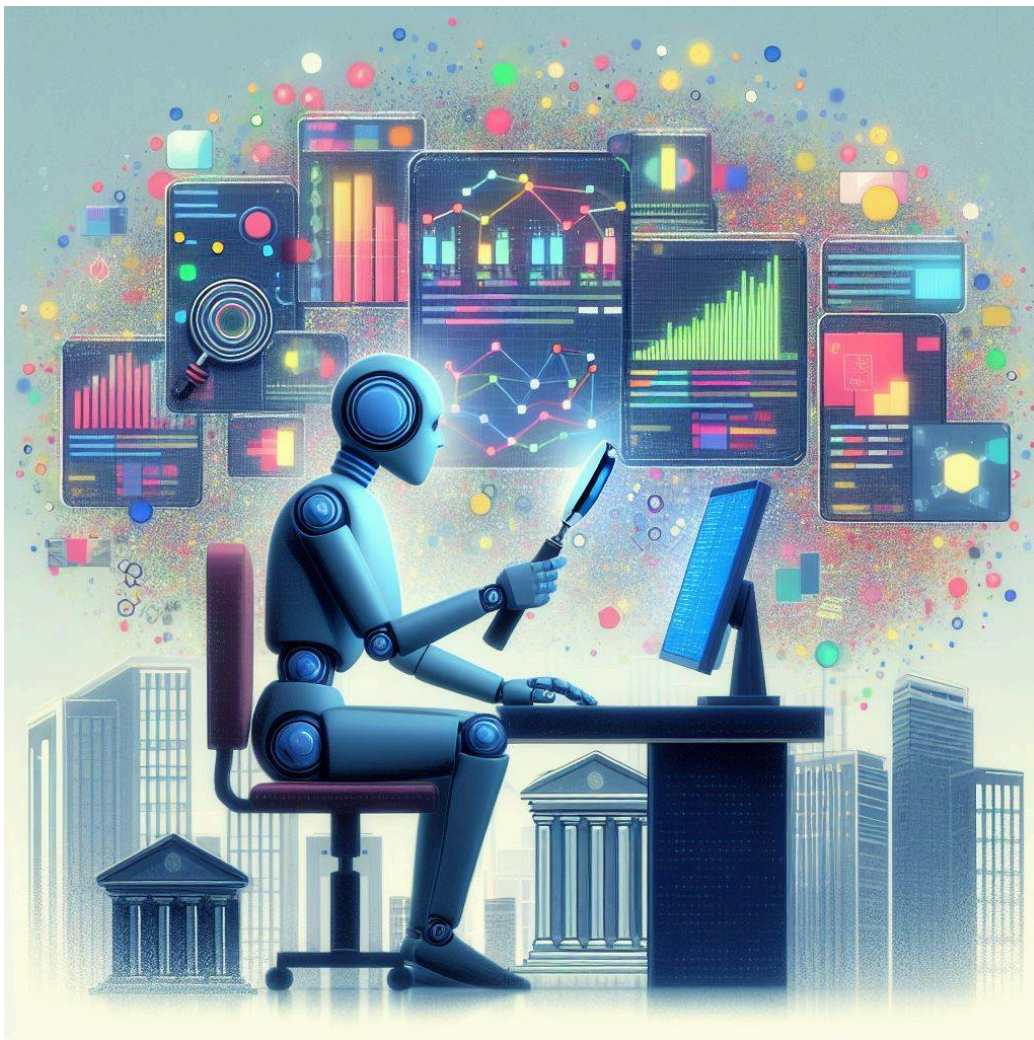


Algoritmos y herramientas para el aprendizaje NO supervisado



Néstor Batista Díaz

ÍNDICE

INTRODUCCIÓN	2
SELECCIÓN DEL DATASET DE ESTUDIO	3
Bank Marketing	3
Statlog (German Credit Data)	6
CONCLUSIÓN	7
DATASET (Statlog (German Credit Data))	8
Importación	8
Overview	8
Tabla de variables	8
Visualización del dataset	9
Análisis (Sweetviz)	9
Tipos de datos	9
Tabla de variables	9
Report de sweetviz	9
Densidad del target	10
Mapa de correlación	10
PREPROCESAMIENTO	11
Convertir valores a numéricos	11
Normalización	11
HOPKINS TEST	11
PCA	12
CLUSTERING	14
K-Means	15
DBSCAN	16
Hierarchical Clustering (Agglomerative)	18
CONCLUSIÓN	19
Dataset con Clusters de K-Means	20
Exportación a Excel	20


INTRODUCCIÓN

La era digital ha transformado radicalmente la forma en que las organizaciones interactúan con sus clientes, haciendo que la recopilación y análisis de datos jueguen un papel crucial en la creación de estrategias de marketing efectivas. En este contexto, la segmentación de clientes surge como una herramienta indispensable, permitiendo a las empresas personalizar y mejorar sus campañas de marketing. La Actividad 4.1, titulada "Segmentación de clientes según datos bancarios", encapsula este concepto al proponer el uso de técnicas de clustering para analizar conjuntos de datos bancarios y, así, identificar segmentos de clientes que pueden ser cruciales para futuras campañas de marketing.

Los conjuntos de datos proporcionados, relacionados con el marketing bancario y el crédito alemán, ofrecen la base perfecta para aplicar estas técnicas. Al centrarnos en estos datos, no solo podremos entender mejor la interacción entre los clientes y los productos bancarios, sino también explorar nuevos enfoques para la segmentación de clientes. Esta introducción sienta las bases para una exploración profunda de cómo la segmentación de clientes basada en datos bancarios puede revolucionar las estrategias de marketing, ofreciendo una perspectiva única sobre la relación entre los datos del cliente y la adquisición de productos/servicios.

SELECCIÓN DEL DATASET DE ESTUDIO


Bank Marketing

**Bank Marketing**
Donated on 2/13/2012

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y).

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Business	Classification
Feature Type	# Instances	# Features
Categorical, Integer	45211	16

Los datos están relacionados con campañas de marketing directo de una entidad bancaria portuguesa. Las campañas de marketing se basaban en llamadas telefónicas. A menudo, era



necesario más de un contacto con el mismo cliente para saber si el producto (depósito bancario a plazo) se suscribía ("sí") o no ("no").

Hay cuatro conjuntos de datos:

1. bank-additional-full.csv con todos los ejemplos (41188) y 20 inputs, ordenados por fecha (de mayo de 2008 a noviembre de 2010), muy próximos a los datos analizados en [Moro et al., 2014].
2. bank-additional.csv con el 10% de los ejemplos (4119), seleccionados aleatoriamente de 1), y 20 inputs.
3. bank-full.csv con todos los ejemplos y 17 inputs, ordenados por fecha (versión antigua de este conjunto de datos con menos inputs).
4. bank.csv con el 10% de los ejemplos y 17 entradas, seleccionadas aleatoriamente de 3 (versión más antigua de este conjunto de datos con menos entradas).


Los conjuntos de datos más pequeños se proporcionan para probar algoritmos de aprendizaje automático más exigentes desde el punto de vista computacional (por ejemplo, SVM).

El objetivo de la clasificación es predecir si el cliente suscribirá (sí/no) un depósito a plazo (variable y).

Si nos fijamos en la tabla de variable podemos encontrar que hay varias columnas con datos nulos:

Description	Units	Missing Values
		no
type of job (categorical: 'admin','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')		no
marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)		no
(categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')		no
has credit in default?		no
average yearly balance	euros	no
has housing loan?		no
has personal loan?		no
contact communication type (categorical: 'cellular','telephone')		yes
last contact day of the week		no
last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')		no
last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.		no
number of contacts performed during this campaign and for this client (numeric, includes last contact)		no
number of days that passed by after the client was last contacted from a previous campaign (numeric; -1 means client was not previously contacted)		yes
number of contacts performed before this campaign and for this client		no
outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')		yes
has the client subscribed a term deposit?		no

Statlog (German Credit Data)

**Statlog (German Credit Data)**
Donated on 11/16/1994

This dataset classifies people described by a set of attributes as good or bad credit risks. Comes in two formats (one all numeric). Also comes with a cost matrix

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Social Science	Classification
Feature Type	# Instances	# Features
Categorical, Integer	1000	20

Se proporcionan dos conjuntos de datos. El conjunto de datos original, en el formato proporcionado por el profesor Hofmann, contiene atributos categóricos/simbólicos y se encuentra en el archivo "german.data".

Para los algoritmos que necesitan atributos numéricos, la Universidad de Strathclyde elaboró el archivo "german.data-numeric". Este fichero se ha modificado y se le han añadido varias variables indicadoras para adaptarlo a los algoritmos que no pueden trabajar con variables categóricas. Varios atributos que son categóricos ordenados (como el atributo 17) se han codificado como enteros. Esta era la forma utilizada por StatLog.

Este conjunto de datos requiere el uso de una matriz de costes (véase más abajo)

```
..... 1      2
-----
1 0      1
-----
2 5      0
```

(1 = Bueno, 2 = Malo)

Las filas representan la clasificación real y las columnas la clasificación prevista.

Es peor clasificar a un cliente como bueno cuando es malo (5), que clasificar a un cliente como malo cuando es bueno (1).

En este caso no observamos nulos en la tabla de variables:

Description	Units	Missing Values
Status of existing checking account		no
Duration	months	no
Credit history		no
Purpose		no
Credit amount		no
Savings account/bonds		no
Present employment since		no
Installment rate in percentage of disposable income		no
Personal status and sex		no
Other debtors / guarantors		no
Present residence since		no
Property		no
Age	years	no
Other installment plans		no
Housing		no
Number of existing credits at this bank		no
Job		no
Number of people being liable to provide maintenance for		no
Telephone		no
foreign worker		no

CONCLUSIÓN

Para esta actividad elegiré el dataset de “**Statlog (German Credit Data)**”, aunque tenga solo 1000 registros no tiene datos nulos y espero que eso me facilite la preparación del dataset para el entrenamiento.

DATASET (Statlog (German Credit Data))

Importación

```
# fetch dataset
statlog_german_credit_data = fetch_ucirepo(id=144)

# data (as pandas dataframes)
df = statlog_german_credit_data.data.original
```

Overview

Tabla de variables

```
# variable information
df_variables = statlog_german_credit_data.variables
df_variables.style.hide(axis="index")
```

name	role	type	demographic		description	units	missing_values
Attribute1	Feature	Categorical	None		Status of existing checking account	None	no
Attribute2	Feature	Integer	None		Duration	months	no
Attribute3	Feature	Categorical	None		Credit history	None	no
Attribute4	Feature	Categorical	None		Purpose	None	no
Attribute5	Feature	Integer	None		Credit amount	None	no
Attribute6	Feature	Categorical	None		Savings account/bonds	None	no
Attribute7	Feature	Categorical	Other		Present employment since	None	no
Attribute8	Feature	Integer	None		Installment rate in percentage of disposable income	None	no
Attribute9	Feature	Categorical	Marital Status		Personal status and sex	None	no
Attribute10	Feature	Categorical	None		Other debtors / guarantors	None	no
Attribute11	Feature	Integer	None		Present residence since	None	no
Attribute12	Feature	Categorical	None		Property	None	no
Attribute13	Feature	Integer	Age		Age	years	no
Attribute14	Feature	Categorical	None		Other installment plans	None	no
Attribute15	Feature	Categorical	Other		Housing	None	no
Attribute16	Feature	Integer	None		Number of existing credits at this bank	None	no
Attribute17	Feature	Categorical	Occupation		Job	None	no
Attribute18	Feature	Integer	None		Number of people being liable to provide maintenance for	None	no
Attribute19	Feature	Binary	None		Telephone	None	no
Attribute20	Feature	Binary	Other		foreign worker	None	no
class	Target	Binary	None		1 = Good, 2 = Bad	None	no

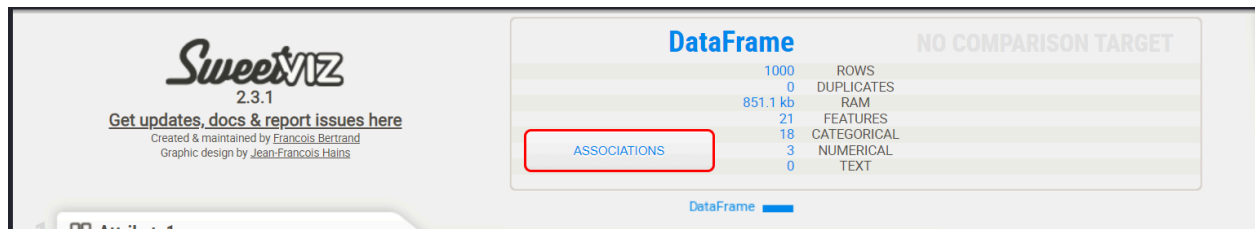
Visualización del dataset

```
df.head().style.hide(axis="index")
```

Attribute1	Attribute2	Attribute3	Attribute4	Attribute5	Attribute6	Attribute7	Attribute8	Attribute9	Attribute10	Attribute11	Attribute12	Attribute13	Attribute14	Attribute15
A11	6	A34	A43	1169	A65	A75	4	A93	A101	4	A121	67	A143	A152
A12	48	A32	A43	5951	A61	A73	2	A92	A101	2	A121	22	A143	A152
A14	12	A34	A46	2096	A61	A74	2	A93	A101	3	A121	49	A143	A152
A11	42	A32	A42	7882	A61	A74	2	A93	A103	4	A122	45	A143	A153
A11	24	A33	A40	4870	A61	A73	3	A93	A101	4	A124	53	A143	A153

Análisis (Sweetviz)

Tipos de datos



Aquí podemos observar que efectivamente hay 1000 filas y 21 características que son las columnas. De estas 21 características 18 son Categóricas y 3 numéricas, esto discrepa con los datos de la tabla de variables vista anteriormente, se debe a que hay características que son numéricas que solo tienen 4 datos únicos, por lo cual entiende que se refieren a una categoría y no a un dato numérico. Por ejemplo, lo podemos ver en el Attribute8:

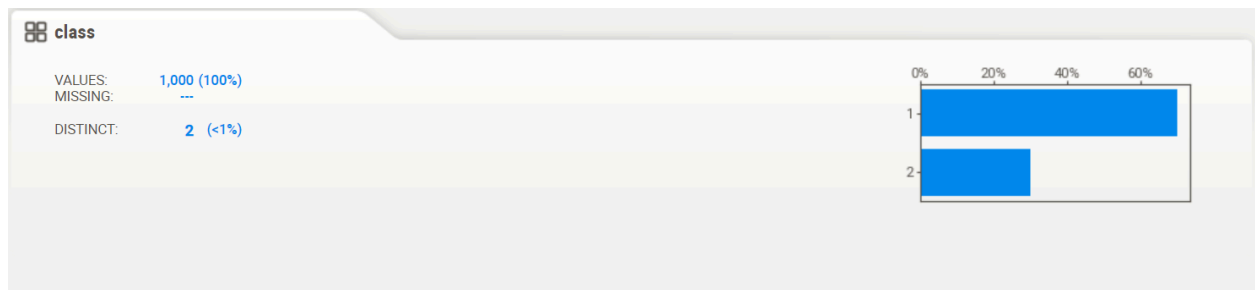
Tabla de variables

Attribute8	Feature	Integer	None	Installment rate in percentage of disposable income	None	no
------------	---------	---------	------	---	------	----

Report de sweetviz

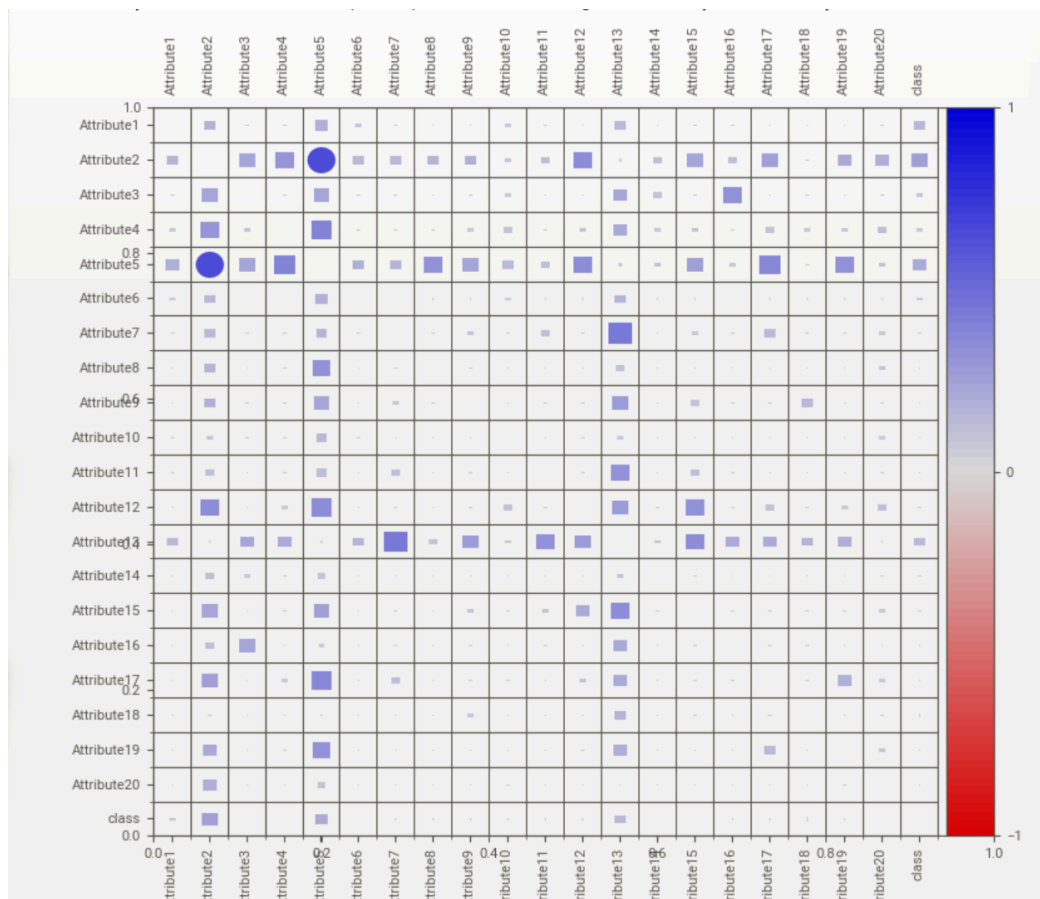


Densidad del target



Podemos observar que el dataset está desbalanceado.

Mapa de correlación



Aquí podemos observar que las categorías Attribute5 y Attribute2 están muy relacionadas entre sí, también podemos ver lo mismo con Attribute13 y Attribute7.

PREPROCESAMIENTO

Convertir valores a numéricos

```
# Conversión de variables catgóricas a numéricas
le = LabelEncoder()

pd.options.mode.copy_on_write = True # Para que no muestre el warning

categorical_columns = df_variables.loc[df_variables['type'] == 'Categorical', 'name'].tolist()

# Añadimos las columnas Attribute19 y Attribute20, porque pese a tener el type = binary
# en la tabla de variables, contienen variables str
additional_columns = ['Attribute19', 'Attribute20']

categorical_columns.extend(additional_columns)

print(cldr.start+"Columnas convertidas:", cldr.color+'{}'.format(categorical_columns))

for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
```

✓ 0.0s

Columnas convertidas: ['Attribute1', 'Attribute3', 'Attribute4', 'Attribute6', 'Attribute7', 'Attribute9', 'Attribute10', 'Attribute12', 'Attribute14', 'Attribute15',

Normalización

```
column_names = df.columns.values

scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)
X = pd.DataFrame(scaled_data, columns=column_names)
```

```
X.head().style.hide(axis="index")
```

Attribute1	Attribute2	Attribute3	Attribute4	Attribute5	Attribute6	Attribute7	Attribute8	Attribute9
-1.254566	-1.236478	1.344014	0.264068	-0.745131	1.833169	1.338078	0.918477	0.449326
-0.459026	2.248194	-0.503428	0.264068	0.949817	-0.699707	-0.317959	-0.870183	-0.963650
1.132053	-0.738668	1.344014	1.359785	-0.416562	-0.699707	0.510060	-0.870183	0.449326
-1.254566	1.750384	-0.503428	-0.101171	1.634247	-0.699707	0.510060	-0.870183	0.449326
-1.254566	0.256953	0.420293	-1.196889	0.566664	-0.699707	-0.317959	0.024147	0.449326

HOPKINS TEST

Vamos a determinar mediante este test si tiene sentido aplicar clustering en este dataset.

El test de Hopkins es una prueba estadística utilizada para medir la tendencia al agrupamiento de datos en un conjunto de datos preprocesados. En este contexto, la prueba de Hopkins ayuda a determinar en qué medida existen clusters significativos en los datos que se van a agrupar.

La hipótesis nula (H_0) en el test de Hopkins establece que el conjunto de datos no está distribuido de manera uniforme y que contiene clusters significativos. Por otro lado, la hipótesis alternativa (H_1) indica que el conjunto de datos está distribuido de manera uniforme y que no contiene clusters significativos.

En términos simples, al realizar el test de Hopkins, si el valor obtenido está en el rango de $\{0.7, \dots, 0.99\}$, se acepta la hipótesis nula (H_0), lo que significa que el conjunto de datos tiene una alta tendencia a formar clusters significativos. Por lo tanto, se concluye que existen clusters en los datos que se van a agrupar.

En nuestro caso el resultado es el siguiente:

```
ols: Hopkins Test :.  
*****  
Resultado: 0.6742  
>> Según el resultado anterior, No contiene agrupaciones significativas  
  
*****  
... Conclusiones: Rechazamos  $H_0$  ...  
*****
```

Esto indica que no hay agrupaciones significativas y por lo tanto no se van a poder identificar bien los clusters. Aun así proseguiremos para completar la tarea.

PCA

El análisis de componentes principales (PCA) es un método utilizado en el aprendizaje automático no supervisado (como el clustering) que reduce los datos de alta dimensión a dimensiones más pequeñas preservando tanta información como sea posible. Al utilizar el PCA antes de aplicar el algoritmo de clustering, permite reducir las dimensiones, el ruido de los datos y disminuir el coste computacional. En este cuaderno, el número de características se reducirá a 2 dimensiones para poder visualizar los resultados del clustering.

```

columns = X.columns

# Reducimos la dimensionalidad de los datos (a dos dimensiones)
pca = PCA(n_components = 2)
X = pca.fit_transform(X)
# Mostramos el porcentaje de varianza explicada por cada uno de los componentes seleccionados.
print(clr.color+'{}'.format(pca.explained_variance_ratio_))

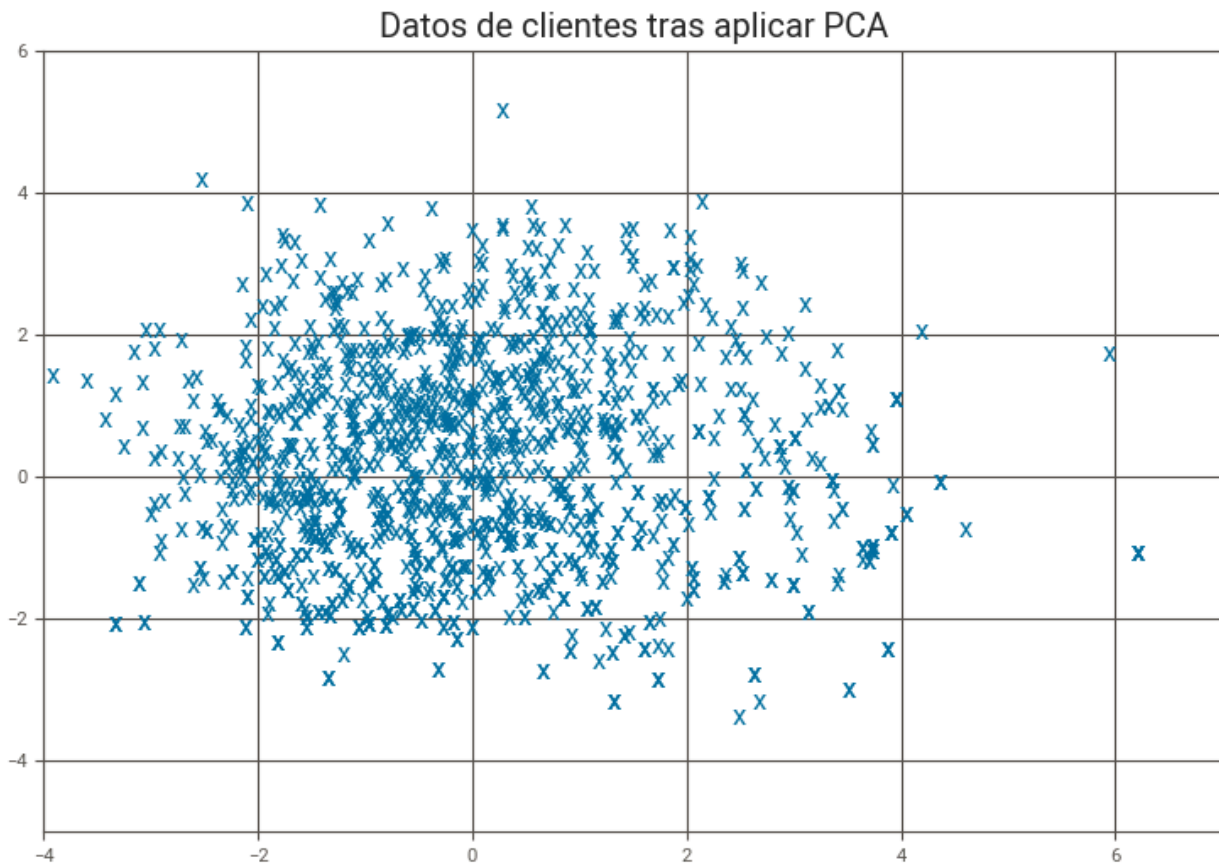
# Visualizar la "importancia" de cada variable original del problema en las nuevas dimensiones
pd.DataFrame(np.matrix.transpose(pca.components_), columns=['PC-1', 'PC-2'], index= columns)

```

✓ 0.0s

[0.12111642 0.1019251]

	PC-1	PC-2
Attribute1	0.005823	0.328902
Attribute2	0.410367	-0.216894
Attribute3	0.014731	0.415422
Attribute4	0.014197	-0.038463
Attribute5	0.421876	-0.199063
Attribute6	0.086542	0.205854
Attribute7	0.167439	0.323872
Attribute8	0.047567	0.100224
Attribute9	0.024394	0.143962
Attribute10	-0.123337	-0.065111
Attribute11	0.159025	0.192135
Attribute12	0.412791	-0.070934
Attribute13	0.181322	0.334888

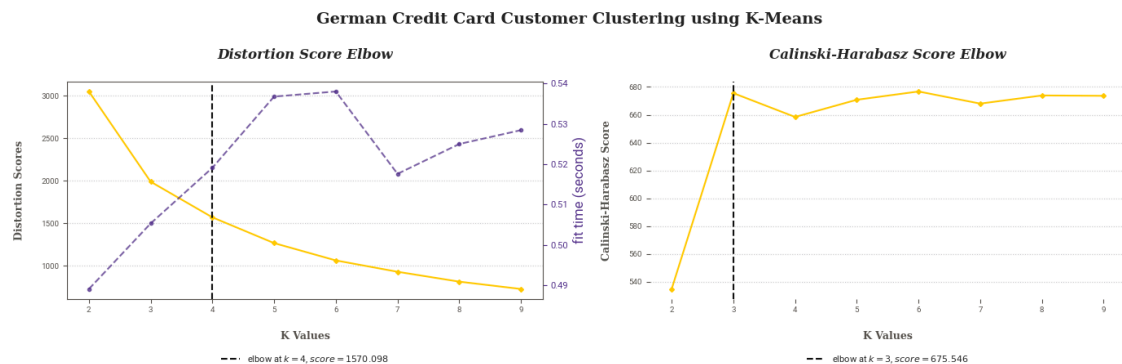


En este caso no se logra apreciar separación de clusters a primera vista.

CLUSTERING

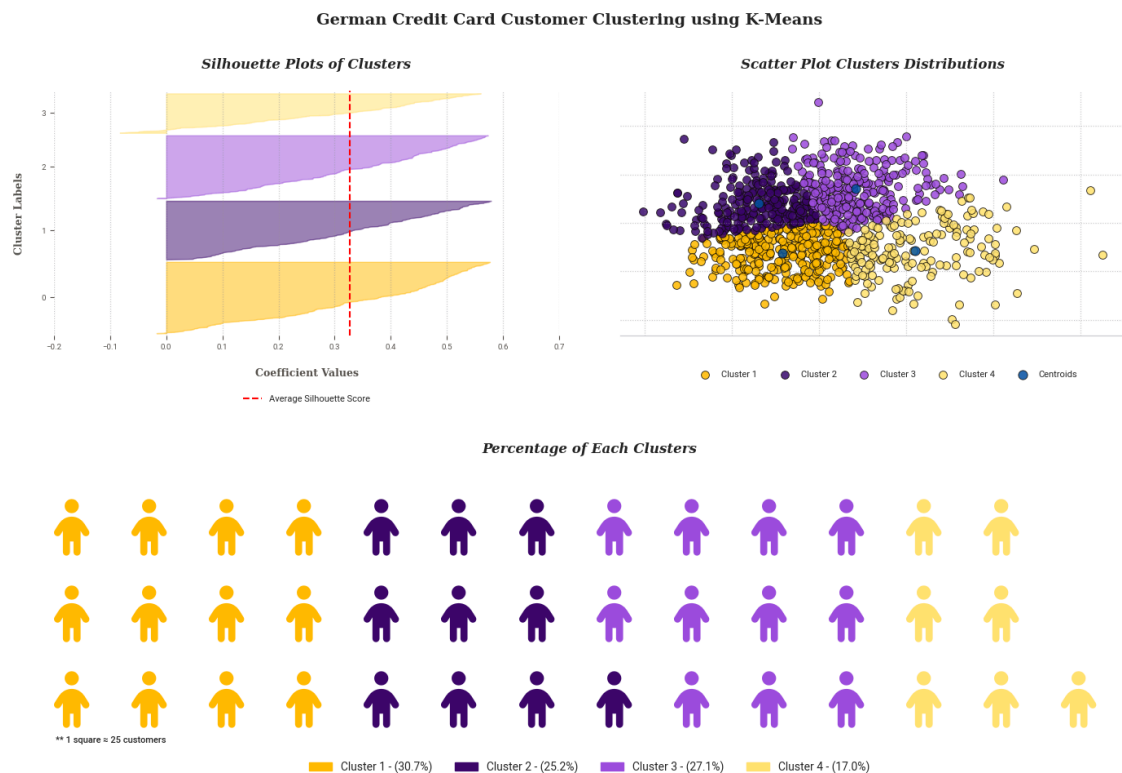
Para esta actividad he decidido trabajar con tres modelos (K-Means, DBSCAN, Hierarchical Clustering) para compararlos.

K-Means



Néstor Batista Díaz

Como podemos observar, hay una discrepancia entre las dos gráficas. En este caso voy a elegir $k=4$, ya que es la mejor en la gráfica de distorsión y tiene un buen pico en la gráfica de Calinski-harabasz.



Néstor Batista Díaz

Podemos ver que en la gráfica de dispersión los clusters están bastante dispersos, en especial el 2 y el 3, eso significa que K-Means interpreta que hay outliers en estos clusters.

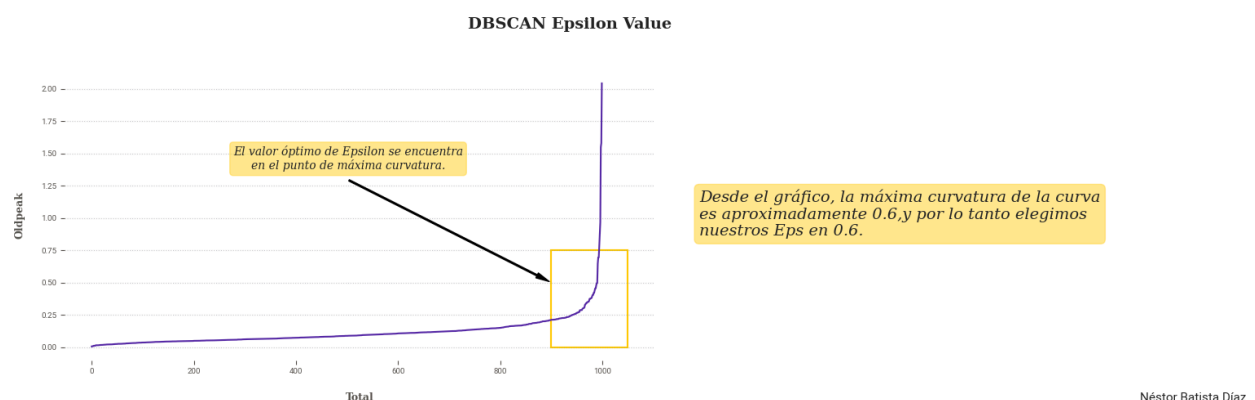
El siguiente paso consiste en evaluar la calidad de la agrupación proporcionada por K-Means. Para evaluar la calidad se utilizarán el índice de Davies-Bouldin, la puntuación de Silhouette y el índice de Calinski-Harabasz.

```
.: Evaluate Clustering Quality :.  
*****  
.: Davies-Bouldin Index: 0.943  
.: Silhouette Score: 0.327  
.: Calinski Harabasz Index: 658.413
```

Basándonos en la puntuación de la evaluación anterior, la calidad de la agrupación utilizando K-Means con 4 clusters es decente. Esto se debe al solapamiento entre clústeres y la gran dispersión que hay en cada uno de ellos, como se muestra en el gráfico de dispersión de la sección anterior.

DBSCAN

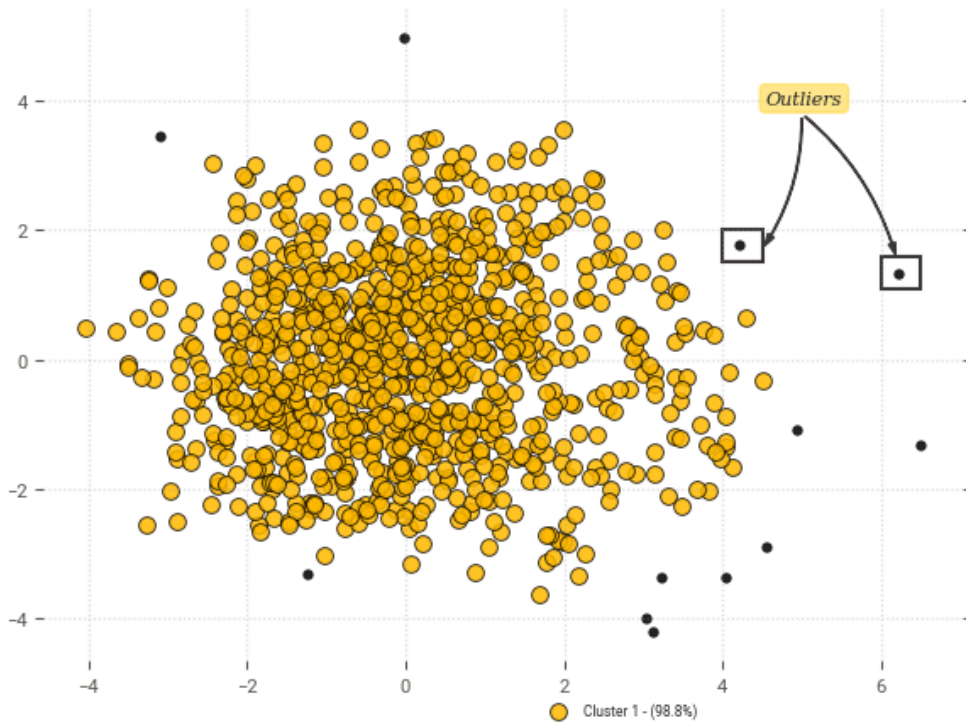
Antes de aplicar el algoritmo DBSCAN, tenemos que definir los parámetros DBSCAN mencionados anteriormente. Para MinPoint, dado que PCA ya se realiza en dos dimensiones, utilizaremos el valor por defecto (4) como valores MinPoint. Para los valores Epsilon, usando los Vecinos más cercanos, determinaremos la separación entre cada punto de datos y su vecino más cercano, y luego los ordenaremos antes de graficarlos. Después, podemos determinar el mayor valor en la curva del gráfico a partir del trazado.



Basándonos en los resultados de la curvatura máxima anteriores y en los valores de MinPoint anteriores, el siguiente paso es implementar DBSCAN y evaluar los resultados.

German Credit Card Customer Clustering using DBSCAN

Se formaron dos grupos de clientes de tarjetas de crédito. También se detectaron algunos valores atípicos.



Néstor Batista Díaz

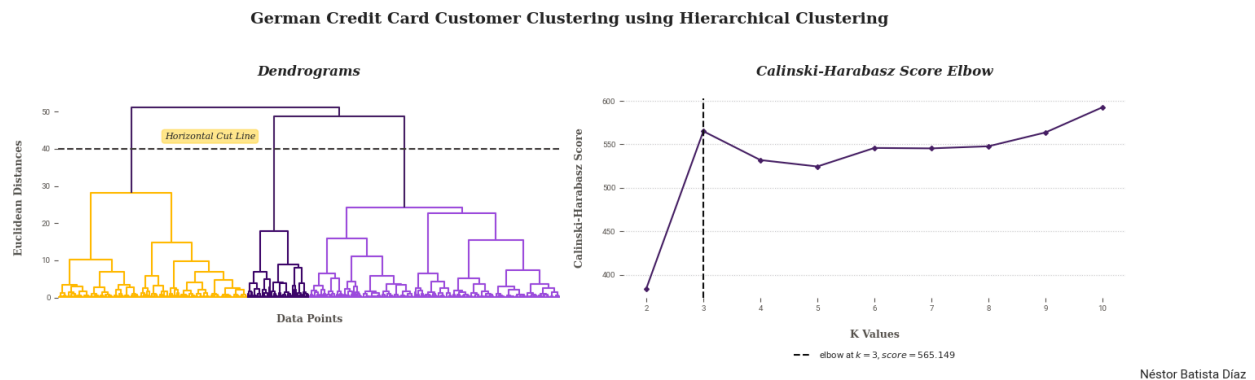
A partir de la implementación de DBSCAN, solo se forma un cluster. También, se han detectado algunos valores atípicos, ya que algunos puntos están demasiado alejados de otros puntos de datos (DBSCAN los considera valores atípicos y les asigna la etiqueta -1). El siguiente paso consiste en evaluar la calidad de agrupación que proporciona DBSCAN.

```
.: Evaluate Clustering Quality :.  
*****  
.: Davies-Bouldin Index: 1.792  
.: Silhouette Score: 0.486  
.: Calinski Harabasz Index: 25.881
```

La calidad de la agrupación mediante DBSCAN con dos conglomerados y valores atípicos es muy bajo según la puntuación de la evaluación anterior. El modelo K-means es superior en todos los parámetros medidos.

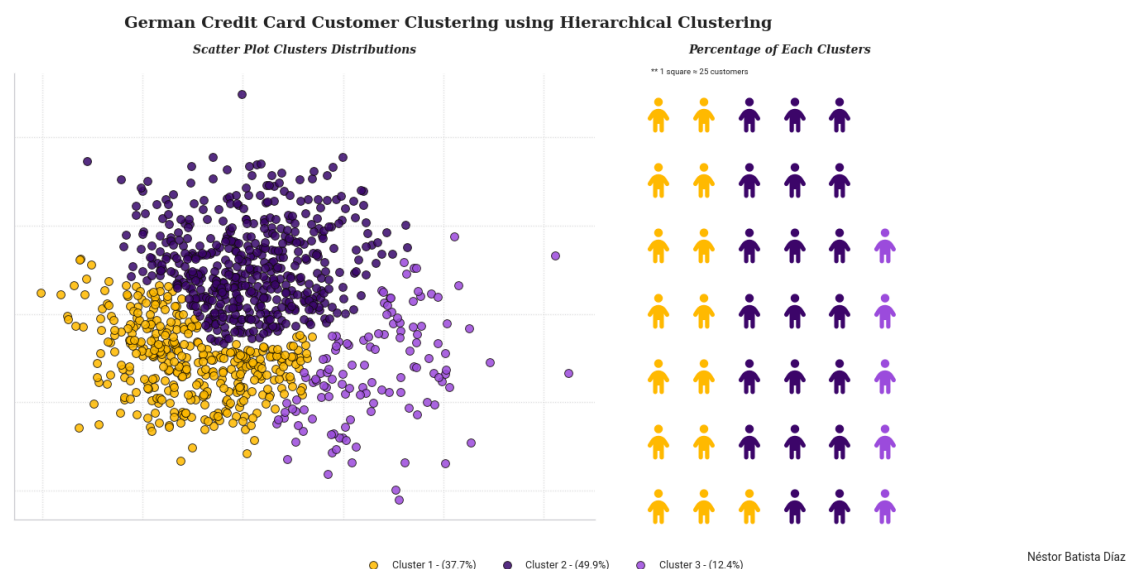
Hierarchical Clustering (Agglomerative)

Como primer paso, tenemos que hacer primero un dendrograma y luego trazar una línea horizontal entre ambos puntos. A continuación, evaluaremos el dendrograma creado y compararemos sus resultados con la puntuación de Calinski-Harabasz.



Basándose en la distancia euclídea del dendrograma anterior, puede concluirse que el número de conglomerados será tres, ya que la línea vertical más alta/distancia más grande se encuentra en la primera línea/rama (a la izquierda de la imagen) y el umbral corta el dendrograma en tres partes. Además, basándonos en la puntuación Calinski-Harabasz, el cluster óptimo obtenido es 3.

A continuación, implementaremos este número en el algoritmo de agrupación aglomerativa y visualizaremos y evaluaremos los clusters creados.



De la aplicación del Hierarchical Clustering se desprende que se han formado 3 clusters. De los 3 clusters, el cluster 2 tiene la mayoría de los puntos de datos, seguido del cluster 1. También podemos observar que el cluster 1 y 3 están superpuestos.

El último paso consiste en evaluar la calidad de agrupación que ofrece la hierarchical clustering. Para evaluar la calidad se utilizarán la puntuación de Silhouette y el índice de Davies-Bouldin.

```
# --- Evaluate DBSCAN Cluster Quality ---
db_agg, ss_agg, ch_agg = evaluate_clustering(X, y_agg_cluster)
✓ 0.0s

.: Evaluate Clustering Quality :.
*****
.: Davies-Bouldin Index: 0.987
.: Silhouette Score: 0.335
.: Calinski Harabasz Index: 565.149
```

A partir de los resultados de la evaluación de la calidad de la agrupación utilizando la hierarchical clustering, se observa que los resultados obtenidos son ligeramente diferentes de los de K-Means. Al utilizar la hierarchical clustering, la puntuación de Silhouette obtenida se aproxima a 0, lo que indica que los conglomerados se solapan. Además, un índice Davies-Bouldin alto indica una calidad de agrupación decente. En comparación con K-Means, las puntuaciones de Silhouette y Davies-Bouldin de hierarchical clustering son más altas. El índice Calinski-Harabasz obtenido es ligeramente inferior en comparación con K-Means, pero superior en comparación con DBSCAN.

CONCLUSIÓN

```
.: Model Accuracy Comparison :.
*****
```

Model	Davies-Bouldin Index	Silhouette Score	Calinski-Harabasz Index
K-Means	0.875000	0.358000	1000.974000
Hierarchical Clustering	0.900000	0.325000	890.433000
DBSCAN	0.717000	0.292000	29.740000

± Código

El Índice Davies-Bouldin revela que K-Means forma clusters más compactos y mejor separados en comparación con DBSCAN y ligeramente superior al Hierarchical Clustering. Además, el

Score de Silhouette de K-Means es el mejor de los tres, indicando una buena cohesión y separación de los clusters. Finalmente, el Índice Calinski-Harabasz muestra que K-Means supera con creces a los otros modelos en formar clusters densos e internamente cohesivos. Por lo tanto, K-Means es la opción más equilibrada y efectiva para el clustering de este dataset específico.

Dataset con Clusters de K-Means

Column Name	Metrics	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Overall
Attribute1	mean	0.931596	2.027778	2.158672	1.147059	1.577000
Attribute2	mean	19.052117	12.583333	21.059041	36.329412	20.903000
Attribute3	mean	1.964169	2.869048	3.166052	2.123529	2.545000
Attribute4	mean	3.270358	3.238095	3.247232	3.394118	3.277000
Attribute5	mean	2577.671010	1735.321429	3040.166052	7168.982353	3271.258000
Attribute6	mean	0.641694	1.035714	1.763838	0.994118	1.105000
Attribute7	mean	1.791531	2.369048	3.121771	2.300000	2.384000
Attribute8	mean	2.859935	2.849206	3.295203	2.847059	2.973000
Attribute9	mean	1.514658	1.773810	1.808118	1.647059	1.682000
Attribute10	mean	0.208469	0.182540	0.066421	0.100000	0.145000
Attribute11	mean	2.446254	2.726190	3.346863	2.941176	2.845000
Attribute12	mean	1.162866	0.619048	1.715867	2.235294	1.358000
Attribute13	mean	28.853420	36.134921	41.830258	36.741176	35.546000
Attribute14	mean	1.635179	1.769841	1.741697	1.500000	1.675000
Attribute15	mean	0.713355	0.825397	1.081181	1.229412	0.929000

Exportación a Excel

```
df.to_excel('results.xlsx', index=False)
```

✓ 0.6s



REFERENCIAS

- ❖ [Dataset seleccionado](#)
- ❖ [Clustering for Effective Marketing Strategy](#)
- ❖ [GitHub](#)