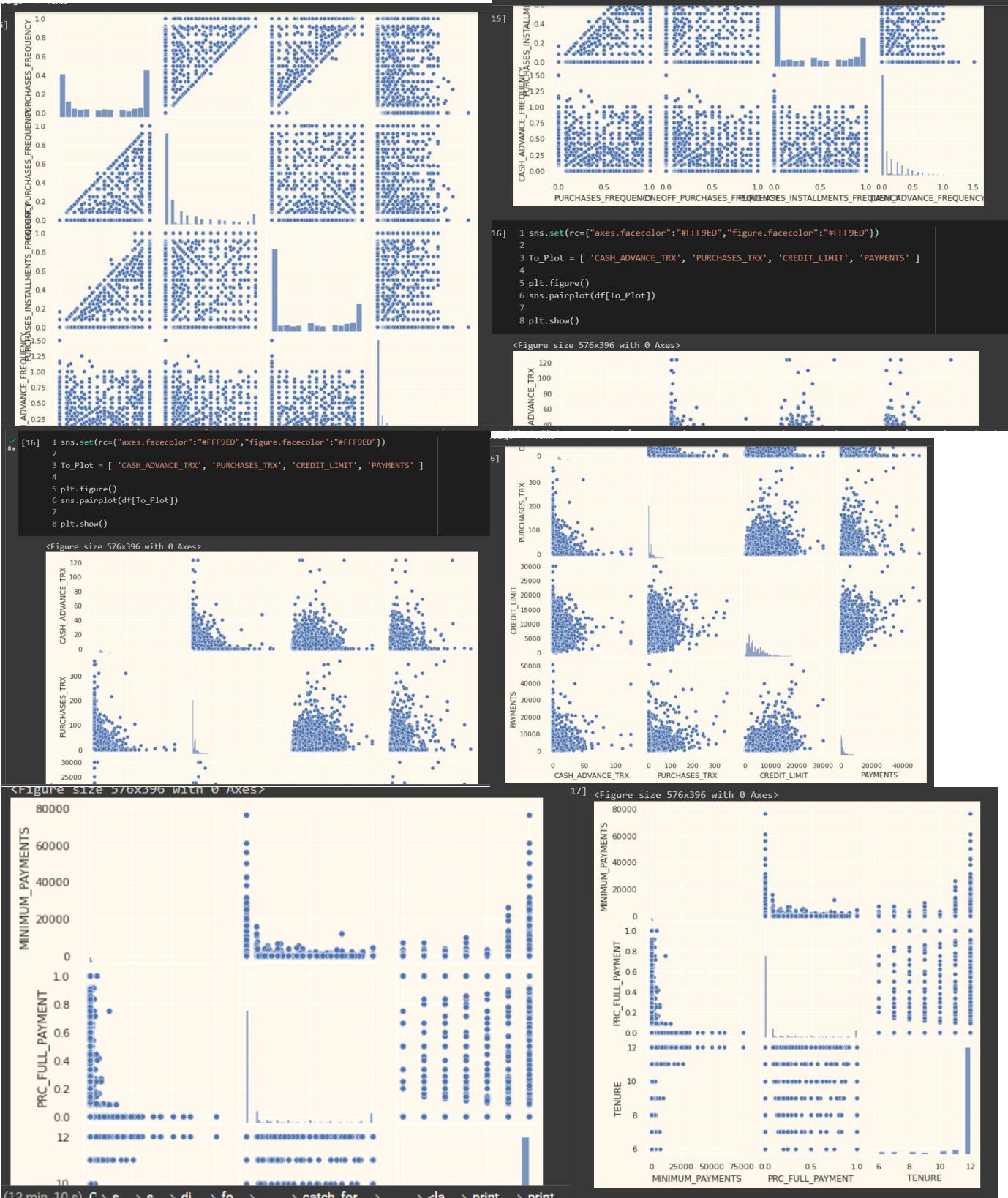
1 df.describe()
2
3   CUST_ID      BALANCE  BALANCE_FREQUENCY  PURCHASES  ONEOFF_PURCHASES  INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INST/
4
5 count  8636.000000  8636.000000  8636.000000  8636.000000  8636.000000  8636.000000  8636.000000  8636.000000  8636.000000
6 mean   14602.540875  1601.224893  0.895035  1025.433874  604.901438  420.843533  994.175523  0.496000  0.205909
7 std    2632.772750  2095.571300  0.207697  2167.107984  1684.307803  917.245182  2121.458303  0.401273  0.300054
8 min    10001.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
9 25%   12337.750000  148.095189  0.909091  43.367500  0.000000  0.000000  0.000000  0.083333  0.000000
10 50%  14592.500000  916.855459  1.000000  375.405000  44.995000  94.785000  0.000000  0.500000  0.083333
11 75%  16885.250000  2105.195853  1.000000  1145.980000  599.100000  484.147500  1132.385490  0.916667  0.333333
12 max   19190.000000  19043.138560  1.000000  49039.570000  40761.250000  22500.000000  47137.211760  1.000000  1.000000

```

Una vez realizada la Ingeniería de características sigue el Análisis descriptivo de la data (EDA) en donde grafico todos los datos del dataset y analizo los datos, aquí los graficos hechos con matplotlib y seaborn.





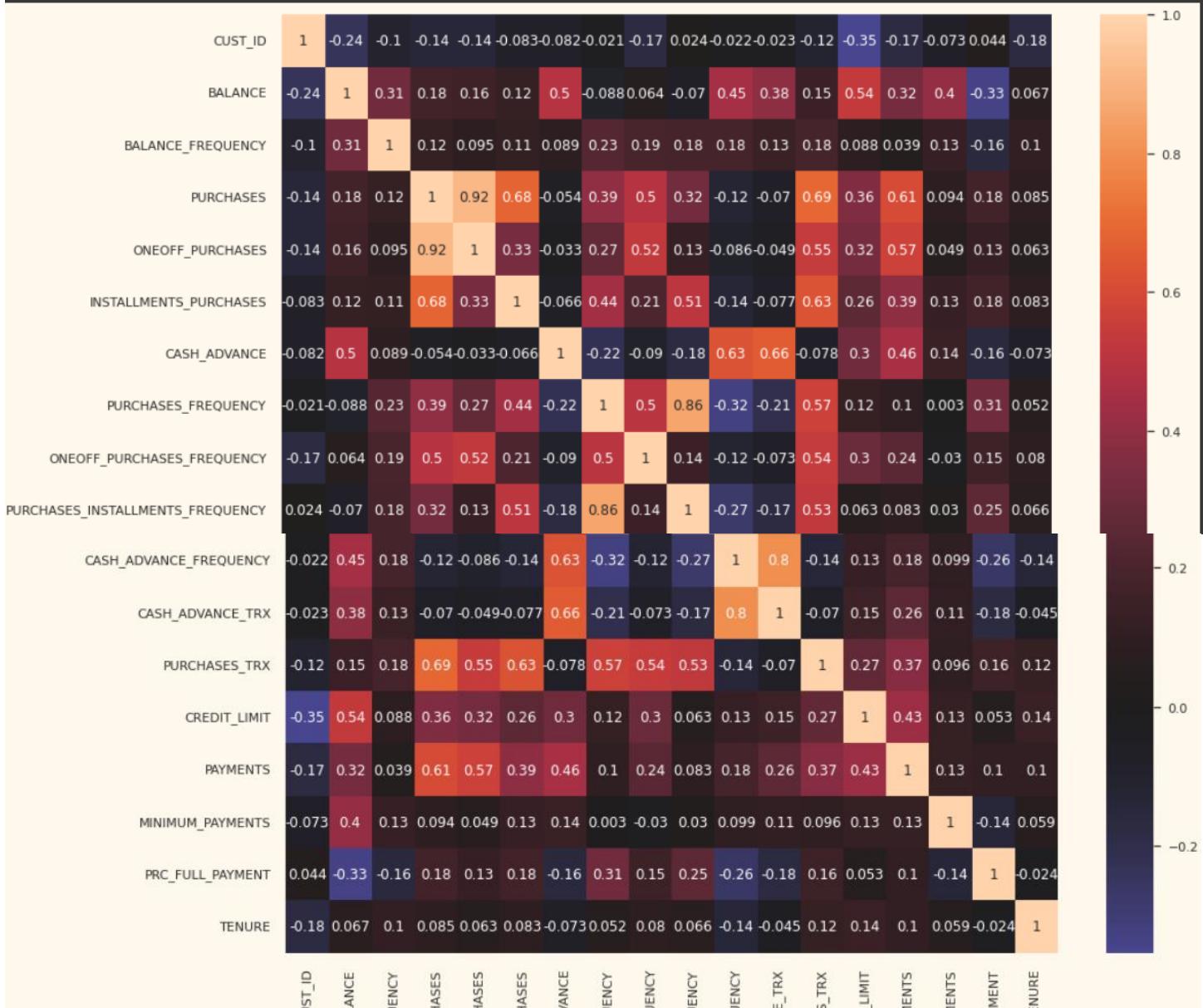


```

1 corrrmat= df.corr()
2 plt.figure(figsize=(15,15))
3 sns.heatmap(corrmat, annot=True, center=0)

```

Axes: >



Despues de analizar los datos y ver la correlacion que hay entre las columnas cree una copia del dataframe y escala los datos a un dataframe de tres columnas con PCA y cree una proyección del dataframe a 3D para poder visualizar los datos.

```

[19] 1 ds = df.copy()
2
3 scaler = StandardScaler()
4 scaler.fit(ds)
5 scaled_ds = pd.DataFrame(scaler.transform(ds), columns=ds.columns)

[20] 1 scaled_ds.head(5)

  CUST_ID BALANCE BALANCE_FREQUENCY PURCHASES ONEOFF_PURCHASES INSTALLMENTS_PURCHASES CASH_ADVANCE PURCHASES_FREQUENCY ONEOFF_PURCHASES_FREQUENCY PURCHASES_INSTALLMENTS_FREQUENCY CASH_ADVANCE_FREQUENCY
0 -1.747894 -0.744625 -0.370047 -0.429184 -0.359160 -0.354826 -0.468655 -0.820769 -0.686280 -0.717179
1 -1.747514 0.764152 0.067679 0.473208 -0.359160 -0.458839 2.568556 -1.236139 -0.686280 -0.926522
2 -1.747134 0.426602 0.505405 -0.116413 0.099909 -0.458839 -0.468655 1.256077 2.646651 -0.926522
3 -1.746374 -0.373910 0.505405 -0.465825 -0.349660 -0.458839 -0.468655 -1.028455 -0.408536 -0.926522
4 -1.745995 0.099551 0.505405 0.142062 -0.359160 0.994815 -0.468655 0.425339 -0.686280 0.538882

```

```

[21] 1 pca = PCA(n_components=3)
2 pca.fit(scaled_ds)
3 PCA_ds = pd.DataFrame(pca.transform(scaled_ds), columns=["col1", "col2", "col3"])
4 PCA_ds.describe().T

   count      mean       std      min     25%     50%     75%      max
col1  8636.0 -1.316429e-17 2.161561 -2.952363 -1.464990 -0.465274 0.771716 29.231539
col2  8636.0 5.265717e-17 1.872302 -2.656806 -1.274295 -0.457024 0.767176 24.762281
col3  8636.0 -7.898576e-17 1.233420 -0.923007 -0.799949 0.042830 0.564734 15.321041

```

```

[22] 1 x = PCA_ds["col1"]
2 y = PCA_ds["col2"]
3 z = PCA_ds["col3"]
4
5 fig = plt.figure(figsize=(10,8))
6 ax = fig.add_subplot(111, projection="3d")
7 ax.scatter(x,y,z, c="maroon", marker="o" )
8 ax.set_title("Proyeccion 3D del DataFrame escalado")
9 plt.show()

```

Proyección 3D del DataFrame escalado

```

[22] 1 x = PCA_ds["col1"]
2 y = PCA_ds["col2"]
3 z = PCA_ds["col3"]
4
5 fig = plt.figure(figsize=(10,8))
6 ax = fig.add_subplot(111, projection="3d")
7 ax.scatter(x,y,z, c="maroon", marker="o" )
8 ax.set_title("Proyeccion 3D del DataFrame escalado")
9 plt.show()

```

Proyección 3D del DataFrame escalado

Con el Dataframe escalado entrene los algoritmos K-Means, Mean-shift, Agglomerative, clustering, DBSCAN, OPTICS, Gaussian mixtures.

The figure is a line graph titled "Distortion Score Elbow for KMeans Clustering". The x-axis is labeled "k" and ranges from 2 to 10. The y-axis is labeled "distortion score" and ranges from 0.4 to 60,000. A solid blue line with circular markers shows the distortion score for each value of k. A dashed orange line with triangular markers shows the elbow method's score for each k. A vertical dashed line is drawn at k=5, where the elbow method's score is approximately 28,568.175. The plot area has a light gray background with white grid lines.

k	distortion score (solid blue line)	elbow at k = 5, score (dashed orange line)
2	~60,000	~18,000
3	~45,000	~55,000
4	~35,000	~15,000
5	~28,568.175	~28,568.175
6	~25,000	~22,000
7	~22,000	~28,000
8	~20,000	~25,000
9	~18,000	~23,000
10	~16,000	~22,000

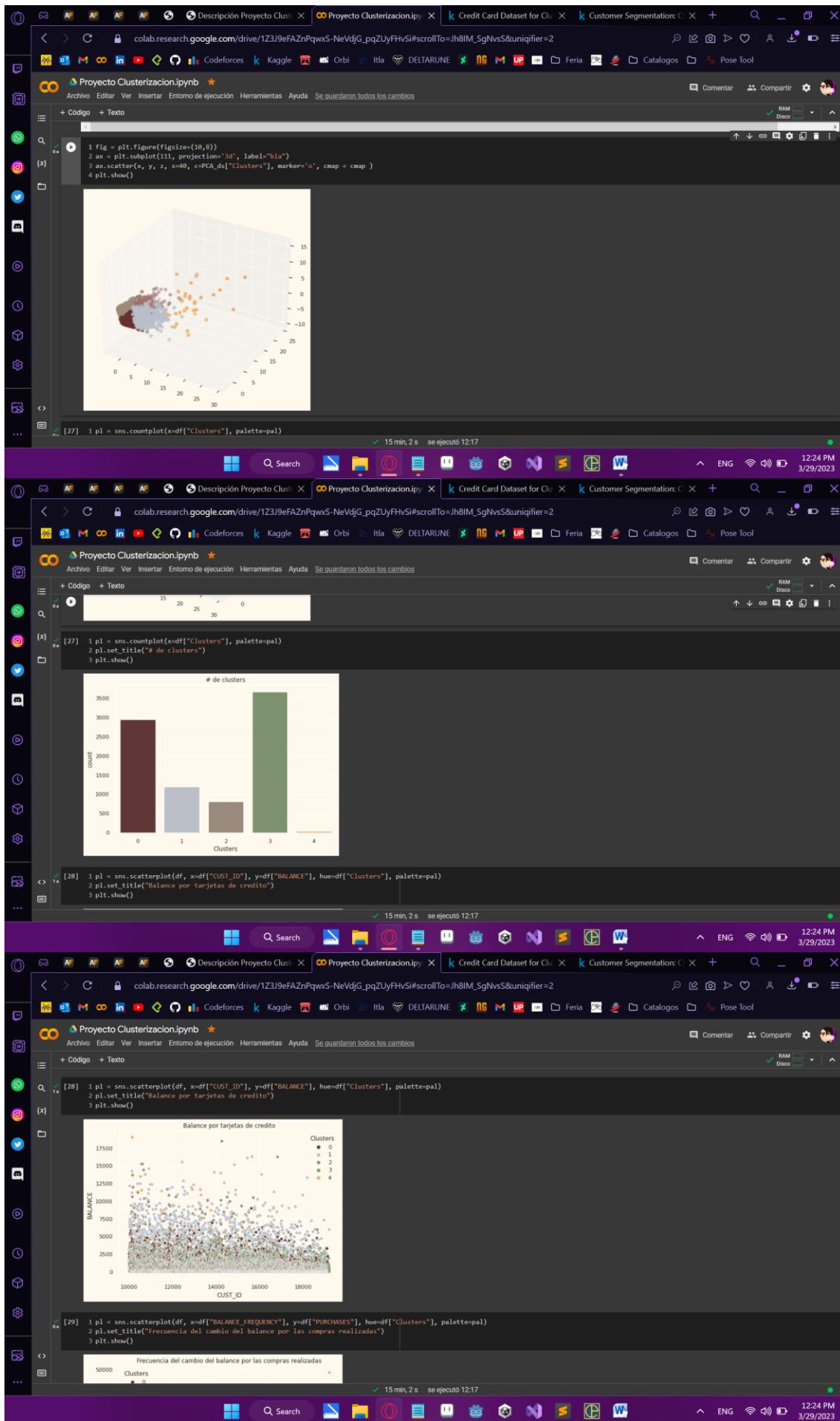
```
<Axes: title={'center': 'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='distortion score'>

[25]: 1 KM = KMeans(n_clusters=5)
2
3 yhat_KM = KM.fit_predict(PCA.ds)
4 PCA.ds["Clusters"] = yhat_KM
5
6 df["Clusters"] = yhat_KM

/usr/local/lib/python3.9/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of 'n_init' will change from 10 to 'auto' in 1.4. Set the value of 'n_init' explicitly to suppress
warnings.warn()

[4]: 1 fig = plt.figure(figsize=(10,8))
2 ax = plt.subplot(111, projection='3d', label="bla")
3 ax.scatter(x, y, z,s=40, c=PCA.ds["Clusters"], marker='o', cmap = cmap )
4 plt.show()

15 min, 2 s  se ejecutó 12:17
```



Colab research.google.com/drive/1Z3J9efAZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=Jh8IM_SgNvsS&uniqifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

[28]

```
1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia del cambio del balance por las compras realizadas")
3 plt.show()
```

Frecuencia del cambio del balance por las compras realizadas

[29]

```
1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALMENTS_PURCHASES"], hue=df["Clusters"], palette=palet)
2 pl.set_title("La compra mas elevada por todas las compras hechas con la tarjeta")
3 plt.show()
```

15 min, 2 s se ejecutó 12:17

12:24 PM 3/29/2023

[30]

```
1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALMENTS_PURCHASES"], hue=df["Clusters"], palette=palet)
2 pl.set_title("La compra mas elevada por todas las compras hechas con la tarjeta")
3 plt.show()
```

La compra mas elevada por todas las compras hechas con la tarjeta

[31]

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Avance de la tarjeta de la tarjeta de crédito con la frecuencias de compras")
3 plt.show()
```

Avance de la tarjeta de la tarjeta de crédito con la frecuencias de compras

15 min, 2 s se ejecutó 12:17

12:24 PM 3/29/2023

[32]

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Avance de la tarjeta de la tarjeta de crédito con la frecuencias de compras")
3 plt.show()
```

Avance de la tarjeta de la tarjeta de crédito con la frecuencias de compras

[33]

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de crédito es pagado con el numero de transacciones hechas")
```

15 min, 2 s se ejecutó 12:17

12:25 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=4X-InAwQ4Gl&unqidifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[x] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de crédito es pagado con el numero de transacciones hechas")
3 plt.show()
```

Frecuencia con la que el dinero de las tarjetas de crédito es pagado con el numero de transacciones hechas

[x] 1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Cantidad de transacciones hechas por el limite de la tarjeta de credito")
3 plt.show()

Cantidad de transacciones hechas por el limite de la tarjeta de credito

[x] 1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Cantidad de transacciones hechas por el limite de la tarjeta de credito")
3 plt.show()

Cantidad de transacciones hechas por el limite de la tarjeta de credito

[x] 1 pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el minimo del pago realizado con la tarjeta")
3 plt.show()

Pagos hechos por el cliente por el minimo del pago realizado con la tarjeta

[x] 1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Cantidad de transacciones hechas por el limite de la tarjeta de credito")
3 plt.show()

Cantidad de transacciones hechas por el limite de la tarjeta de credito

[x] 1 pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el minimo del pago realizado con la tarjeta")
3 plt.show()

Pagos hechos por el cliente por el minimo del pago realizado con la tarjeta

Mean-shift

10 celdas ocultas

15 min, 2 s se ejecutó 12:17

12:25 PM 3/29/2023

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** Colab research.google.com/drive/1Z3j9eFaZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=nXTOpp90Ur&utm_medium=referral
- Toolbar:** Includes icons for back, forward, search, and various file operations.
- Header:** Proyecto Clusterización.ipynb (highlighted), Archivo, Editar, Ver, Insertar, Entorno de ejecución, Herramientas, Ayuda, Se guardaron todos los cambios.
- Section:** Mean-shift
- Code Cells:**
 - [35]:

```
1 MS = MeanShift(seeds=PCA_ds.sample(4).iloc[:, :-1].to_numpy(), bandwidth=0.01)
2
3 yhat_MS = MS.fit_predict(PCA_ds.iloc[:, :-1])
4
5 PCA_ds["Clusters"] = yhat_MS
6 df["Clusters"] = yhat_MS
```
 - [36]:

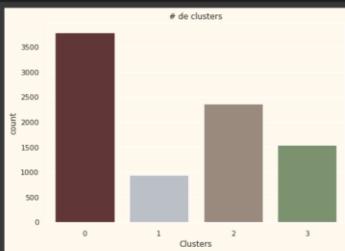
```
1 fig = plt.figure(figsize=(10,8))
2 ax = plt.subplot(111, projection='3d', label='bla')
3 ax.scatter(x, y, z, s=40, c=PCA_ds["Clusters"], marker='o', cmap = cmap)
4 plt.show()
```
- Output:** The output of cell [36] is a 3D scatter plot showing data points colored by their assigned cluster (0, 1, or 2).



✓ 15 min, 2 s se ejecutó 12:11

A screenshot of a web browser window. The address bar shows a URL from Google Colab. The main content area displays a Jupyter Notebook titled "Proyecto Clusterizacion.ipynb". The code in the notebook is as follows:

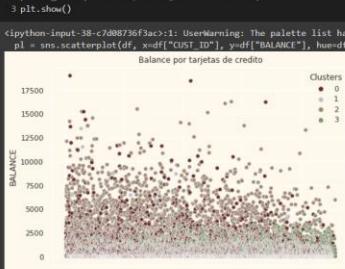
```
[37] 1 plt = sns.countplot(x=edff["Clusters"], palette=palet  
2 plt.set_xlabel("# de clusters")  
3 plt.show()
```



A screenshot of a Jupyter Notebook interface. The title bar shows tabs for 'Descripción Proyecto Clust...', 'Proyecto Clusterización.ipynb', 'Credit Card Dataset for Clu...', and 'Customer Segmentation...'. The main area displays the notebook file 'Proyecto Clusterización.ipynb'. The code cell [37] contains:

```
1 pl = sns.scatterplot(df, x=df['CUST_ID'], y=df['BALANCE'], hue=df['clusters'], palette=palettes)
2 pl.set_title('Balance por tarjetas de crédito')
```

The output cell [38] shows the resulting scatter plot with three distinct clusters labeled 0, 1, and 2. The x-axis is 'CUST_ID' and the y-axis is 'BALANCE'. A color bar at the bottom indicates the 'Clusters' values.



1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=palet) 15 min, 2 s se ejecutó 12:17

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=nXTO0pp9OUrJ&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[38] 10000 12000 14000 16000 18000 CUST_ID
```

```
[x] [39] 1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Frecuencia del cambio del balance por las compras realizadas")
3 plt.show()
```

<ipython-input-39-7eb4f67eecl>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=pal)
```

```
[40] 1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALLMENTS_PURCHASES"], hue=df["Clusters"], palette=pal)
2 pl.set_title("La compra mas elevada por todas las compras hechas con la tarjeta")
3 plt.show()
```

<ipython-input-40-c564f449c067>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALLMENTS_PURCHASES"], hue=df["Clusters"], palette=pal)
```

```
[x] [41] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Avance de la tarjeta de la tarjeta de credito con la frecuencias de compras")
3 plt.show()
```

<ipython-input-41-9cb5da36c895>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=pal)
```

```
[42] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de credito es pagado con el numero de transacciones hechas")
3 plt.show()
```

<ipython-input-42-6dbd8bc07af5>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

15 min, 2 s se ejecutó 12:17

12:25 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=nXTO0pp9OUrJ&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
BALANCE_FREQUENCY
```

```
[x] [40] 1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALLMENTS_PURCHASES"], hue=df["Clusters"], palette=pal)
2 pl.set_title("La compra mas elevada por todas las compras hechas con la tarjeta")
3 plt.show()
```

<ipython-input-40-c564f449c067>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALLMENTS_PURCHASES"], hue=df["Clusters"], palette=pal)
```

```
[x] [41] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Avance de la tarjeta de la tarjeta de credito con la frecuencias de compras")
3 plt.show()
```

<ipython-input-41-9cb5da36c895>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=pal)
```

```
[42] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de credito es pagado con el numero de transacciones hechas")
3 plt.show()
```

<ipython-input-42-6dbd8bc07af5>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

15 min, 2 s se ejecutó 12:17

12:26 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=nXTO0pp9OUrJ&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
PURCHASES_FREQUENCY
```

```
[x] [41] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Avance de la tarjeta de la tarjeta de credito con la frecuencias de compras")
3 plt.show()
```

<ipython-input-41-9cb5da36c895>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=pal)
```

```
[42] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de credito es pagado con el numero de transacciones hechas")
3 plt.show()
```

<ipython-input-42-6dbd8bc07af5>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

15 min, 2 s se ejecutó 12:17

12:26 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=nXTO0pp9OurJ&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

[41]

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de crédito es pagado con el numero de transacciones hechas")
3 plt.show()
```

<ipython-input-42-6dd8bc87af5>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)

Frecuencia con la que el dinero de las tarjetas de crédito es pagado con el numero de transacciones hechas

120
100
80
60
40
20
0

CASH_ADVANCE_FREQUENCY

Clusters
● 0
○ 1
△ 2
◆ 3

0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4

15 min, 2 s se ejecutó 12:17

12:26 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=nXTO0pp9OurJ&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

[42]

```
1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Cantidad de transacciones hechas por el límite de la tarjeta de crédito")
3 plt.show()
```

<ipython-input-43-b3ea8c905d14>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)

Cantidad de transacciones hechas por el límite de la tarjeta de crédito

30000
25000
20000
15000
10000

PURCHASES_TRX

Clusters
● 0
○ 1
△ 2
◆ 3

0 50 100 150 200 250 300 350

15 min, 2 s se ejecutó 12:17

12:26 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=nXTO0pp9OurJ&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

[43]

```
1 pl = sns.scatterplot(df, x=df["PURCHASES"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta")
3 plt.show()
```

<ipython-input-44-d4b5571d52ad>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

pl = sns.scatterplot(df, x=df["PURCHASES"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)

Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta

80000
70000
60000
50000
40000

PURCHASES

Clusters
● 0
○ 1
△ 2
◆ 3

0 50 100 150 200 250 300 350

15 min, 2 s se ejecutó 12:17

12:26 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=EbMpC-ADNr1c&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

[44] 0 10000 20000 30000 40000 50000 PAYMENTS

[x] Agglomerative clustering

[45]

```
1 AC = AgglomerativeClustering(n_clusters=4)
2
3 yhat_AC = AC.fit_predict(PCA_ds)
4 PCA.ds["Clusters"] = yhat_AC
5
6 df[["Clusters"]] = yhat_AC
```

[46]

```
1 fig = plt.figure(figsize=(10,8))
2 ax = plt.subplot(111, projection='3d', label="bla")
3 ax.scatter(x, y, z, s=40, c=PCA.ds["Clusters"], marker='o', cmap = cmap )
4 plt.show()
```

15 min, 2 s se ejecutó 12:17

12:26 PM 3/29/2023

[46]

```
1 fig = plt.figure(figsize=(10,8))
2 ax = plt.subplot(111, projection='3d', label="bla")
3 ax.scatter(x, y, z, s=40, c=PCA.ds["Clusters"], marker='o', cmap = cmap )
4 plt.show()
```

15 min, 2 s se ejecutó 12:17

[47]

```
1 pl = sns.countplot(x=df[["Clusters"]], palette=pa)
```

de clusters

Clusters	Count
0	~3500
1	~3000
2	~1000
3	~500

15 min, 2 s se ejecutó 12:17

[47]

```
1 pl = sns.countplot(x=df[["Clusters"]], palette=pa)
2 pl.set_title("# de clusters")
3 plt.show()
```

de clusters

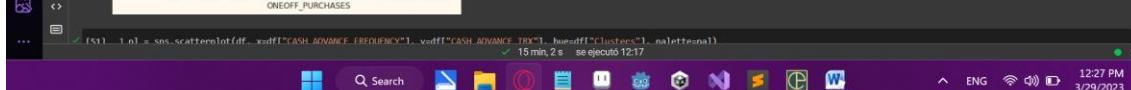
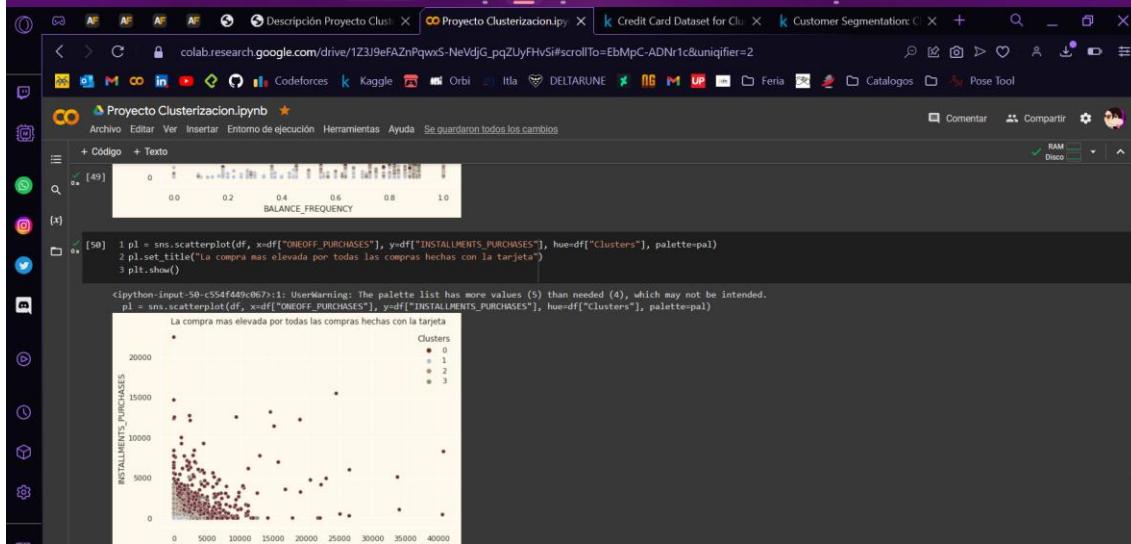
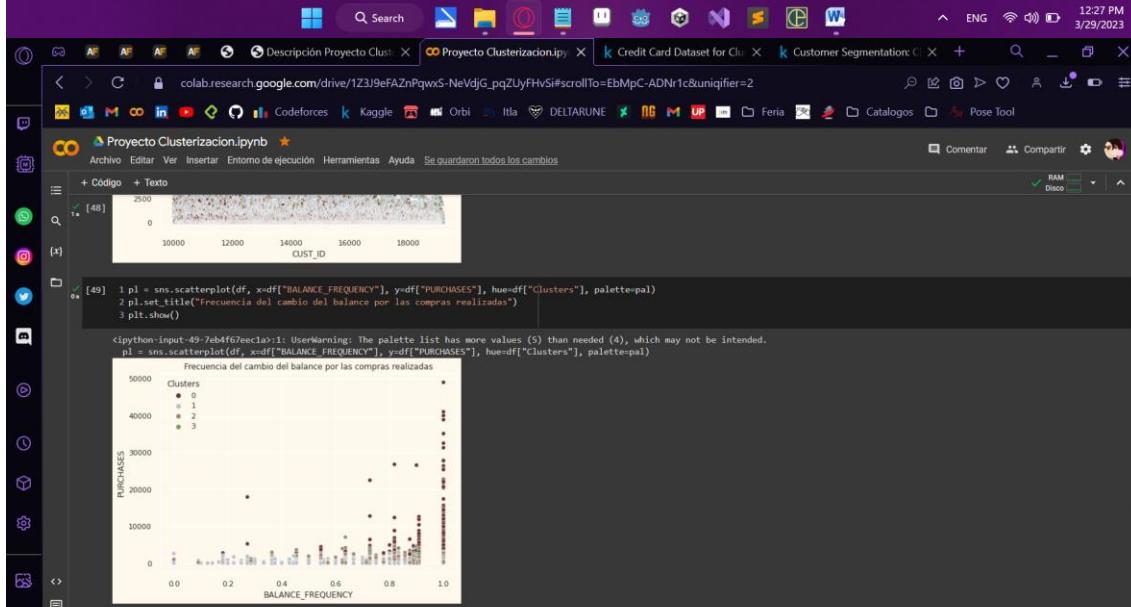
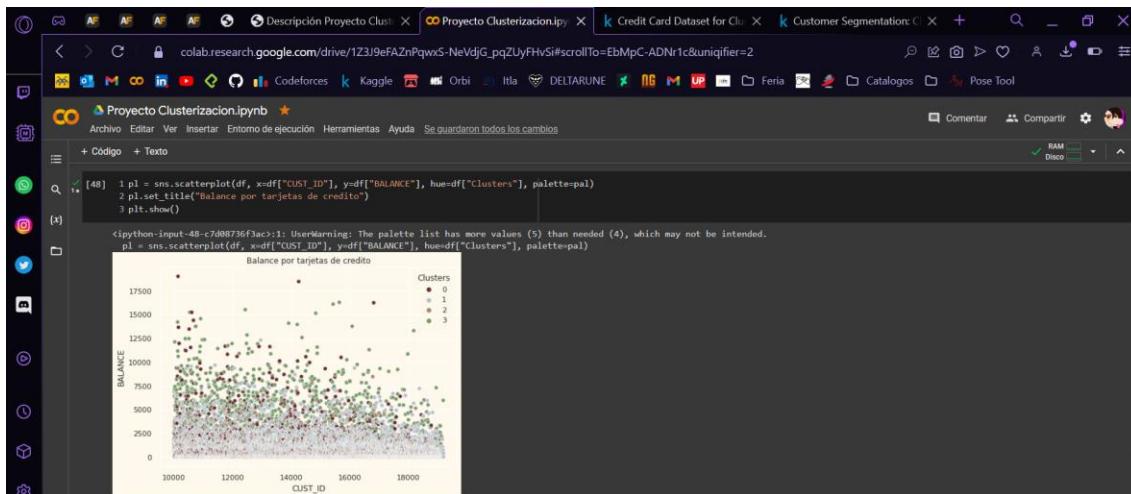
Clusters	Count
0	~3500
1	~3000
2	~1000
3	~500

15 min, 2 s se ejecutó 12:17

[48]

```
1 pl = sns.scatterplot(df, x=df[["CUST_ID"]], y=df[["BALANCE"]], hue=df[["Clusters"]], palette=pa)
2 pl.set_title("Balance por tarjetas de crédito")
3 plt.show()
```

12:26 PM 3/29/2023



Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=EbMpC-ADNr1c&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[50] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de crédito es pagado con el numero de transacciones hechas")
3 plt.show()
```

<ipython-input-51-6dbd8bc87af>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)

Frecuencia con la que el dinero de las tarjetas de crédito es pagado con el numero de transacciones hechas

```
[51] 1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Cantidad de transacciones hechas por el limite de la tarjeta de credito")
3 plt.show()
```

<ipython-input-52-b3ea8c905d14>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)

Cantidad de transacciones hechas por el limite de la tarjeta de credito

```
[53] 1 pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el minimo del pago realizado con la tarjeta")
3 plt.show()
```

<ipython-input-53-d40957da52ad>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)

Pagos hechos por el cliente por el minimo del pago realizado con la tarjeta

DBSCAN

410 celdas ocultas

15 min, 2 s se ejecutó 12:17

12:27 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | - | X

colab.research.google.com/drive/1Z3J9efAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=15C8Si3_j4jC&uniquifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

DBSCAN

```
[54]: 1 DB = DBSCAN(eps=0.8, min_samples=15, leaf_size=50)
2
3 yhat_DB = DB.fit_predict(PCA_ds)
4 PCA_ds["Clusters"] = yhat_DB
5
6 df["Clusters"] = yhat_DB
```

```
[55]: 1 fig = plt.figure(figsize=(10,6))
2 ax = plt.subplot(111, projection='3d', label="bla")
3 ax.scatter(x, y, z, s=40, c=PCA_ds["Clusters"], marker='o', cmap = cmap )
4 plt.show()
```

15 min, 2 s se ejecutó 12:17

12:27 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | - | X

colab.research.google.com/drive/1Z3J9efAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=15C8Si3_j4jC&uniquifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[54]: 6 df["Clusters"] = yhat_DB
```

```
[55]: 1 fig = plt.figure(figsize=(10,6))
2 ax = plt.subplot(111, projection='3d', label="bla")
3 ax.scatter(x, y, z, s=40, c=PCA_ds["Clusters"], marker='o', cmap = cmap )
4 plt.show()
```

15 min, 2 s se ejecutó 12:17

12:27 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | - | X

colab.research.google.com/drive/1Z3J9efAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=15C8Si3_j4jC&uniquifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[55]: 1 plt = sns.countplot(x=df["Clusters"], palette=palet)
```

```
[56]: 1 plt.set_title("# de clusters")
2 plt.show()
```

Clusters	Count
-1	~300
0	~3200
1	~800
2	~500
3	~500

```
[57]: 1 plt = sns.scatterplot(df, x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=palet)
2 plt.set_title("Balance por tarjetas de crédito")
3 plt.show()
```

15 min, 2 s se ejecutó 12:17

12:28 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | ⌂ | ⌂ | ⌂

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=15C8Si3_j4jC&uniquifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[57]: 1 pl = sns.scatterplot(df, x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Balance por tarjetas de crédito")
3 plt.show()
```

Balance por tarjetas de crédito

BALANCE

CUST_ID

Clusters

17500
15000
12500
10000
7500
5000
2500
0

```
[58]: 1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Frecuencia del cambio del balance por las compras realizadas")
3 plt.show()
```

Frecuencia del cambio del balance por las compras realizadas

PURCHASES

BALANCE_FREQUENCY

Clusters

50000
40000
30000
20000
10000
0

```
[59]: 1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALMENTS_PURCHASES"], hue=df["Clusters"], palette=pal)
2 pl.set_title("La compra mas elevada por todas las compras hechas con la tarjeta")
3 plt.show()
```

La compra mas elevada por todas las compras hechas con la tarjeta

INSTALMENTS_PURCHASES

ONEOFF_PURCHASES

Clusters

20000
15000
10000
5000
0

```
[60]: 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["clusters"], palette=pa)
2 pl.set_title("Avance de la tarjeta de la tarjeta de crédito con la frecuencias de compras")
3 plt.show()
```

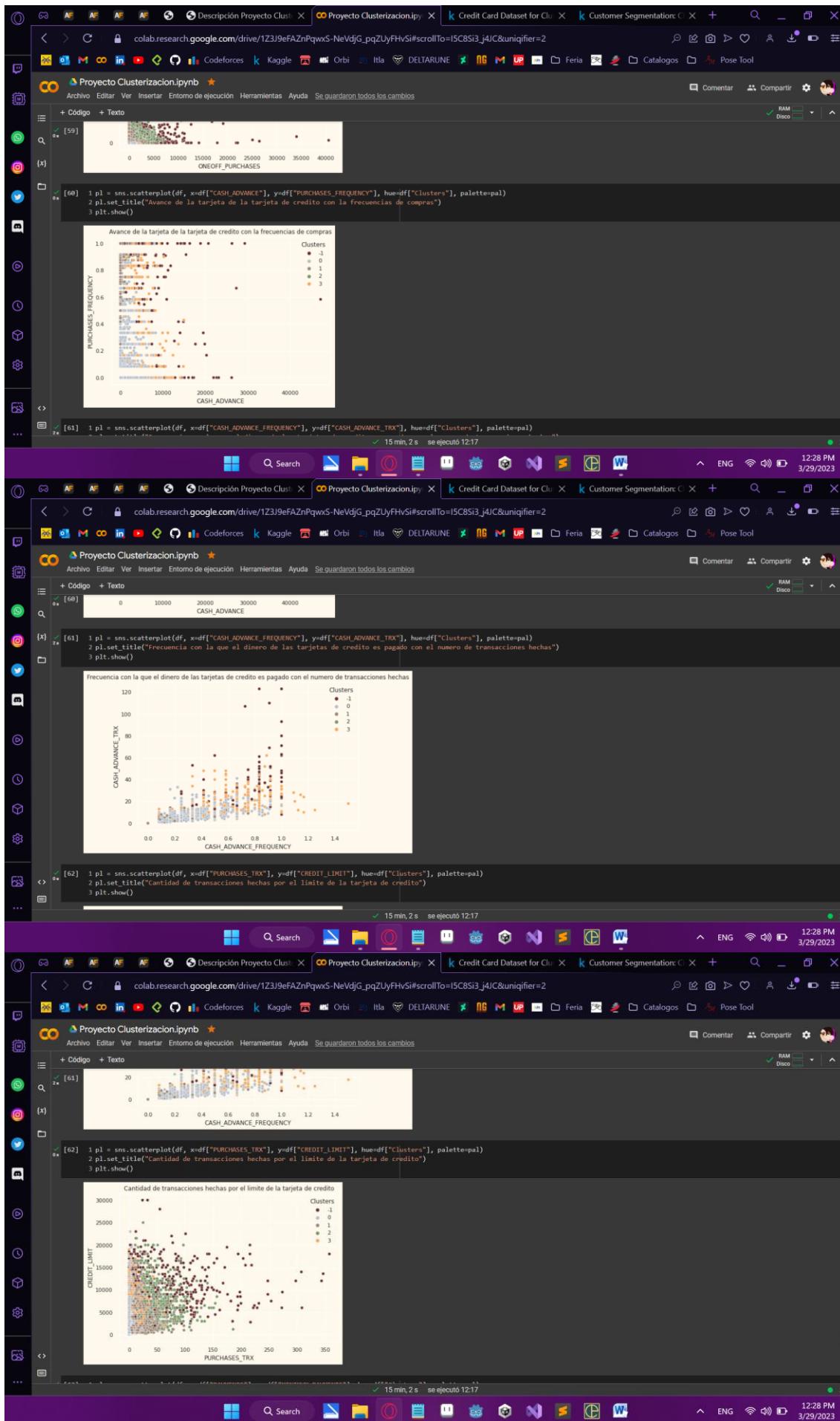
Avance de la tarjeta de la tarjeta de crédito con la frecuencias de compras

15 min, 2 s se ejecutó 12:17

RAM Discos

Commentar Compartir

12:28 PM 3/29/2023



The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** Proyecto Clusterización.ipynb
- Code Cell 62:** Displays the code for generating a scatter plot.
- Code Cell 63:** Displays the resulting scatter plot titled "Pagos hechos por el cliente por el minimo del pago realizado con la tarjeta". The x-axis is labeled "PAYMENTS" and ranges from 0 to 50,000. The y-axis is labeled "MINIMUM PAYMENTS" and ranges from 0 to 80,000. The plot shows data points colored by cluster, with a legend titled "Clusters" indicating four distinct groups: 1 (red), 2 (blue), 3 (green), and 4 (orange).

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** OPTICS
- Header:** 15 min, 2 s se ejecutó 12:17
- Toolbar:** Search, File, Project, Notebook, Cluster, Project Description, Project Clusterización.ipynb, Credit Card Dataset for Clustering, Customer Segmentation, etc.
- Code Cell [64]:**

```
1 OP = OPTICS(min_samples=3, x_min=0.5, max_eps=4, min_cluster_size=0.2)
2
3 yhat_OP = OP.fit_predict(PCA_ds)
4 PCA_ds["Clusters"] = yhat_OP
5
6 df["Clusters"] = yhat_OP
```
- Code Cell [65]:**

```
1 fig = plt.figure(figsize=(10,8))
2 ax = plt.subplot(111, projection='3d', label='bla')
3 ax.scatter(x, y, z, s=40, c=PCA_ds["Clusters"], marker='o', cmap = cmap )
4 plt.show()
```
- Output:** A 3D scatter plot showing data points colored by their assigned cluster. The axes are labeled from -10 to 25.
- Bottom Bar:** 15 min, 2 s se ejecutó 12:17, ENG, 12:28 PM, 3/29/2023

The screenshot shows a Jupyter Notebook interface with several tabs at the top: 'Descripción Proyecto Clust', 'Proyecto Clusterización.ipynb', 'Credit Card Dataset for Clu...', and 'Customer Segmentation'. The main area displays a Python script and its output. The script uses matplotlib to create a 3D scatter plot. The plot shows data points colored by cluster, with axes ranging from -10 to 35. Below the plot, another cell shows a sns.countplot command.

```
[65]: 1 fig = plt.figure(figsize=(10,8))
2 ax = plt.subplot(111, projection='3d', label="bla")
3 ax.scatter(x, y, z, s=10, c=PCA_ds["Clusters"], marker='o', cmap = cmap )
4 plt.show()
```

```
[66]: 1 p1 = sns.countplot(x=df["Clusters"], palette=pa1)
```

Colab research.google.com/drive/1Z3J9efAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=bxP2C1i-Ozy1&unifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

[65]

```
1 pl = sns.countplot(x=df["Clusters"], palette=pal)
2 pl.set_title("# de clusters")
3 plt.show()
```

de clusters

[66]

```
1 pl = sns.countplot(x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Balance por tarjetas de crédito")
3 plt.show()
```

15 min, 2 s se ejecutó 12:17

12:29 PM 3/29/2023

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

[66]

[67]

```
1 pl = sns.scatterplot(df, x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Balance por tarjetas de crédito")
3 plt.show()
```

<ipython-input-67-c7d08736f3ac>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.

pl = sns.scatterplot(df, x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=pal)

Balance por tarjetas de crédito

15 min, 2 s se ejecutó 12:17

12:29 PM 3/29/2023

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

[67]

[68]

```
1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=pal)
2 pl.set_title("Frecuencia del cambio del balance por las compras realizadas")
3 plt.show()
```

<ipython-input-68-7ab4f76ee1c>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.

pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=pal)

Frecuencia del cambio del balance por las compras realizadas

15 min, 2 s se ejecutó 12:17

12:29 PM 3/29/2023

Descripción Proyecto Clus... | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu... | Customer Segmentation:...

colab.research.google.com/drive/1Z3J9eFAZnPqwxS-NeVdjG_pqZUyFhvSi#scrollTo=bxP2C1i-Ozy1&uniquifier=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[68] 0 0.2 0.4 0.6 0.8 1.0
      0.0 0.2 0.4 0.6 0.8 1.0
      BALANCE_FREQUENCY
```

```
[69] 1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALMENTS_PURCHASES"], hue=df["Clusters"], palette=palette)
2 pl.set_title("La compra mas elevada por todas las compras hechas con la tarjeta")
3 plt.show()
```

cipython-input-69-c554f449c067>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.
pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALMENTS_PURCHASES"], hue=df["Clusters"], palette=palette)

```
[70] 0 5000 10000 15000 20000 25000 30000 35000 40000
      ONEOFF_PURCHASES
```

```
[70] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=palette)
2 pl.set_title("Avance de la tarjeta de la tarjeta de credito con la frecuencias de compras")
3 plt.show()
```

cipython-input-70-9cb5da36c895>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.
pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=palette)

```
[71] 0 10000 20000 30000 40000
      CASH_ADVANCE
```

```
[71] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palette)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de credito es pagado con el numero de transacciones hechas")
3 plt.show()
```

cipython-input-71-6dbd0bc07af5>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.
pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palette)

```
[72] 0 10000 20000 30000 40000
      CASH_ADVANCE
```

```
[72] 1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palette)
2 pl.set_title("Cantidad de transacciones hechas por el limite de la tarjeta de credito")
3 plt.show()
```

cipython-input-72-1a2a2a2a2a2a>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.
pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palette)

Descripción Proyecto Clus... X Proyecto Clusterizacion.ipynb X Credit Card Dataset for Clu... X Customer Segmentation: C... X + Search

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

CASH_ADVANCE_FREQUENCY

```
[72]: 1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Cantidad de transacciones hechas por el límite de la tarjeta de crédito")
3 plt.show()

<ipython-input-72-b3eab8c9d5d4>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.
pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
```

Cantidad de transacciones hechas por el límite de la tarjeta de crédito

Clusters
● -1
○ 0

PURCHASES_TRX

CREDIT_LIMIT

```
[73]: 1 pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta")
3 plt.show()
```

<ipython-input-73-d73a7af6ab8c>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.
pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)

Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta

Clusters
● -1
○ 0

PAYMENTS

MINIMUM_PAYMENTS

Descripción Proyecto Clus... X Proyecto Clusterizacion.ipynb X Credit Card Dataset for Clu... X Customer Segmentation: C... X + Search

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[73]: 1 pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta")
3 plt.show()

<ipython-input-73-d73a7af6ab8c>:1: UserWarning: The palette list has more values (5) than needed (2), which may not be intended.
pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
```

Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta

Clusters
● -1
○ 0

PAYMENTS

MINIMUM_PAYMENTS

Gaussian mixtures

[] ↓ 10 celdas ocultas

Descripción Proyecto Clus... X Proyecto Clusterizacion.ipynb X Credit Card Dataset for Clu... X Customer Segmentation: C... X + Search

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

Gaussian mixtures

```
[74]: 1 GM = GaussianMixture(n_components=4)
2
3 yhat_GM = GM.fit_predict(PCA_ds)
4 PCA_ds["Clusters"] = yhat_GM
5
6 df["Clusters"] = yhat_GM
```

```
[75]: 1 fig = plt.figure(figsize=(10,9))
2 ax = plt.subplot(111, projection='3d', label="blo")
3 ax.scatter(x, y, z=s40, c=PCA_ds["Clusters"], marker='o', cmap = cmap)
4 plt.show()
```

Clusters

z

x

y

blo

s40

c

Clusters

fig

ax

label

projection

marker

cmap

plt

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | - | X

colab.research.google.com/drive/1Z3J9efAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=zi_PY4NMO2ge&uniquifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[75]: 1 fig = plt.figure(figsize=(10,8))
2 ax = plt.subplot(111, projection='3d', label="bla")
3 ax.scatter(x, y, z, s=40, c=df["Clusters"], marker='o', cmap = cmap)
4 plt.show()
```

```
[76]: 1 pl = sns.countplot(x=df["clusters"], palette=palet)
```

15 min, 2 s se ejecutó 12:17 12:30 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | - | X

colab.research.google.com/drive/1Z3J9efAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=zi_PY4NMO2ge&uniquifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[75]: 1 pl = sns.countplot(x=df["clusters"], palette=palet)
```

```
[76]: 2 pl.set_title("# de clusters")
3 plt.show()
```

```
[77]: 1 pl = sns.scatterplot(df, x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Balance por tarjetas de crédito")
3 plt.show()
```

15 min, 2 s se ejecutó 12:17 12:30 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | - | X

colab.research.google.com/drive/1Z3J9efAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=zi_PY4NMO2ge&uniquifier=2

Proyecto Clusterizacion.ipynb ★ Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[77]: 1 pl = sns.scatterplot(df, x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Balance por tarjetas de crédito")
3 plt.show()
```

<ipython-input-77-c7d08736f3ac>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
[77]: 1 pl = sns.scatterplot(df, x=df["CUST_ID"], y=df["BALANCE"], hue=df["Clusters"], palette=palet)
```

```
[78]: 1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia del cambio del balance por las compras realizadas")
3 plt.show()
```

<ipython-input-78-7eb4f67ee1ca>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.

```
[78]: 1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=palet)
```

15 min, 2 s se ejecutó 12:17 12:30 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=zi_PY4NMO2ge&uniquer=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[77] 1 pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia del cambio del balance por las compras realizadas")
3 plt.show()
```

```
<ipython-input-78-7eb4f67eecl>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.
pl = sns.scatterplot(df, x=df["BALANCE_FREQUENCY"], y=df["PURCHASES"], hue=df["Clusters"], palette=palet)
```

15 min, 2 s se ejecutó 12:17

12:30 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=zi_PY4NMO2ge&uniquer=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[78] 1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALMENTS_PURCHASES"], hue=df["Clusters"], palette=palet)
2 pl.set_title("La compra mas elevada por todas las compras hechas con la tarjeta")
3 plt.show()
```

```
<ipython-input-79-c554f449c067>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.
pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], y=df["INSTALMENTS_PURCHASES"], hue=df["Clusters"], palette=palet)
```

La compra mas elevada por todas las compras hechas con la tarjeta

Clusters

15 min, 2 s se ejecutó 12:17

12:30 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=zi_PY4NMO2ge&uniquer=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[79] 1 pl = sns.scatterplot(df, x=df["ONEOFF_PURCHASES"], hue=df["Clusters"], palette=palet)
2 plt.show()
```

```
<ipython-input-80-9cb5d036c895>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.
pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=palet)
```

Avance de la tarjeta de la tarjeta de credito con la frecuencias de compras

Clusters

15 min, 2 s se ejecutó 12:17

12:30 PM 3/29/2023

Descripción Proyecto Clus | Proyecto Clusterizacion.ipynb | Credit Card Dataset for Clu | Customer Segmentation: | + | Search | - | X

colab.research.google.com/drive/1Z3J9eFAZnPqwx5-NeVdjG_pqZUyFhvSi#scrollTo=zi_PY4NMO2ge&uniquer=2

Proyecto Clusterizacion.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Se guardaron todos los cambios

+ Código + Texto

```
[80] 1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Avance de la tarjeta de la tarjeta de credito con la frecuencias de compras")
3 plt.show()
```

```
<ipython-input-81-9cb5d036c895>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.
pl = sns.scatterplot(df, x=df["CASH_ADVANCE"], y=df["PURCHASES_FREQUENCY"], hue=df["Clusters"], palette=palet)
```

Avance de la tarjeta de la tarjeta de credito con la frecuencias de compras

Clusters

15 min, 2 s se ejecutó 12:17

12:30 PM 3/29/2023

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** Proyecto Clusterización.ipynb
- Code Cell [81]:**

```
1 pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Frecuencia con la que el dinero de las tarjetas de credito es pagado con el numero de transacciones hechas")
3 plt.show()
```
- Output:**

```
<ipython-input-81-6dbdb8bc07af5>:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.
pl = sns.scatterplot(df, x=df["CASH_ADVANCE_FREQUENCY"], y=df["CASH_ADVANCE_TRX"], hue=df["Clusters"], palette=palet)
```
- Scatter Plot:** A scatter plot with 'CASH_ADVANCE_FREQUENCY' on the x-axis (ranging from 0.0 to 1.4) and 'CASH_ADVANCE_TRX' on the y-axis (ranging from 0 to 320). The plot shows data points colored by cluster (0, 1, 2, 3). A legend indicates:
 - Cluster 0: Red circle
 - Cluster 1: Blue square
 - Cluster 2: Green triangle
 - Cluster 3: Orange diamond

The screenshot shows a Jupyter Notebook interface with two code cells and their corresponding plots.

Code Cell 1:

```
[82]: 1 pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Cantidad de transacciones hechas por el límite de la tarjeta de crédito")
3 plt.show()
```

Output 1:

Python input: 82:b-8eab9c95d14t:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.
pl = sns.scatterplot(df, x=df["PURCHASES_TRX"], y=df["CREDIT_LIMIT"], hue=df["Clusters"], palette=palet)

Cantidad de transacciones hechas por el límite de la tarjeta de crédito

A scatter plot titled "Cantidad de transacciones hechas por el límite de la tarjeta de crédito". The x-axis is labeled "PURCHASES_TRX" and ranges from 0 to 350. The y-axis is labeled "CREDIT_LIMIT" and ranges from 0 to 30000. The plot shows a dense cloud of points with a color legend titled "Clusters" indicating five categories: 0 (red), 1 (blue), 2 (green), 3 (orange), and 4 (purple). Most points are clustered between PURCHASES_TRX values of 0-200 and CREDIT_LIMIT values of 0-15000.

Code Cell 2:

```
[83]: 1 pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta")
3 plt.show()
```

Output 2:

Python input: 83:b-8eab9c95d14t:1: UserWarning: The palette list has more values (5) than needed (4), which may not be intended.
pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)

Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta

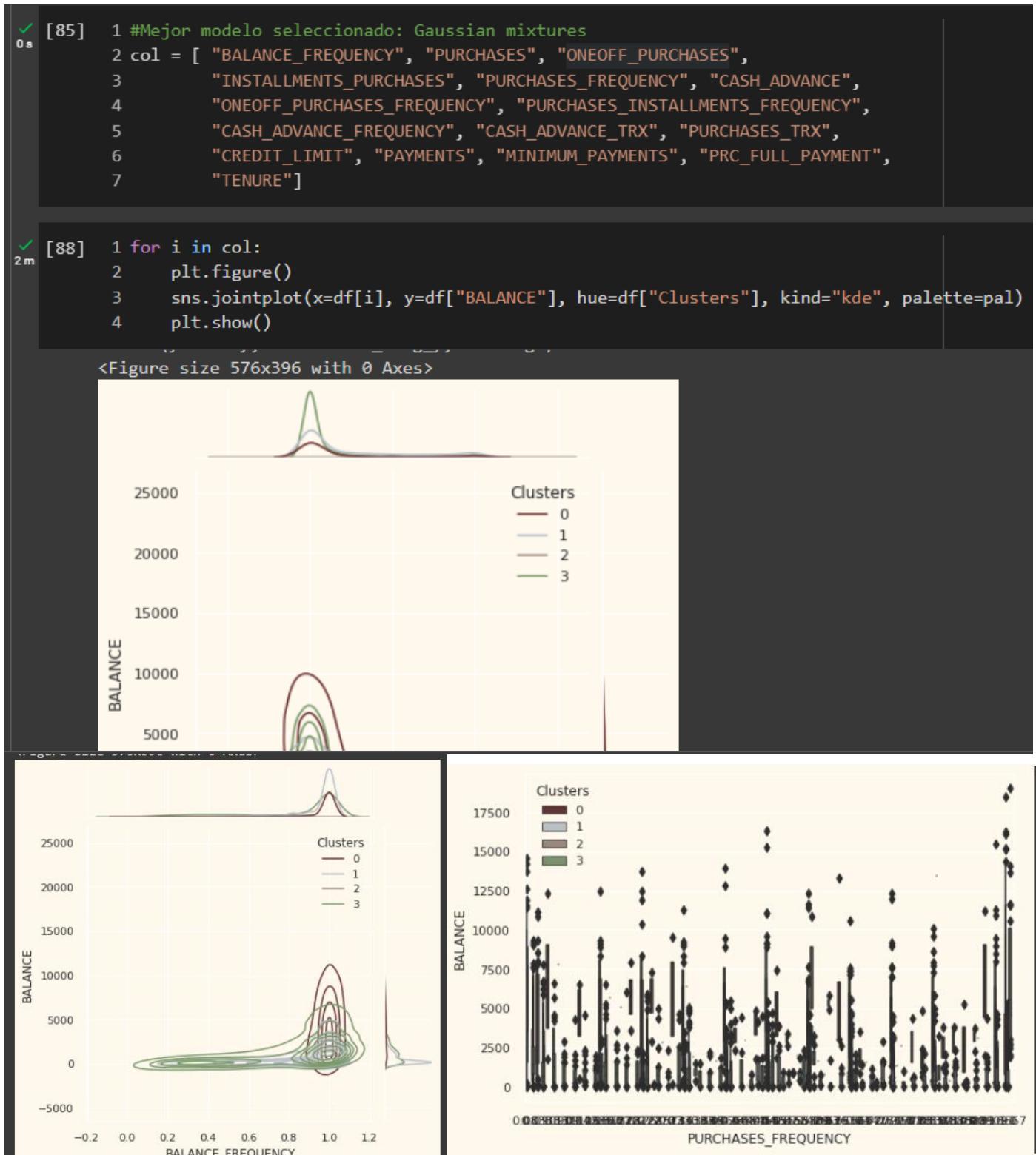
A scatter plot titled "Pagos hechos por el cliente por el mínimo del pago relajado con la tarjeta". The x-axis is labeled "PAYMENTS" and ranges from 0 to 350. The y-axis is labeled "MINIMUM_PAYMENTS" and ranges from 0 to 30000. The plot shows a dense cloud of points with a color legend titled "Clusters" indicating five categories: 0 (red), 1 (blue), 2 (green), 3 (orange), and 4 (purple). Most points are clustered between PAYMENTS values of 0-200 and MINIMUM_PAYMENTS values of 0-15000.

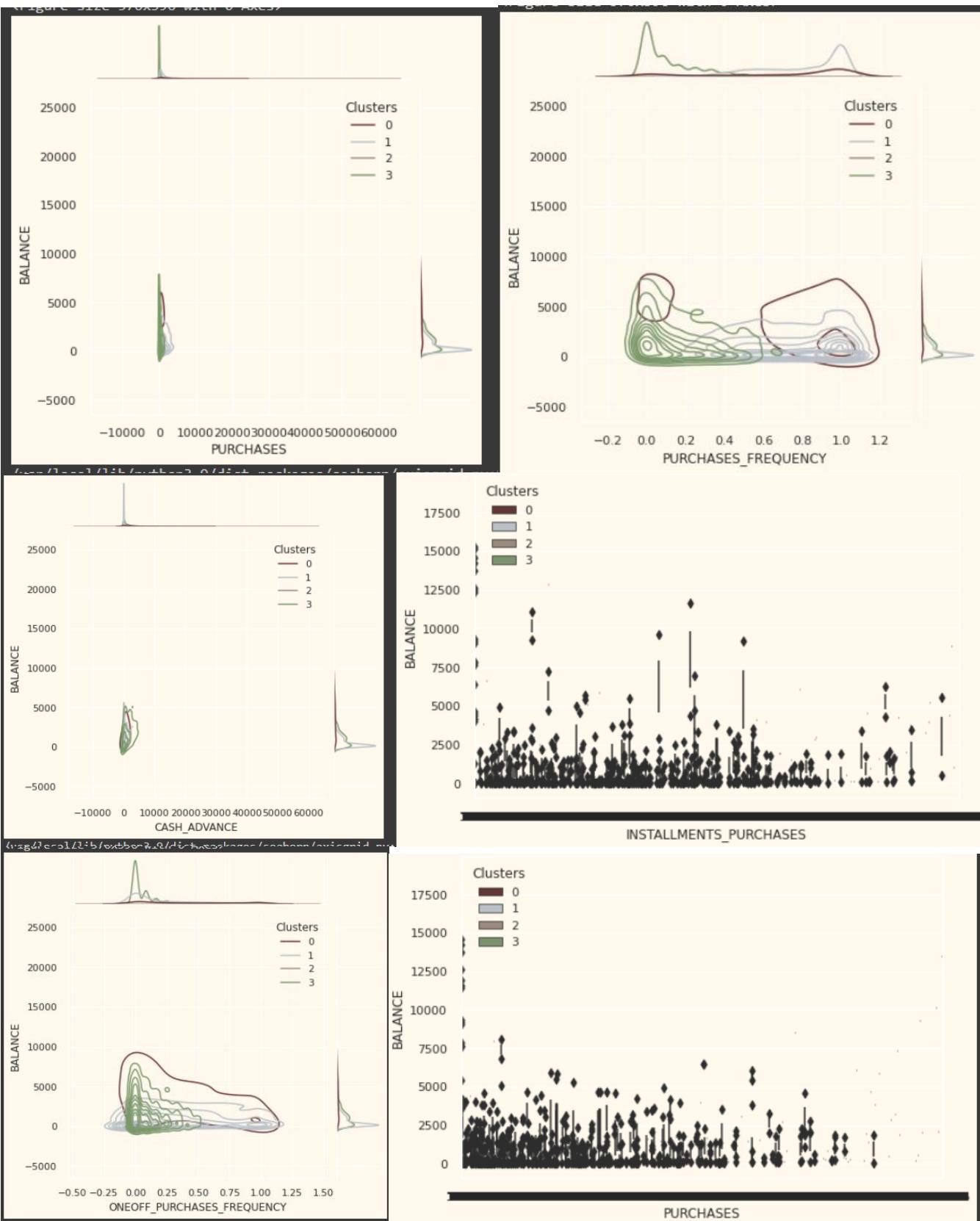
The screenshot shows a Jupyter Notebook interface with the following details:

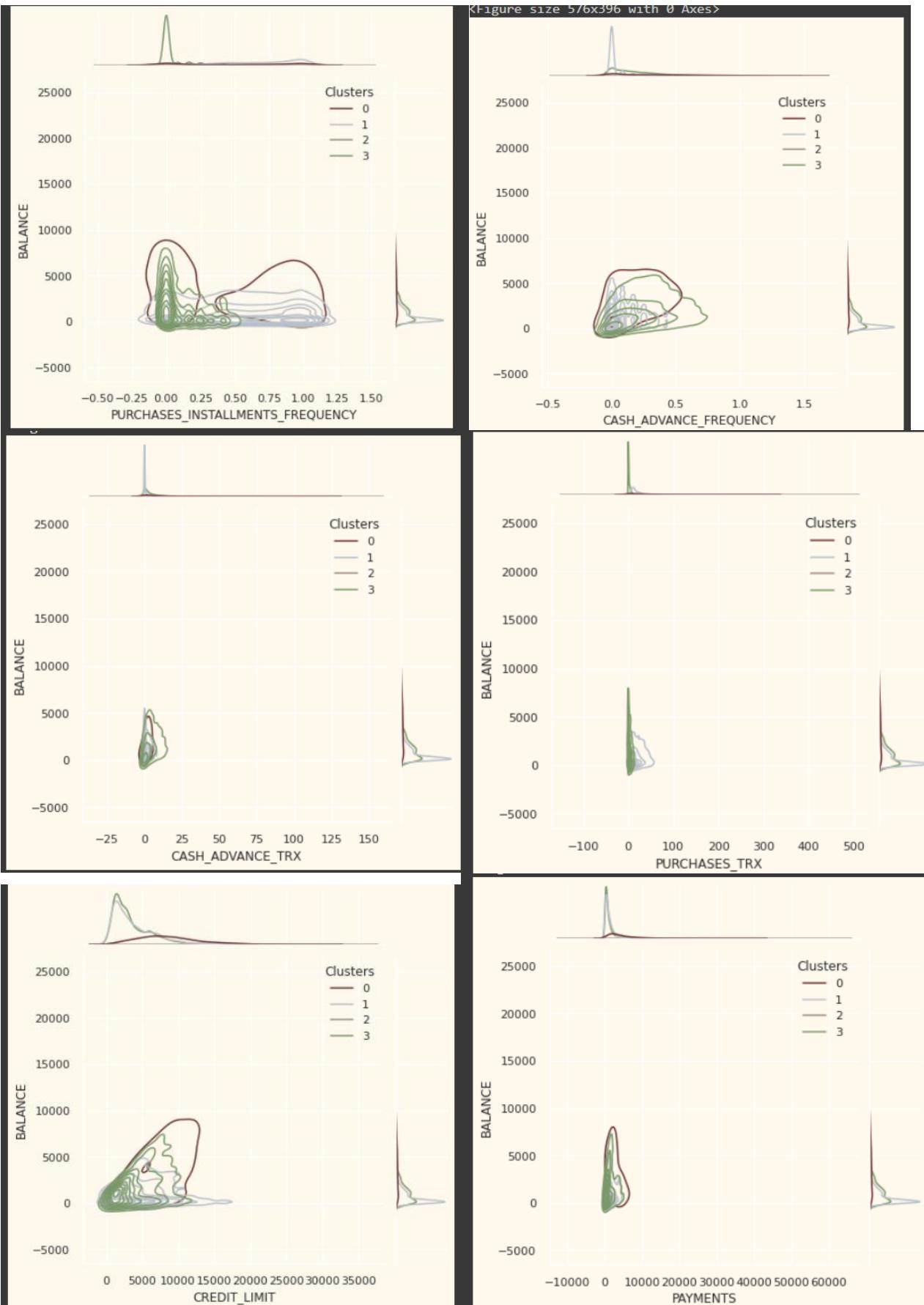
- Title Bar:** Colab research.google.com/drive/1Z3j9eFAZnpxS-NeVdjG_pqZUyFhVsI#scrollTo=z_PY4NMO2ge&uniquifier=2
- Toolbar:** Includes icons for search, refresh, back, forward, and various file operations.
- Header:** Shows the notebook title "Proyecto Clusterizacion.ipynb" and a status message "Se guardaron todos los cambios".
- Code Cell [83]:** Contains Python code to create a scatter plot:

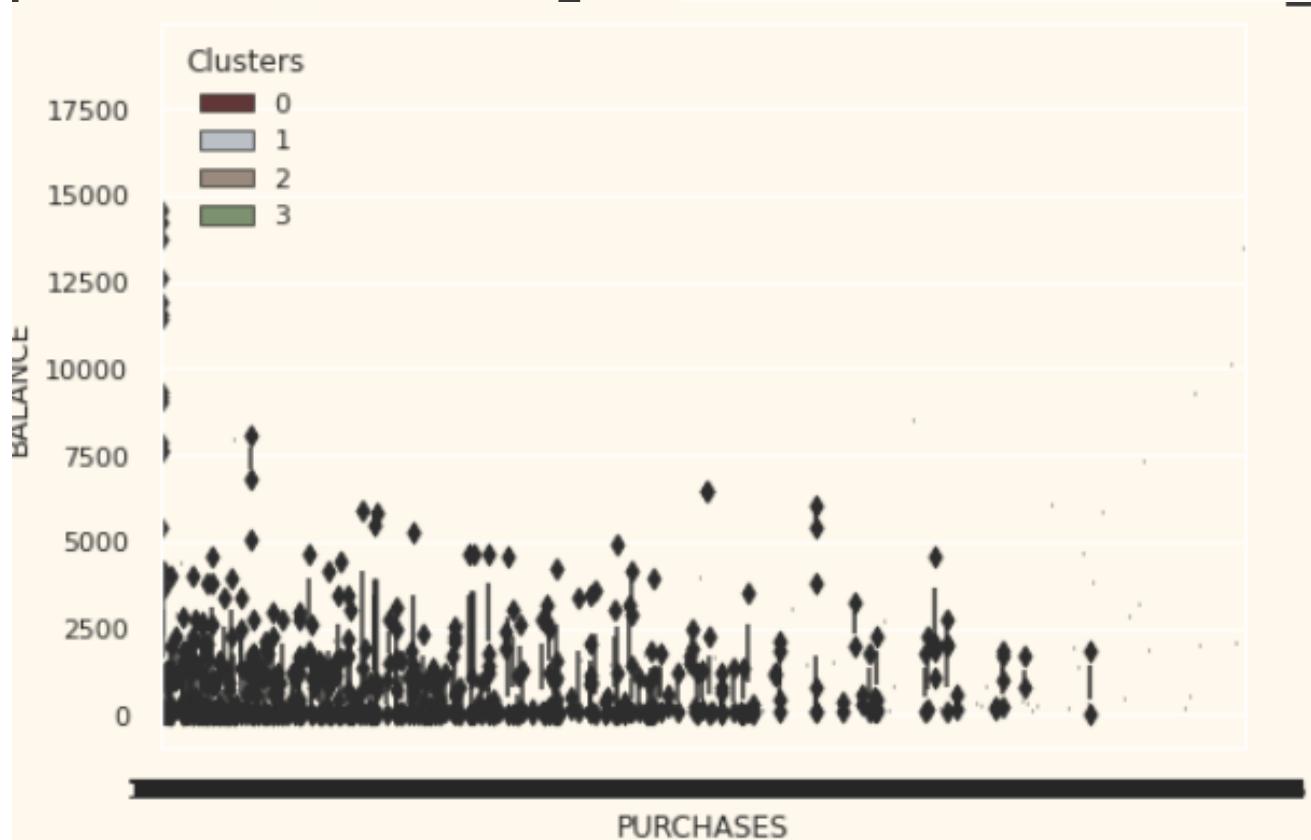
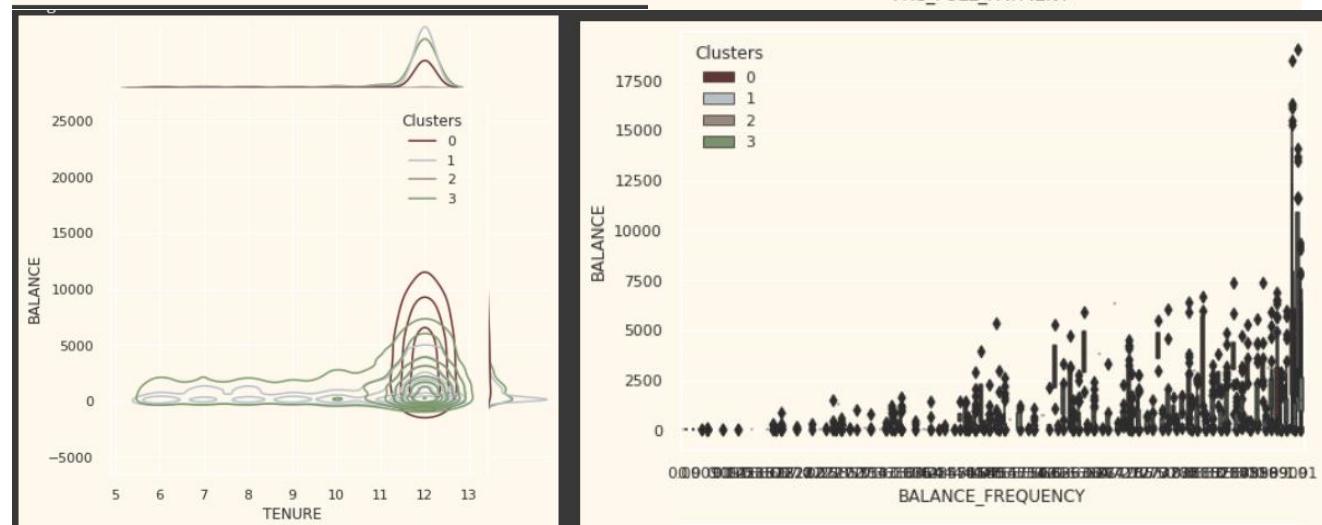
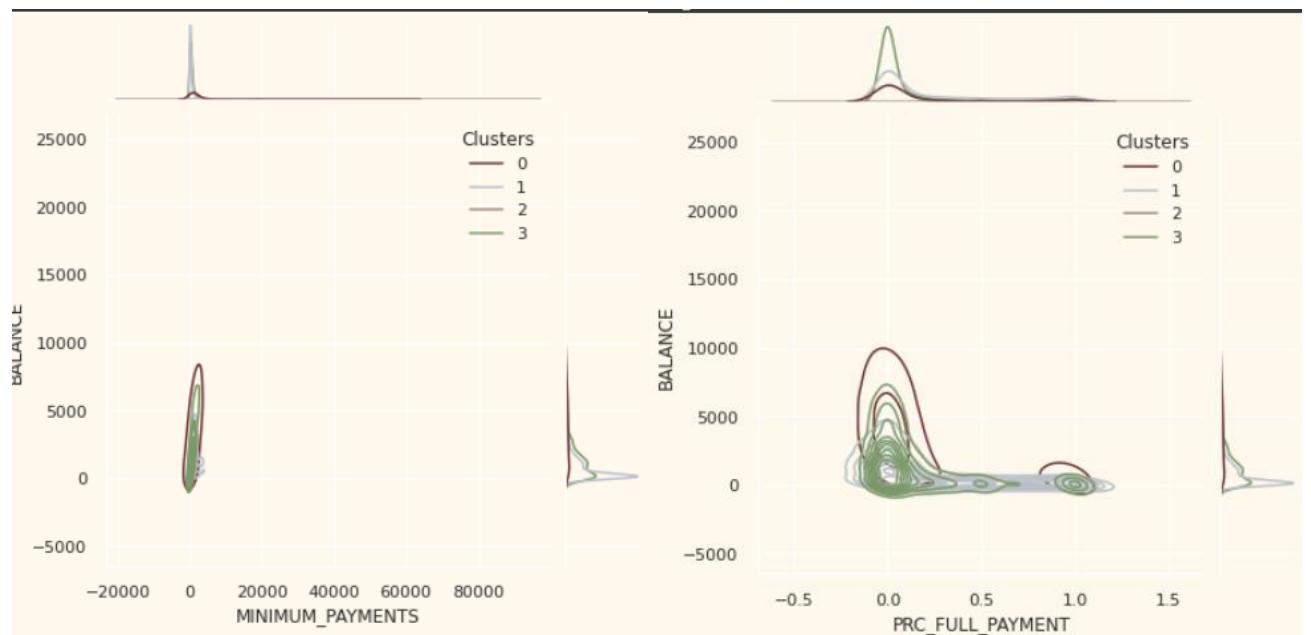
```
1 pl = sns.scatterplot(df, x=df["PAYMENTS"], y=df["MINIMUM_PAYMENTS"], hue=df["Clusters"], palette=palet)
2 pl.set_title("Pagos hechos por el cliente por el minimo del pago relizado con la tarjeta")
3 pl.show()
```
- Output:** A scatter plot titled "Pagos hechos por el cliente por el minimo del pago relizado con la tarjeta". The x-axis is labeled "PAYMENTS" and ranges from 0 to 50,000. The y-axis is labeled "MINIMUM PAYMENTS" and ranges from 0 to 80,000. Data points are colored by cluster (0, 1, 2, 3) according to the legend. The plot shows a clear positive correlation between the two variables.
- Section Header:** "Conclusions"
- Code Cell [85]:** Contains the text "#Mejor modelo seleccionado: Gaussian mixtures".
- Bottom Status Bar:** Shows execution time "15 min, 2 s" and execution date "se ejecutó 12:17 3/29/2023".

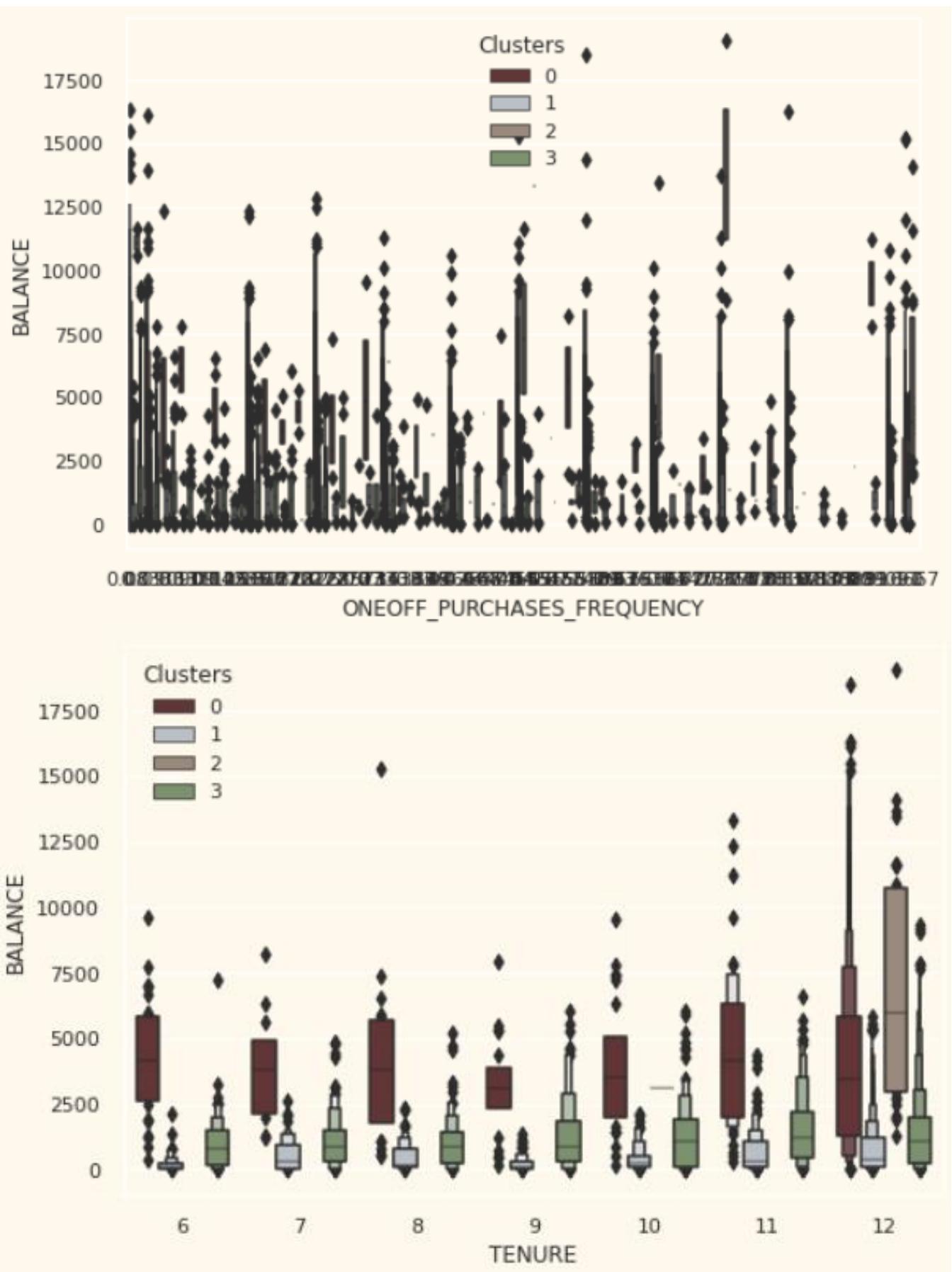
Probe todos los algoritmos mencionados con distintos tipos de hiperparametros y los resultados de arriba son los mejores resultados obtenidos después de todas las pruebas hechas, el mejor algoritmo para mi fue el de Gaussian mixtures, aquí algunos graficos y métricas adicionales hechas con los resultados de Gaussian mixtures.

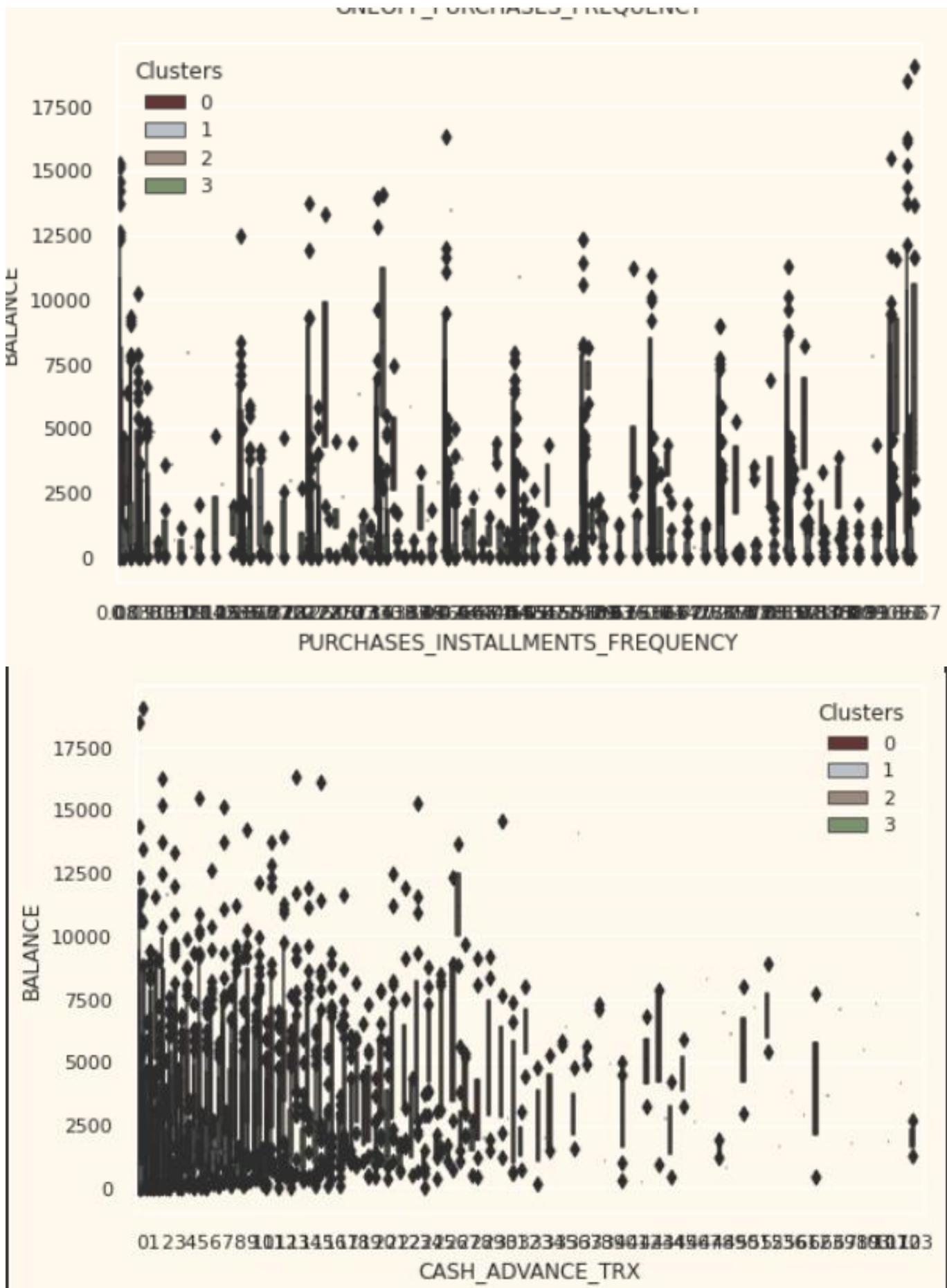


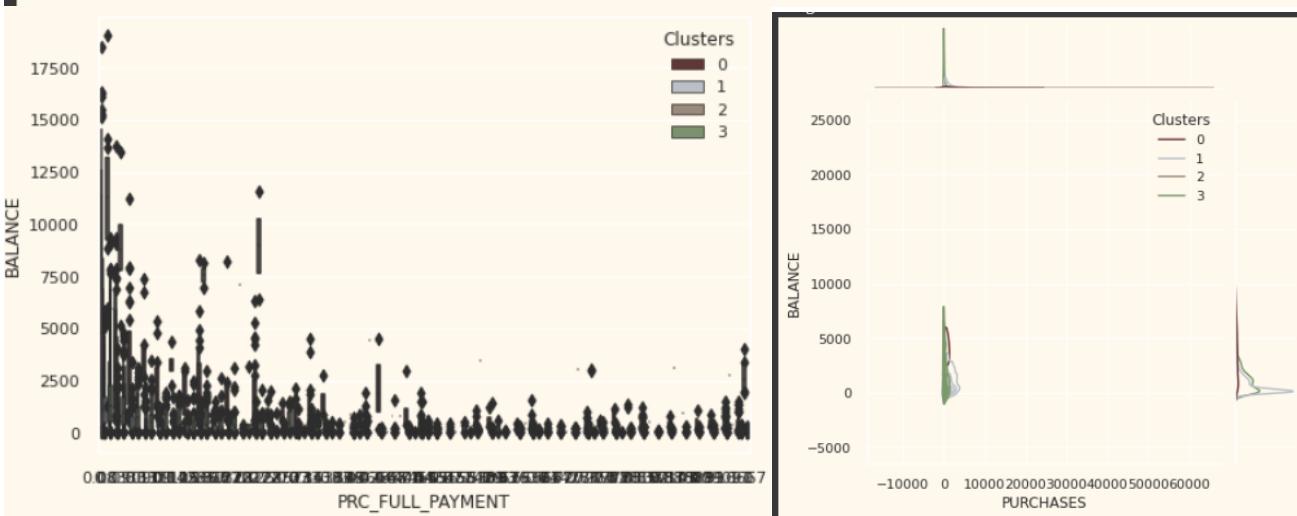
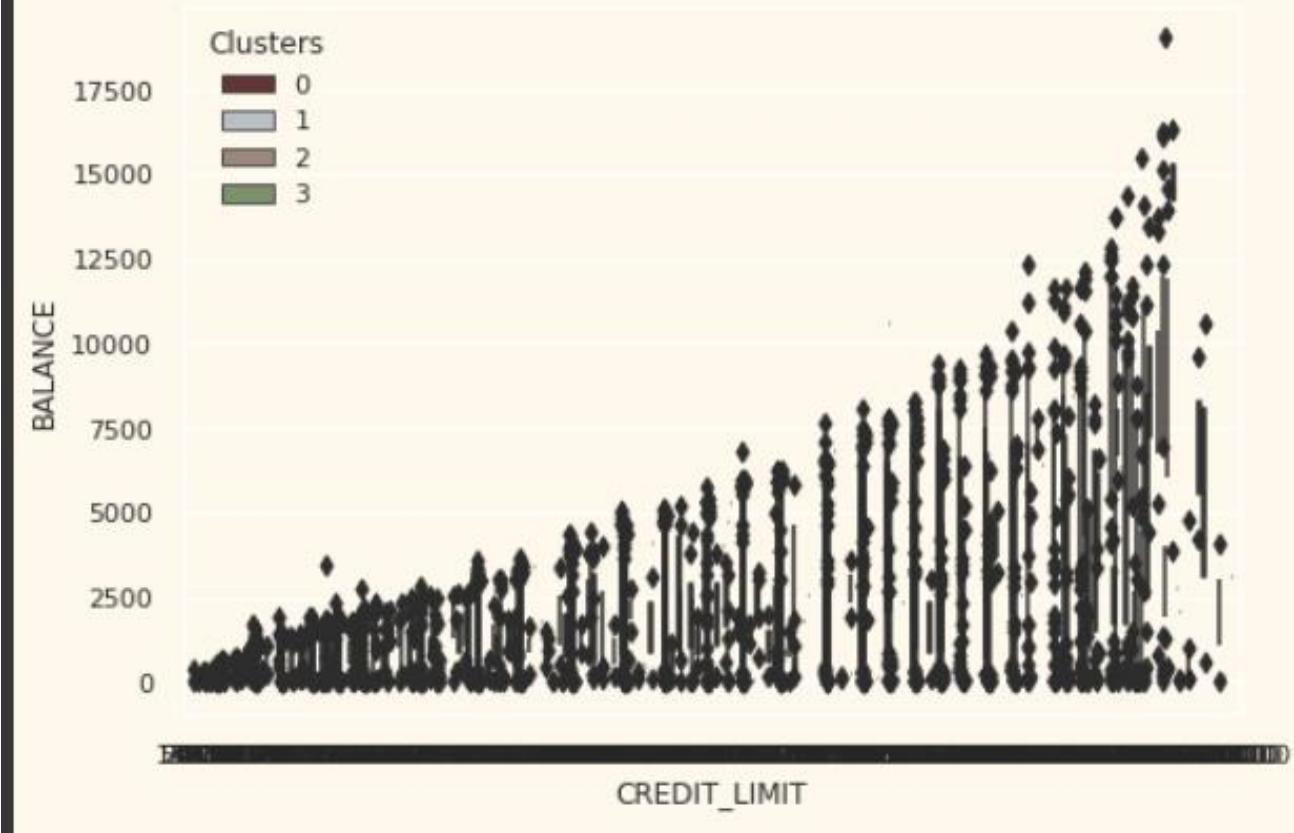
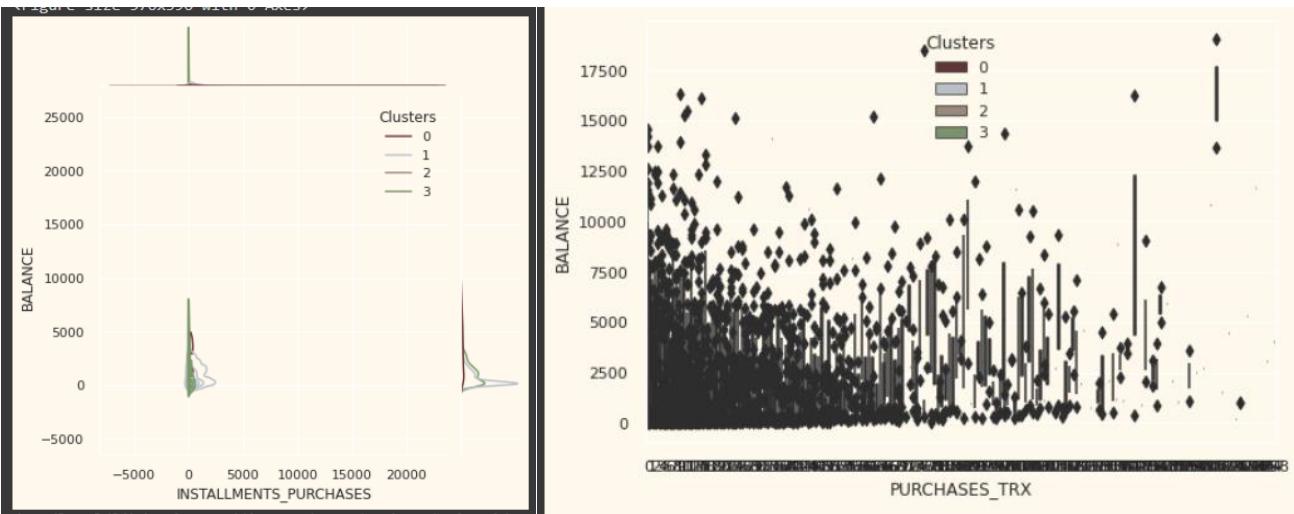












Conclusiones

Como conclusiones llegue a sacar es que en los clusters el primer cluster de color marrón (0) son los usuarios que tienen más balance en la tarjeta y mientras más balance tienen en la tarjeta de crédito más gastos hacen en menos tiempo con diferencia a diferencia de los demás clusters, aparte son los que con más frecuencia compran y por ende tienen una tarjeta de crédito más avanzada. Los demás clusters gris y azul (1, 2) son muy parecidos suelen estar en gastos y sus balance de la tarjeta estando sin estar en el grupo de los que más gastan ni en los que menos gasta Y por otro lado el ultimo cluster a mencionar es el verde que son los que menos balance tienen y los que menos utilizan la tarjeta por ende tienen menor avance en la tarjeta y el menor uso.

Proyecto

https://colab.research.google.com/drive/1Z3J9eFAZnPqwxS-NeVdjG_pqZUyFHvSi?usp=sharing