PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

TEMA:

SEGMENTACIÓN DE DONANTES POTENCIALES DE UNA FUNDACIÓN MEDIANTE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO.

AUTOR:

CARLOZAMA VILLOTA JUAN CARLOS

DIRECTOR:

MELGAREJO HEREDIA RAFAEL

TRABAJO DE TITULACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE INFORMACIÓN CON MENCIÓN EN DATA SCIENCE

QUITO, 2024

DEDICATORIA

Con profundo agradecimiento, dedico este trabajo a Dios, a mi familia, mi fuente de amor y fuerza, y a todos aquellos que creyeron en mí. Gracias por ser mi faro en este camino.

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a mi tutor de tesis por su invaluable guía, paciencia y conocimientos durante la realización de este trabajo. Su orientación fue fundamental para el desarrollo de esta investigación.

RESUMEN

La presente tesis tiene como objetivo desarrollar un modelo de segmentación de donantes potenciales para una fundación, utilizando algoritmos de aprendizaje automático.

Este proyecto busca identificar variables clave para la segmentación basada en comportamientos y patrones de donación, recopilar datos relevantes sobre donantes, y evaluar la precisión y eficiencia del algoritmo implementado. La metodología utilizada se basa en el enfoque CRISP-DM y se aplicarán técnicas de aprendizaje no supervisado.

ÍNDICE

1. Introducción 1

1.1. Planteamiento del Problema 1

1.2. Justificación 1

1.3. Objetivos 1

1.3.1. Objetivo General 1

1.3.2. Objetivos Específicos 1

2. Revisión de la literatura y marco teórico 1

2.1. Antecedentes 1

2.2. Segmentación de Donantes 2

2.3. Aprendizaje Automático 2

2.4. Algoritmos de Segmentación 3

2.4.1. K-means 3

2.4.2. Árbol de Decisión 3

2.4.3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 3

3. Metodología 4

3.1. Enfoque CRISP-DM 4

4. Resultados 5

4.1. Recopilación de datos. 5

4.2. Exploración de datos. 5

4.2.1. Tamaño de dataset. 5

4.2.2. Análisis de Datos nulos. 6

4.2.3. Limpieza de datos. 6

4.2.4. Reemplazo de valores NaN. 8

4.2.5. Verificación y Normalización de la columna 'idsc'. 9

4.2.6. Verificación y Normalización de la columna 'genero'. 10

4.2.7. Segmentación 11

4.2.8. Aplicación del Método del Codo para determinar el número óptimo de Clústeres. 13

5. Entrenamiento de datos 16

5.1. Algoritmo K-MEANS 16

5.2. Algoritmo de Árbol De Decisiones 18

5.3. Algoritmo DBSCAN 20

6. Conclusiones y Recomendaciones. 23

6.1. Conclusiones 23

6.2. Recomendaciones 24

Bibliografía 25

ÍNDICE DE FIGURAS

Figura 1 Tamaño de dataset. 5

Figura 2 Datos nulos dataset. 6

Figura 3 Eliminación de columnas del dataset. 7

Figura 4 Dataset después de la eliminación de las columnas. 7

Figura 5 Renombre de columnas. 8

Figura 6 Reemplazo de valores NaN en dataset. 8

Figura 7 Visualización de dataset. 9

Figura 8 Normalización de columna 'idsc'. 10

Figura 9 Normalización de columna 'genero' 11

Figura 10 Relación de variables de dataset. 12

Figura 11 Diagrama de codo de dataset. 13

Figura 12 Gráfico de agrupamiento 14

Figura 13 Gráfico Cantidad de datos de cada clúster. 15

Figura 14 Gráfico de preparación de datos para el entrenamiento. 16

Figura 15 Gráfico de estandarización de datos. 16

Figura 16 Gráfico de número de clusters a usar. 17

Figura 17 Gráfico de clústeres. 17

Figura 18 Gráfico de evaluación – Matriz de confusión y Precisión. 18

Figura 19 Gráfico de árbol de decisiones. 18

Figura 20 Gráfico de reporte de resultados. 19

Figura 21 Gráfico de evaluación – Matriz de confusión y Precisión Árbol de Decisiones. 19

Figura 22 Gráfico de DBSCAN clusters. 20

Figura 23 Gráfico de codo 21

Figura 24 Gráfico de resultado de épsilon y min_sample. 21

Figura 25 Gráfico de evaluación – DBSCAN. 22

1. Introducción

1.1. Planteamiento del Problema

En la actualidad, las fundaciones sin fines de lucro se enfrentan a un entorno cada vez más competitivo por la captación de fondos. Para poder aumentar sus ingresos, estas organizaciones necesitan encontrar formas de segmentar de manera efectiva a sus donantes potenciales.

1.2. Justificación

La aplicación de algoritmos de aprendizaje automático en este contexto ofrece la oportunidad de mejorar la precisión y eficacia de la segmentación, optimizando así las campañas de recaudación de fondos y maximizando el impacto de las actividades de la fundación.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un modelo de segmentación de donantes potenciales basado en algoritmos de aprendizaje automático.

1.3.2. Objetivos Específicos

Identificar variables clave para la segmentación basada en comportamientos y patrones de donación.

Recopilar datos relevantes sobre donantes.

Evaluar los resultados de los modelos utilizados y seleccionar el algoritmo más eficiente.

2. Revisión de la literatura y marco teórico

2.1. Antecedentes

En la actualidad, las fundaciones sin fines de lucro se enfrentan a un entorno cada vez más competitivo por la captación de fondos.

Se necesitan encontrar formas de segmentar de manera efectiva a sus donantes potenciales.

Los estudios han encontrado que los algoritmos de aprendizaje automático pueden ser una herramienta eficaz para la segmentación de donantes potenciales.

2.2. Segmentación de Donantes

La segmentación de donantes implica dividir a los potenciales donantes en basados en características compartidas, tales como boletines, campañas de donación, para la recaudación de fondos.

2.3. Aprendizaje Automático

El aprendizaje automático es una rama de la inteligencia artificial que permite a las computadoras aprender de los datos y hacer predicciones o decisiones sin estar explícitamente programadas para ello. Los algoritmos de aprendizaje automático pueden identificar patrones, hacer predicciones y tomar decisiones basadas en datos.

- **Aprendizaje supervisado:** Los algoritmos se entrenan con un conjunto de datos etiquetados y utilizan esta información para predecir etiquetas en nuevos datos. Ejemplos incluyen regresión logística y máquinas de soporte vectorial (SVM).

- **Aprendizaje no supervisado:** Los algoritmos buscan patrones en los datos sin etiquetas predefinidas. La segmentación de donantes frecuentemente utiliza técnicas no supervisadas como el análisis de conglomerados (clustering), donde los algoritmos como k-means, árbol de decisión y DBSCAN son comunes.

2.4. Algoritmos de Segmentación y Clasificación

2.4.1. K-means

El algoritmo k-means es uno de los métodos más utilizados para la segmentación no supervisada. Este algoritmo particiona el conjunto de datos en k clusters, donde cada punto de datos pertenece al cluster con la media más cercana. K-means es eficiente y fácil de implementar, pero requiere especificar el número de clusters de antemano.

2.4.2. Árbol de Decisión

El algoritmo de un árbol de decisión es de tipo supervisado no paramétrico que se aplica tanto a tareas de clasificación como de regresión. La estructura del árbol de decisión tiene un nodo raíz, ramas, nodos internos y nodos hoja.

2.4.3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN es un algoritmo de clustering basado en densidad que puede encontrar clusters de forma arbitraria y manejar ruido (puntos de datos que no pertenecen a ningún cluster). A diferencia de k-means, no requiere especificar el número de clusters a priori, pero sí necesita parámetros para definir la densidad mínima de los puntos.

3. Metodología

3.1. Enfoque CRISP-DM

Describir las etapas del enfoque CRISP-DM (Cross-Industry Standard Process for Data Mining) y cómo se aplicarán en este proyecto.

La metodología de esta investigación se basará en los siguientes pasos:

- **Recolección de datos:** Se recopilarán datos de donantes potenciales de una fundación sin fines de lucro.
- **Preparación de datos:** Se prepararán los datos para su análisis mediante el uso de técnicas de limpieza y transformación.
- **Experimentación:** Se seleccionarán algoritmos de aprendizaje automático adecuados para la tarea de segmentación de donantes como k-means, clustering jerárquico, clustering basado en densidad y modelos de mezcla gaussiana.
- **Evaluar el algoritmo:** Evaluar los resultados de los modelos utilizados y seleccionar el algoritmo más eficiente en la experimentación.

Las herramientas y software utilizados en el desarrollo de la investigación incluirán:

- **Python:** Para el desarrollo de algoritmos de aprendizaje automático.
- **Scikit-learn:** Para la implementación de algoritmos de segmentación.
- **Pandas y NumPy:** Para el manejo y preprocesamiento de datos.
- **Matplotlib y Seaborn:** Para la visualización de datos y resultados.

4. Resultados

4.1. Recopilación de datos.

Del dataset recibido se selecciona 6 variables con las que se va a trabajar y se describe a continuación.

- **idc** = ID de Cliente
- **puntaje** = Calificación que obtiene el usuario al participar en la campaña.
- **genero** = Correponde a Masculino, Femenino y NE (Son usuarios de origen genérico)
- **numcam** = Número de Campañas.
- **idgs** = ID del Grupo Suscriptor al que pertenece el usuario.
- **creado** = Año en el que se registró el usuario y está activo.

4.2. Exploración de datos.

4.2.1. Tamaño de dataset.

En la figura se puede observar que el dataset contiene 28980 registros y 28 columnas, que representas las características de los posibles donantes.

Figura 1 *Tamaño de dataset.*

```
1 # Dimensión de dataset
2 df.shape
```

```
(28980, 28)
```

4.2.2. Análisis de Datos nulos.

Al realizar el análisis del dataset se observa que varias columnas tienen valores nulos en la mayoría de las características y esto implica que los datos no fueron ingresados o no eran relevantes.

Figura 2 Datos *nulos dataset.*

```
[ ]  1 # Ver valores faltantes de las columnas
     2 df.isna().sum()
```

```
Id                                                  0
Dirección                                       28899
Ciudad                                          20620
Provincia                                       28906
País                                            28940
Puntaje                                             0
Página web                                      28890
Estado                                              0
Creado en                                           0
Actualizado en                                      0
Genero                                              0
IP de alta                                      28980
Suscrito con aceptación                             0
IP de baja                                      28980
Idioma                                              0
ID de los grupos                                  627
Nombre de los grupos                              627
Reportado como spam - ID del boletín enviado    28980
Reportado como spam - Asunto del boletín enviado 28980
Teléfono SMS                                    28973
Estado SMS                                      28973
cargo                                           27775
Email2                                          19752
Empresa                                         28480
Grupo de Suscriptores                            1299
Público                                          1658
Telefono                                        14043
Teléfono                                        15993
dtype: int64
```

4.2.3. Limpieza de datos.

Para mejorar el dataset se decidió eliminar las columnas que no son relevantes para el análisis y con alto porcentaje de valores nulos como Dirección, Ciudad, Provincia, País, Página web, Estado, Actualizado en, IP de alta, Suscrito con aceptación, IP de baja, Idioma, Nombre de los grupos, Reportado como spam - ID del boletín enviado, Reportado como spam - Asunto del

boletín enviado, Teléfono SMS, Estado SMS, cargo, Email2, Empresa, Público, Telefono y Teléfono.

Después de la eliminación de estas columnas, el dataset se redujo a 6 columnas.

Figura 3 Eliminación de columnas del dataset.

```
1 df.columns.values
```

```
array(['Id', 'Dirección', 'Ciudad', 'Provincia', 'País', 'Puntaje',
       'Página web', 'Estado', 'Creado en', 'Actualizado en', 'Genero',
       'IP de alta', 'Suscrito con aceptación', 'IP de baja', 'Idioma',
       'ID de los grupos', 'Nombre de los grupos',
       'Reportado como spam - ID del boletín enviado',
       'Reportado como spam - Asunto del boletín enviado', 'Teléfono SMS',
       'Estado SMS', 'cargo', 'Email2', 'Empresa',
       'Grupo de Suscriptores', 'Público', 'Telefono', 'Teléfono'],
      dtype=object)
```

```
[ ]  1 # Se almacena en un nuevo dataframe "df1"
     2 df1=df.drop([ 'Dirección', 'Ciudad', 'Provincia',
     3        'País', 'Página web', 'Estado',
     4        'Actualizado en', 'IP de alta',
     5        'Suscrito con aceptación', 'IP de baja', 'Idioma',
     6        'Nombre de los grupos',
     7        'Reportado como spam - ID del boletín enviado',
     8        'Reportado como spam - Asunto del boletín enviado', 'Teléfono SMS',
     9        'Estado SMS', 'cargo', 'Email2', 'Empresa',
    10        'Público', 'Telefono', 'Teléfono'], axis=1)
```

Después de la eliminación de estas columnas, el dataset se redujo a 6 columnas.

Figura 4 Dataset *después de la eliminación de las columnas.*

```
1 # Dimensión de dataset despues de la eliminación de las columnas.
2 df1.shape
```

```
(28980, 6)
```

Se decide renombrar algunas columnas para facilitar la comprensión y manipulación de datos.

Se realizó el siguiente cambio:

- **Id se renombró a idc:** Simplifica el identificador asignado al donante.

- **Puntaje se renombró a puntaje:** Mantiene el mismo nombre, pero en minúsculas para consistencia.
- **Género se renombró a genero:** S elimina el acento para facilitar la manipulación de datos en Python.
- **ID de los grupos se renombró a idg:** Se abrevia el nombre de la columna para evitar nombres largos.
- **Grupo de Suscriptores se renombró a idsc:** También se abrevia para hace más legible.

Los nombres de las columnas son mas cortos sin acentos, espacios y extensos que ayuda a reducir el riesgo de errores.

Figura 5 Renombre de columnas.

```
# Se renombra las columnas
df1.rename(columns={'Id':'idc','Puntaje':'puntaje','Genero':'genero','ID de los grupos':'idg','Grupo de Suscriptores':'idsc'}, inplace=True)
```

4.2.4. Reemplazo de valores NaN.

Se identificó la presencia de valores NaN (nulos) en el dataset y para resolver el problema se decidió reemplazar todos los valores NaN por 0, esto evita perdidas de registros y ayuda a mantener la integridad del dataset.

Figura 6 Reemplazo de valores NaN en dataset.

```
# Se reemplaza los valores que tienen NaN
df1.fillna(value=0, inplace=True)
```

Después de aplicar el reemplazo de valores NaN se visualiza el dataset con los registros con 0.

Figura 7 Visualización de dataset.

```
[ ]  1 # Ver dataset
     2 df1
```

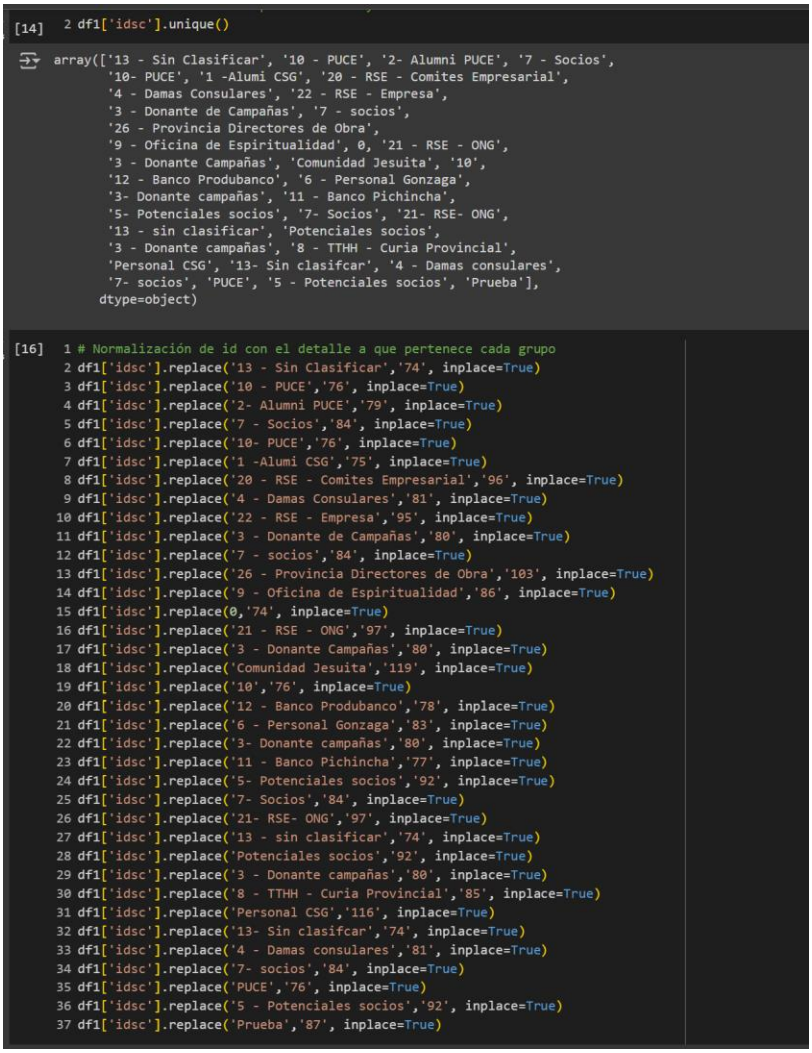
	idc	puntaje	Creado en	genero	idg	Grupo de Suscriptores	numcam
0	1	1	13/09/2018 17:41:06	NE	74,125	13 - Sin Clasificar	2
1	54	1	16/09/2018 21:01:00	Masculino	76,125	10 - PUCE	2
2	65	1	16/09/2018 21:01:00	Masculino	76,125	10 - PUCE	2
3	67	1	16/09/2018 21:01:00	Masculino	76,125	10 - PUCE	2
4	68	1	16/09/2018 21:01:00	Femenino	76,125	10 - PUCE	2
...	...	...	...	...	...	...	...
28975	24676	4	07/02/2024 12:42:01	Femenino	84	7 - Socios	1
28976	24677	4	08/02/2024 12:17:24	Femenino	116	0	1
28977	24678	4	19/02/2024 8:57:13	Masculino	123	0	1
28978	24679	4	01/03/2024 12:21:03	Femenino	84	7 - Socios	1
28979	24681	4	14/03/2024 17:52:52	Femenino	84	7 - Socios	1

28980 rows × 7 columns

4.2.5. Verificación y Normalización de la columna ‘idsc’.

La columna ‘idsc’ es la referencia de Grupos de Suscriptores la misma que contienen inconsistencias que puede dificultar el análisis, por lo que se realizó el proceso de normalización colocando un código numérico único que representa al Grupos de Suscriptores al que pertenece.

Figura 8 Normalización de columna ‘idsc’.



```
[14]  2 df1['idsc'].unique()

array(['13 - Sin Clasificar', '10 - PUCE', '2- Alumni PUCE', '7 - Socios',
       '10- PUCE', '1 -Alumi CSG', '20 - RSE - Comites Empresarial',
       '4 - Damas Consulares', '22 - RSE - Empresa',
       '3 - Donante de Campañas', '7 - socios',
       '26 - Provincia Directores de Obra',
       '9 - Oficina de Espiritualidad', 0, '21 - RSE - ONG',
       '3 - Donante Campañas', 'Comunidad Jesuita', '10',
       '12 - Banco Produbanco', '6 - Personal Gonzaga',
       '3- Donante campañas', '11 - Banco Pichincha',
       '5- Potenciales socios', '7- Socios', '21- RSE- ONG',
       '13 - sin clasificar', 'Potenciales socios',
       '3 - Donante campañas', '8 - TTHH - Curia Provincial',
       'Personal CSG', '13- Sin clasifcar', '4 - Damas consulares',
       '7- socios', 'PUCE', '5 - Potenciales socios', 'Prueba'],
      dtype=object)

[16]  1 # Normalización de id con el detalle a que pertenece cada grupo
      2 df1['idsc'].replace('13 - Sin Clasificar','74', inplace=True)
      3 df1['idsc'].replace('10 - PUCE','76', inplace=True)
      4 df1['idsc'].replace('2- Alumni PUCE','79', inplace=True)
      5 df1['idsc'].replace('7 - Socios','84', inplace=True)
      6 df1['idsc'].replace('10- PUCE','76', inplace=True)
      7 df1['idsc'].replace('1 -Alumi CSG','75', inplace=True)
      8 df1['idsc'].replace('20 - RSE - Comites Empresarial','96', inplace=True)
      9 df1['idsc'].replace('4 - Damas Consulares','81', inplace=True)
     10 df1['idsc'].replace('22 - RSE - Empresa','95', inplace=True)
     11 df1['idsc'].replace('3 - Donante de Campañas','80', inplace=True)
     12 df1['idsc'].replace('7 - socios','84', inplace=True)
     13 df1['idsc'].replace('26 - Provincia Directores de Obra','103', inplace=True)
     14 df1['idsc'].replace('9 - Oficina de Espiritualidad','86', inplace=True)
     15 df1['idsc'].replace(0,'74', inplace=True)
     16 df1['idsc'].replace('21 - RSE - ONG','97', inplace=True)
     17 df1['idsc'].replace('3 - Donante Campañas','80', inplace=True)
     18 df1['idsc'].replace('Comunidad Jesuita','119', inplace=True)
     19 df1['idsc'].replace('10','76', inplace=True)
     20 df1['idsc'].replace('12 - Banco Produbanco','78', inplace=True)
     21 df1['idsc'].replace('6 - Personal Gonzaga','83', inplace=True)
     22 df1['idsc'].replace('3- Donante campañas','80', inplace=True)
     23 df1['idsc'].replace('11 - Banco Pichincha','77', inplace=True)
     24 df1['idsc'].replace('5- Potenciales socios','92', inplace=True)
     25 df1['idsc'].replace('7- Socios','84', inplace=True)
     26 df1['idsc'].replace('21- RSE- ONG','97', inplace=True)
     27 df1['idsc'].replace('13 - sin clasificar','74', inplace=True)
     28 df1['idsc'].replace('Potenciales socios','92', inplace=True)
     29 df1['idsc'].replace('3 - Donante campañas','80', inplace=True)
     30 df1['idsc'].replace('8 - TTHH - Curia Provincial','85', inplace=True)
     31 df1['idsc'].replace('Personal CSG','116', inplace=True)
     32 df1['idsc'].replace('13- Sin clasifcar','74', inplace=True)
     33 df1['idsc'].replace('4 - Damas consulares','81', inplace=True)
     34 df1['idsc'].replace('7- socios','84', inplace=True)
     35 df1['idsc'].replace('PUCE','76', inplace=True)
     36 df1['idsc'].replace('5 - Potenciales socios','92', inplace=True)
     37 df1['idsc'].replace('Prueba','87', inplace=True)
```

4.2.6. Verificación y Normalización de la columna ‘genero’.

Se identificó que la columna ‘genero’ contenía valores inconsistentes en el formato entre minúsculas y mayúsculas, se decidió utilizar abreviaturas “M” para masculino, “F” para femenino y “NE” para no especificado.

Después de la normalización se verifica y como resultado queda en tres categorías “NE”, “M” y “F”.

Figura 9 Normalización de columna ‘genero’

```
[ ]  1 # Visualizar los datos de la columna "genero" para normalizar
     2 df2['genero'].unique()

array(['NE', 'Masculino', 'Femenino', 'masculino', 'femenino'],
      dtype=object)

[ ]  1 # Normalización para que se visualice "Masculino" - "Femenino"
     2 df2['genero'].replace('masculino','M', inplace=True)
     3 df2['genero'].replace('femenino','F', inplace=True)
     4 df2['genero'].replace('Femenino','F', inplace=True)
     5 df2['genero'].replace('Masculino','M', inplace=True)

[ ]  1 # Se verifica la normalización
     2 df2['genero'].unique()

array(['NE', 'M', 'F'], dtype=object)
```

4.2.7. Segmentación

Se realiza la relación de las variables para la segmentación de los donantes y entender las relaciones con las variables que influyen en el comportamiento de los donantes.

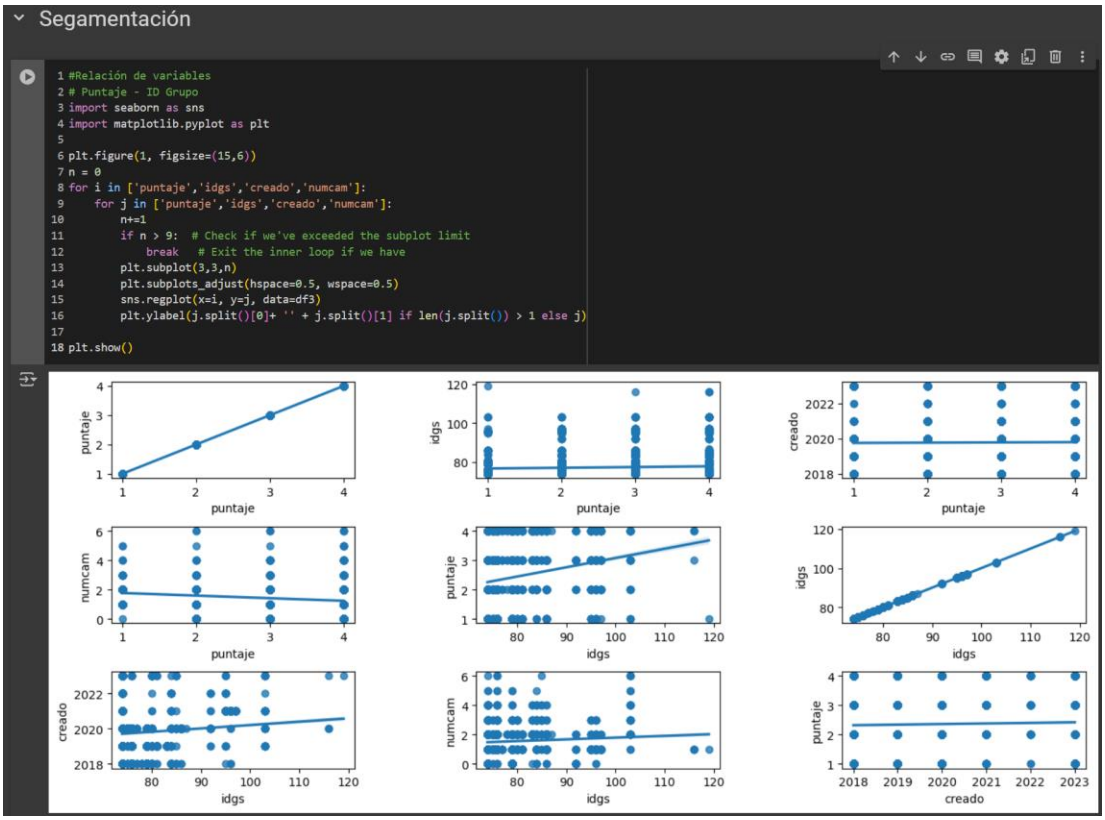
Las variables ‘puntaje’, ‘idsc’, ‘creado’ y ‘numcam’ fueron seleccionadas para la generación de los gráficos y visualizar su comportamiento, aplicando la librería ‘seaborn’ que permite crear gráficos de regresión.

Las variables seleccionadas se explican a continuación.

- **puntaje:** Indica la valoración o importancia de la participación del donante en función a las campañas enviadas por la fundación
- **idsc:** Representa el grupo al que pertenece el donante.
- **creado:** Indica el tiempo de la actividad de los donantes.

- **numcam:** Indica la cantidad de campañas en las que ha participado el donante y la relación con el puntaje puede ayudar a identificar a los donantes más activos.

Figura 10 Relación de variables de dataset.

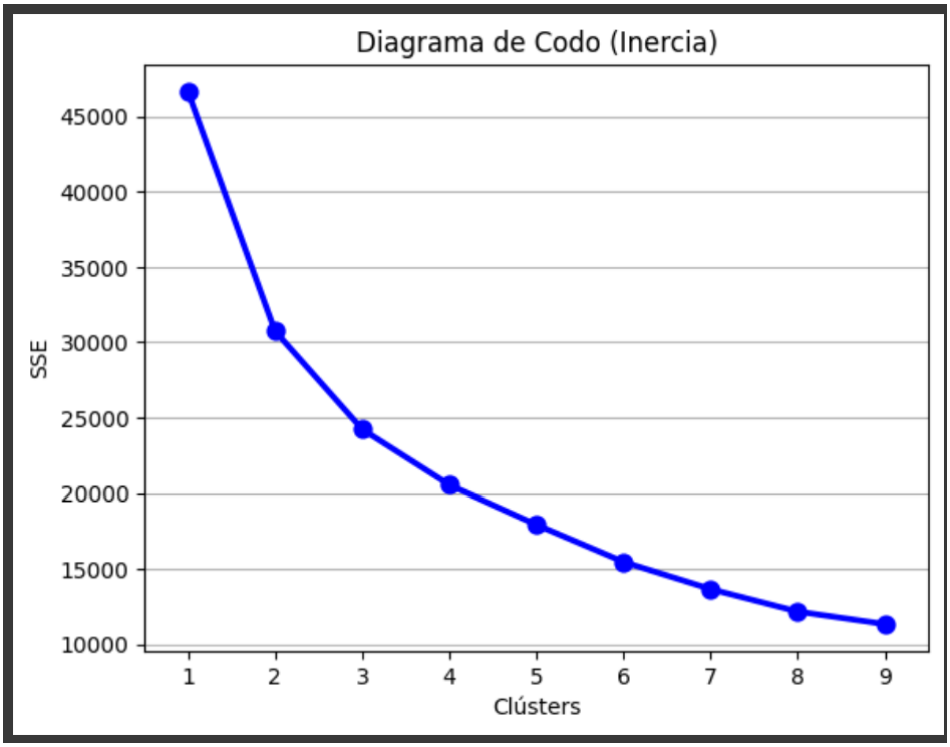


Los resultados obtenidos de los gráficos pueden servir como base para aplicar técnicas de segmentación como el clustering, permitiendo una clasificación más optima de los grupos de donantes según su comportamiento y características.

4.2.8. Aplicación del Método del Codo para determinar el número óptimo de Clústeres.

Para determinar el número optimo de clústeres se aplica el Método del Codo utilizando el algoritmo de K-Means con el que se evaluó un rango de 1 a 10 clústeres para determinar el punto óptimo.

Figura 11 Diagrama de codo de dataset.

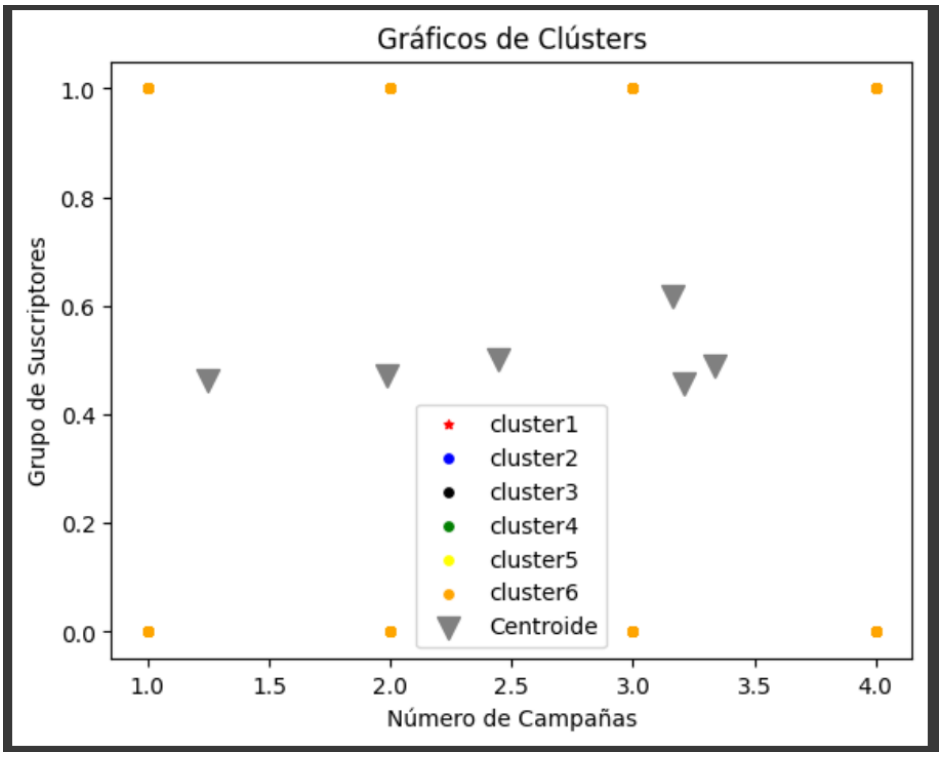


Previo al resultado la curva muestra en el punto 6, este valor se utilizará como la base para la segmentación de donantes en el análisis de del conjunto de datos.

Después de determinar el número óptimo de clústeres se utiliza el algoritmo de K-Means y se trabaja con los tres clústeres en el conjunto de datos.

En el gráfico se puede visualizar los tres clústeres identificados y los centroides marcados con un asterisco verde que identifica a los grupos donantes con característica similares.

Figura 12 Gráfico de agrupamiento



El resultado muestra la cantidad de datos que tiene cada clúster en el que se puede interpretar lo siguiente:

- **Clúster 1**: 4.018 observaciones
- **Clúster 2:** 5.182 observaciones
- **Clúster 3:** 5.933 observaciones
- **Clúster 4:** 2.609 observaciones
- **Clúster 5:** 2.951 observaciones
- **Clúster 6:** 378 observaciones

Los clústeres 1 al 5 son significativamente más grandes en comparación con el clúster 6, con la mayoría de las observaciones del dataset que indican las características similares de los donantes. El clúster 6 es mas pequeño y representa que es un segmento más centrado con características mas específicas de los donantes.

Figura 13 Gráfico Cantidad de datos de cada clúster.

```
Cantidad de datos en cluster 1: 4018
Cantidad de datos en cluster 2: 5182
Cantidad de datos en cluster 3: 5933
Cantidad de datos en cluster 4: 2609
Cantidad de datos en cluster 5: 2951
Cantidad de datos en cluster 6: 378
```

Con los resultados obtenidos se procede a la preparación de los datos para el entrenamiento aplicando la librería 'sklearn' y la división de datos de entrenamiento (X_train, y_train) y prueba (X_test, y_test).

El tamaño o porción que corresponde al test_size, en la que se especifica el 30% para el test, es decir test_size=0.3.

El parámetro random_state es para establecer la semilla de números aleatorios que en el análisis se utilizó 42, que queda de la siguiente manera random_state=42.

Se aplica la normalización de las variables relevantes para evitar que las demás variables dominen la segmentación.

Figura 14 Gráfico de preparación de datos para el entrenamiento.

5. Entrenamiento de datos

5.1. Algoritmo K-MEANS

Se prepara las columnas que son numéricas para usar clustering que son las siguientes columnas:

- **idsc.** Grupo de Suscriptores.
- **numcam.** Número de Campañas.
- **creado.** Año de registro y aun esta activo
- **genero.** Corresponde a Masculino, Femenino y NE (Son usuarios de origen genérico)

Se realiza la estandarización los datos.

Figura 15 Gráfico de estandarización de datos.

```
1 from sklearn.preprocessing import StandardScaler
2
3 # Selecciona solo las columnas numéricas que quieres usar para clustering
4 features = df3[['idsc', 'numcam', 'creado', 'genero']]
5 scaler = StandardScaler()
6 scaled_features = scaler.fit_transform(features)
```

Una vez que los datos de las columnas seleccionadas fueron estandarizados, se aplicó el algoritmo K-Means para la segmentación. El número óptimo de clústeres se determinó utilizando

el diagrama de codo (inercia), el cual sugirió que 3 clústeres era la elección más adecuada para el conjunto de datos.

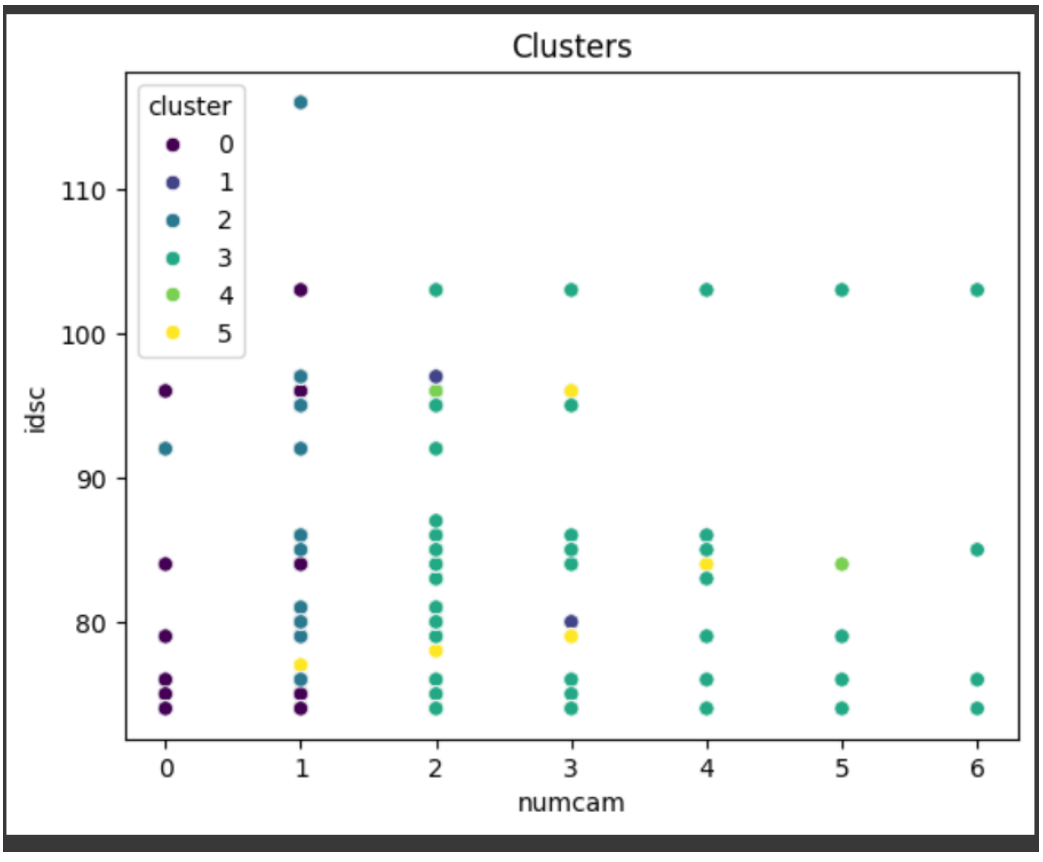
Figura 16 Gráfico de número de clusters a usar.

```
# Aplicando K-Means
kmeans = KMeans(n_clusters=6, random_state=42) # Se agrega los 6 clusteres
kmeans.fit(features_scaled)

# Añadir las etiquetas de los clusters al DataFrame original
df3['cluster'] = kmeans.labels_
```

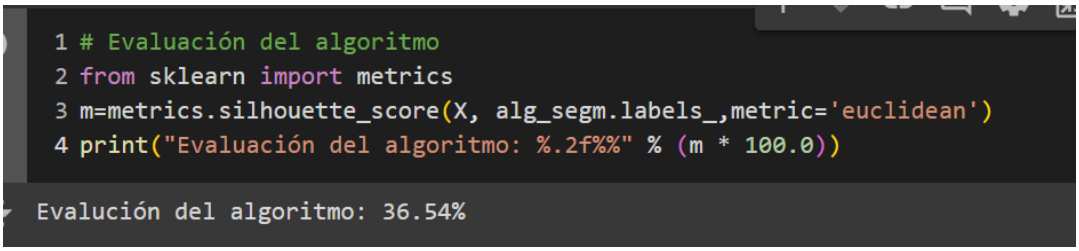
Como resultado el gráfico de dispersión indica como los donantes están distribuidos por cada punto de clúster al que pertenece.

Figura 17 Gráfico de clústeres.



La evaluación del algoritmo indica que el modelo tiene una precisión del 36.54%.

Figura 18 Gráfico de evaluación.



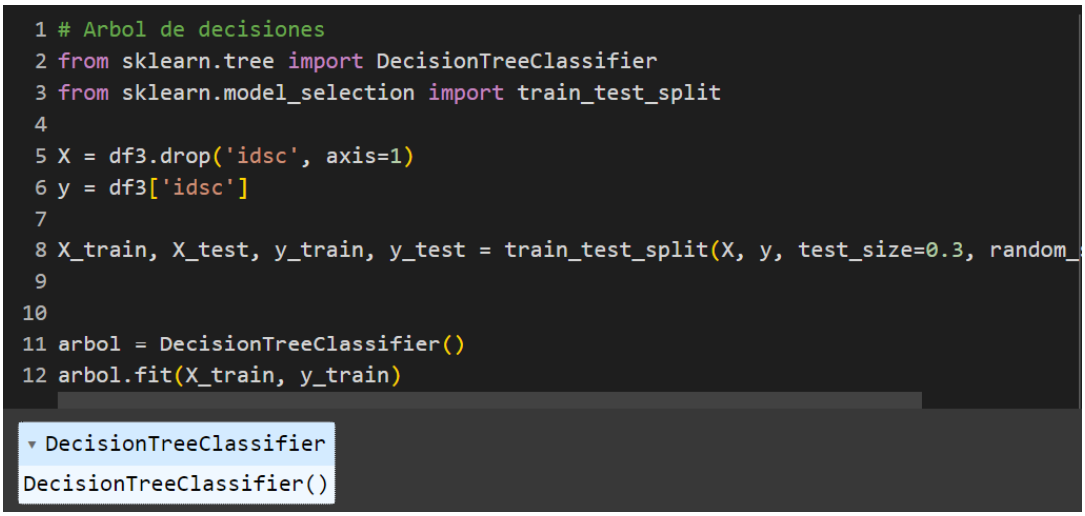
```
# Evaluación del algoritmo
from sklearn import metrics
m=metrics.silhouette_score(X, alg_segm.labels_,metric='euclidean')
print("Evaluación del algoritmo: %.2f%%" % (m * 100.0))
```

```
Evalución del algoritmo: 36.54%
```

5.2. Algoritmo de Árbol De Decisiones

Se utiliza 'DecisionTreeClassifier' para el entramiento de los datos y evaluar el resultado dividiendo en subconjuntos pequeños de acuerdo a las características que encuentre en el dataset.

Figura 19 Gráfico de árbol de decisiones.



```
# Arbol de decisiones
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

X = df3.drop('idsc', axis=1)
y = df3['idsc']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_

arbol = DecisionTreeClassifier()
arbol.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier()
```

El siguiente resultado del conjunto de datos en el que se observa que los ítems 116, 103, 84 y 85 tienen puntuaciones muy bajas, lo que puede afectar en la clasificación de las clases.

Figura 20 Gráfico de reporte de resultados.

```
1 print(classification_report(y_test, predicciones1))
```

	precision	recall	f1-score	support
74	0.61	0.62	0.61	1277
75	0.93	0.92	0.93	93
76	0.90	0.92	0.91	420
77	1.00	0.25	0.40	4
78	1.00	1.00	1.00	3
79	0.88	0.88	0.88	4210
80	0.67	0.72	0.69	47
81	0.83	0.83	0.83	12
83	0.98	1.00	0.99	53
84	0.55	0.49	0.51	37
85	0.88	0.88	0.88	8
86	0.97	0.90	0.94	41
92	0.67	0.44	0.53	9
95	0.94	0.81	0.87	42
96	0.90	0.96	0.93	49
97	1.00	1.00	1.00	11
103	0.31	0.80	0.44	5
116	0.00	0.00	0.00	1
accuracy			0.82	6322
macro avg	0.78	0.75	0.74	6322
eighted avg	0.83	0.82	0.82	6322

La evaluación de algoritmo tiene una precisión del 82.41%, lo que indica que el modelo es aceptable.

Figura 21 Gráfico de evaluación – Árbol de Decisiones.

```
1 # Evaluación del algoritmo - Accuracy
2 from sklearn.metrics import accuracy_score
3
4 accuracy=accuracy_score(y_test, predicciones1)
5 print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

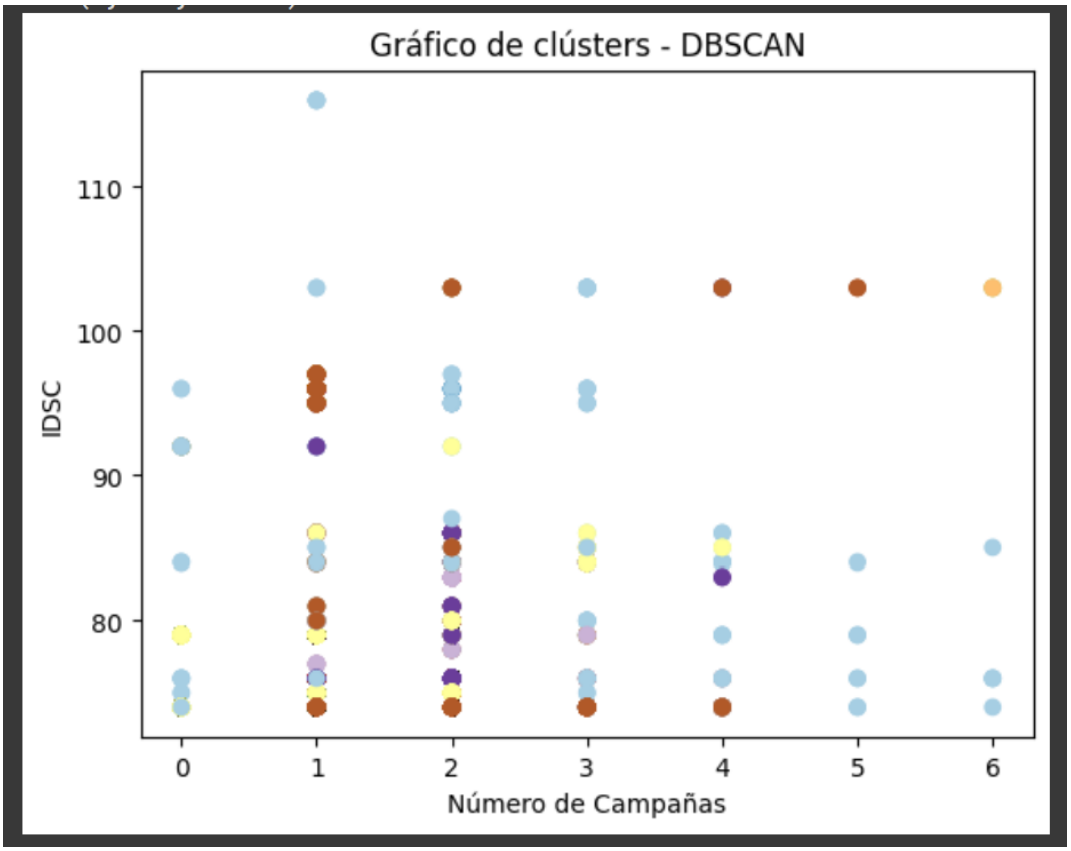
```
Accuracy: 82.41%
```

5.3. Algoritmo DBSCAN

Se procede a implementar el algoritmo DBSCAN para validar el resultado y comparar con los algoritmos anteriores.

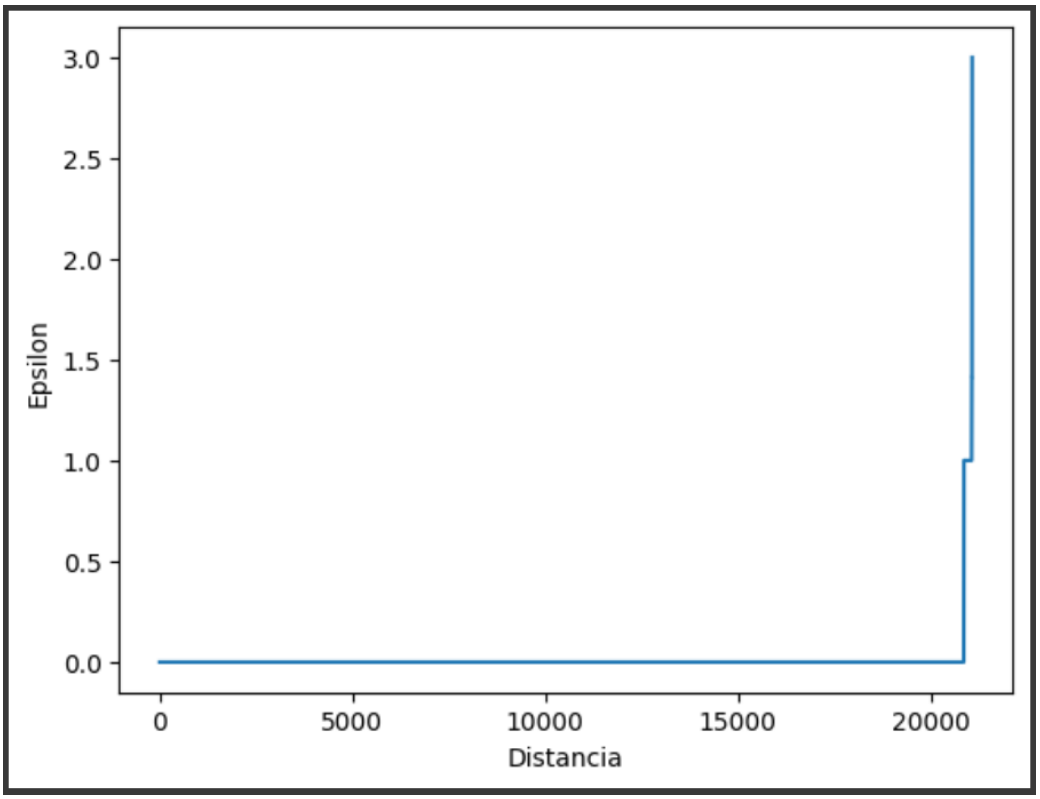
Se genera el gráfico en el que se visualiza los clústeres creados y cada punto en está con un color según al clúster que pertenece.

Figura 22 Gráfico de DBSCAN clusters.



Se calcula la distancia entre los puntos mas cercanos para determinar el valor adecuado de ‘epsilon’ que es 0.3 y min_samples=2 para ver el número estimado de conglomerados y puntos de ruido.

Figura 23 Gráfico de codo.



Después de identificar los conglomerados y puntos de ruido se obtiene el siguiente resultado.

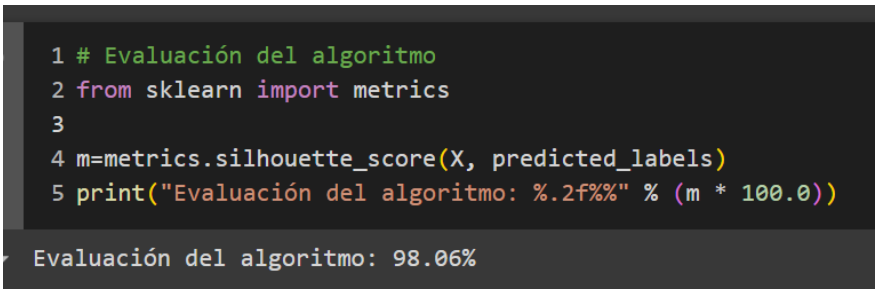
- **350 Clústeres:** Indica que los datos se han agrupado en un gran número de grupos, lo que puede ser evidencia de una estructura compleja en los datos.
- **219 Puntos de Ruido:** Se considera como outliers, es decir, puntos que no se agrupan con ninguno de los conglomerados.

Figura 24 Gráfico de resultado de épsilon y min_sample.

```
Número estimado de conglomerados: 350
Número estimado de puntos de ruido: 219
```

La evaluación del algoritmo tiene una precisión del algoritmo que es de 98.06%, lo indica que el resultado es el esperado.

Figura 25 Gráfico de evaluación – DBSCAN.



```
# Evaluación del algoritmo
from sklearn import metrics

m=metrics.silhouette_score(X, predicted_labels)
print("Evaluación del algoritmo: %.2f%%" % (m * 100.0))
```

```
Evaluación del algoritmo: 98.06%
```

6. Conclusiones y Recomendaciones.

6.1. Conclusiones

- Se implementaron y compararon los algoritmos K-Means, Árbol de Decisiones y DBSCAN para la segmentación de donantes potenciales, cada uno de estos algoritmos ofreció diferentes perspectivas y resultados, lo que permitió una evaluación para determinar qué modelo es más eficiente.
- Las variables seleccionadas, como el número de campañas, el género, y el grupo de suscriptores, demostraron ser efectivas para la segmentación de donantes. Estas variables reflejan comportamientos y participación de los donantes en las campañas realizadas por la fundación y así evaluar su impacto en la segmentación.
- La aplicación del algoritmo K-Means permitió agrupar a los donantes en tres clústers y ver el comportamiento según los patrones que el modelo clasificó con una precisión del 36.54% lo que indica un ajuste o mejora.
- El algoritmo de Árbol de Decisiones muestra una precisión razonable con el 82.41% lo que podría no ser el algoritmo adecuado.
- La aplicación del algoritmo DBSCAN tiene una exactitud del 98.06% lo que indica que el algoritmo identificó características comunes y ser mas eficiente en la segmentación de donantes.

6.2. Recomendaciones

- Dado que K-Means, Árbol de Decisiones, y DBSCAN presentaron resultados diversos, se sugiere realizar una selección más de características sobre el comportamiento de los donantes, como el historial de donaciones, la interacción con campañas, y la fidelidad. Esto podría mejorar la capacidad de los modelos para identificar patrones y segmentar efectivamente.
- Es fundamental que los datos recopilados estén completos, actualizados y garantizar que los registros de los donantes estén completos
- Se recomienda explorar otros algoritmos de segmentación, que podrían ofrecer una segmentación más precisa y manejar mejor la variabilidad y la complejidad de los datos de donantes.

Bibliografía

Sandoval, L. (2018). Algoritmos de aprendizaje automático para análisis y predicción de datos. Revista Tecnológica, (11), 36-40. *http://redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf*

Moreno, A., Armengol, E., Béjar Alonso, J., Belanche Muñoz, L. A., Cortés García, C. U., Gavaldà Mestre, R., ... & Sànchez-Marrè, M. (1994). Aprendizaje automático. *https://upcommons.upc.edu/bitstream/handle/2099.3/36157/9788483019962.pdf?sequence=1&isAllowed=y*

1.10. Decision Trees. (n.d.). Scikit-Learn. Retrieved August 23, 2024, from https://scikit-learn.org/stable/modules/tree.html

Arce, J. I. B. (2019, July 26). La matriz de confusión y sus métricas. Juan Barrios; Juan Ignacio Barrios Arce. https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

DBSCAN. (n.d.). Scikit-Learn. Retrieved August 23, 2024, from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

KMeans. (n.d.). Scikit-Learn. Retrieved August 23, 2024, from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

¿Qué es el aprendizaje no supervisado? (2023, May 4). Ibm.com. https://www.ibm.com/es-es/topics/unsupervised-learning

¿Qué es el aprendizaje supervisado? (2024, May 14). Ibm.com. https://www.ibm.com/mx-es/topics/supervised-learning

Rodríguez, D. (2023, June 9). Método del codo (Elbow method) para seleccionar el número óptimo de clústeres en K-means. Analytics Lane. https://www.analyticslane.com/2023/06/09/metodo-del-codo-elbow-method-para-seleccionar-el-numero-optimo-de-clusteres-en-k-means/

SPSS Modeler Subscription. (2021, August 17). Ibm.com. https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview

(2023 4). Unir.net. https://ecuador.unir.net/actualidad-unir/clustering-datos/