INTRODUCCIÓN

PROYECTO DE MACHINE LEARNING CLUSTERING

- REALIZADO POR: IVAN GONZALO TAPIA
- INSTITUCIÓN: UNIVERSIDAD PROVINCIAL DEL SUDESTE
- MATERIA: BIG DATA Y APRENDIZAJE DE MÁQUINA
- DOCENTE: ING. VALENTÍN BARCO
- LUGAR: PUNTA ALTA
- FECHA: 20 DE NOVIEMBRE DE 2023

- REPOSITORIO PROYECTO: GITHUB PROYECTO 3 MACHINE LEARNING CLUSTERING
- INFORME: INFORME EN WORD
- NOTEBOOK: COLAB

JUSTIFICACIÓN DEL INFORME Y PROPÓSITOS

NOS CENTRAREMOS EN UN CONJUNTO DE DATOS QUE INCLUYE REGISTROS DE CRÍMENES DE 1973 EN DISTINTOS ESTADOS DE EE. UU., ASÍ COMO EL PORCENTAJE DE LA POBLACIÓN QUE RESIDE EN ZONAS URBANAS EN ESOS ESTADOS. EL PROPÓSITO ES CLASIFICAR LOS ESTADOS EN DIFERENTES CATEGORÍAS BASADAS EN ESTOS FACTORES.

1- IMPORTACIÓN DE LIBRERIAS, CARGA Y PREPROCESAMIENTO DE DATOS.
2- VISUALIZACIÓN DE DATOS CON MATPLOTLIB Y SEABORN.
3- CREACIÓN DE MODELO KMEANS DE SKLEARN.
4- CREACIÓN DE MODELO MEAN SHIFT DE SKLEARN.
5- CREACIÓN DE MODELO DBSCAN DE SKLEARN.

1.1 Importación de librerías:

Se importaron las librerías necesarias para el análisis de datos, incluyendo Pandas para la manipulación eficiente de datos, NumPy para operaciones numéricas y el módulo LabelEncoder de SKLearn para la codificación de variables categóricas.

1.2 Carga de datos:

El conjunto de datos se cargó desde un archivo CSV a un DataFrame utilizando la librería Pandas. Este paso nos permitió acceder y trabajar con la información de manera estructurada ver Figura 1.

```
df = pd.read_csv(path)    #cargar el archivo csv en un df
df_crimenes = df.copy()   #copia del df original
df_crimenes.head()        #observamos un panorama de los datos
```

	Estado	Asesinatos	Asaltos	Poblacion Urbana	Violaciones
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6

Figura 1: Carga de datos desde .csv con Pandas.

1.3 Selección de características relevantes:

Identificamos y seleccionamos las características pertinentes para nuestro análisis. En este caso, nos enfocamos en las variables 'Asesinatos', 'Asaltos', 'Poblacion Urbana' y 'Violaciones', las cuales consideramos cruciales para nuestro modelo de machine learning.

1.4 Codificación de variables categóricas:

Para poder incorporar la variable categórica 'Estado' en nuestros modelos de machine learning, aplicamos una codificación numérica utilizando el método Label Encoding del módulo LabelEncoder de SKLearn. Este paso es fundamental, ya que muchos algoritmos de machine learning requieren variables numéricas para su procesamiento.

Estos primeros pasos sientan las bases para el análisis y modelado subsiguiente, permitiéndonos trabajar con datos preparados y relevantes para los objetivos de nuestro proyecto. En las secciones posteriores, exploraremos técnicas más avanzadas y construiremos nuestro modelo de machine learning para abordar preguntas específicas y obtener conclusiones significativas.

2. Exploración y Visualización de Datos:

En esta etapa del proyecto, nos enfocamos en explorar y visualizar el conjunto de datos a trabajar con el objetivo de obtener una comprensión inicial de las relaciones entre las diferentes características. Para ello, empleamos herramientas de visualización como seaborn y matplotlib.

2.1 Visualización mediante Pair Plots:

Se utilizó la función pairplot de la librería seaborn para crear gráficos de dispersión entre pares de características. Este enfoque nos proporcionó una visión detallada de la distribución y relaciones entre las variables seleccionadas ('Asesinatos', 'Asaltos', 'Poblacion Urbana', 'Violaciones') ver Figura 2.

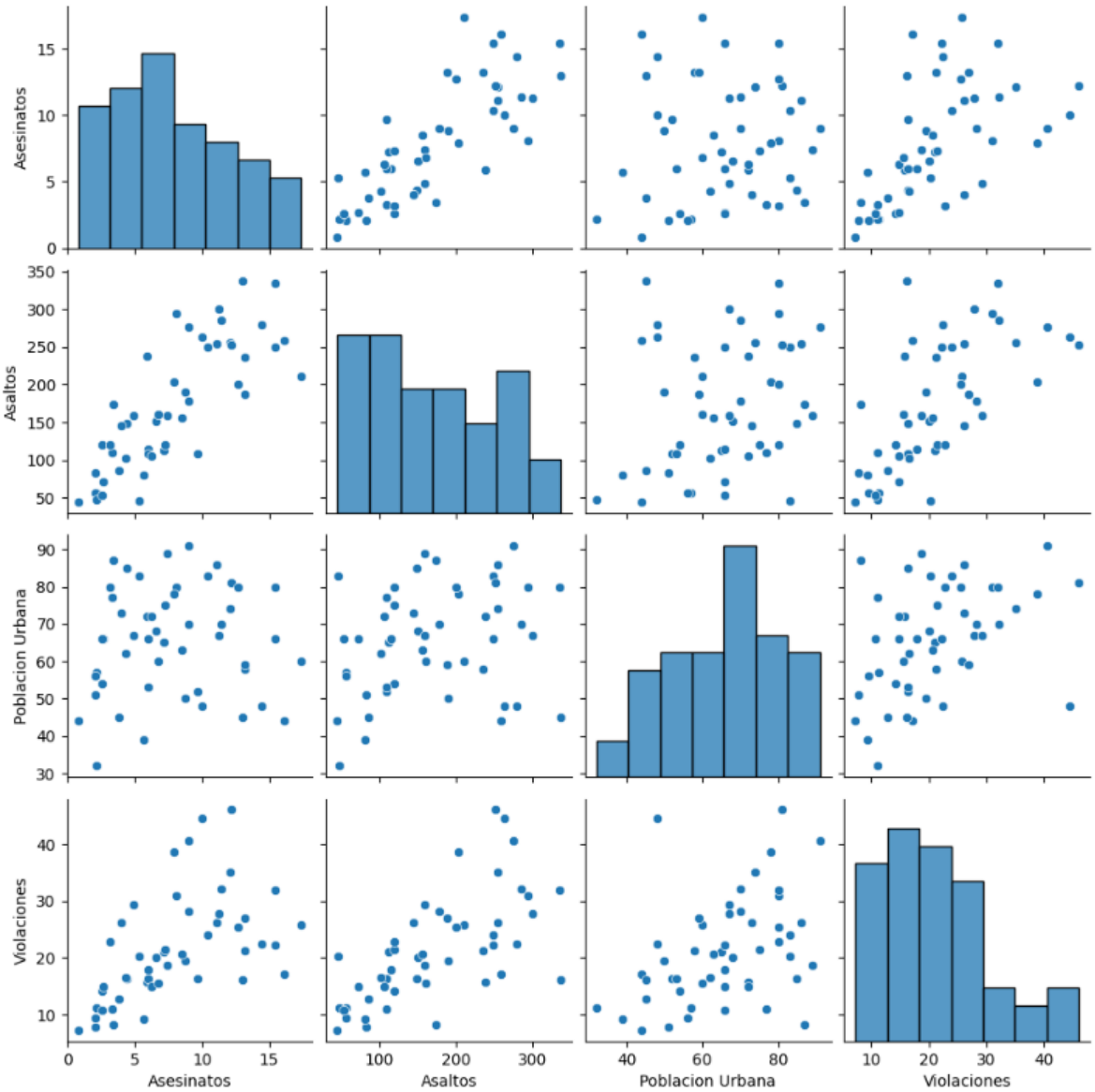

Figura 2: Gráfico realizado con pairplot de las variables cruciales.

2.2 Visualización con Hue por 'Poblacion Urbana':

Para profundizar en nuestra comprensión, se creó un segundo gráfico de dispersión utilizando pairplot, pero esta vez incorporando el matiz ('hue') basado en la variable 'Poblacion Urbana'. Esto nos permitió ver cómo la población urbana podía influir en las relaciones entre las otras variables, y observamos que no resulta tan relevante ver Figura 3.

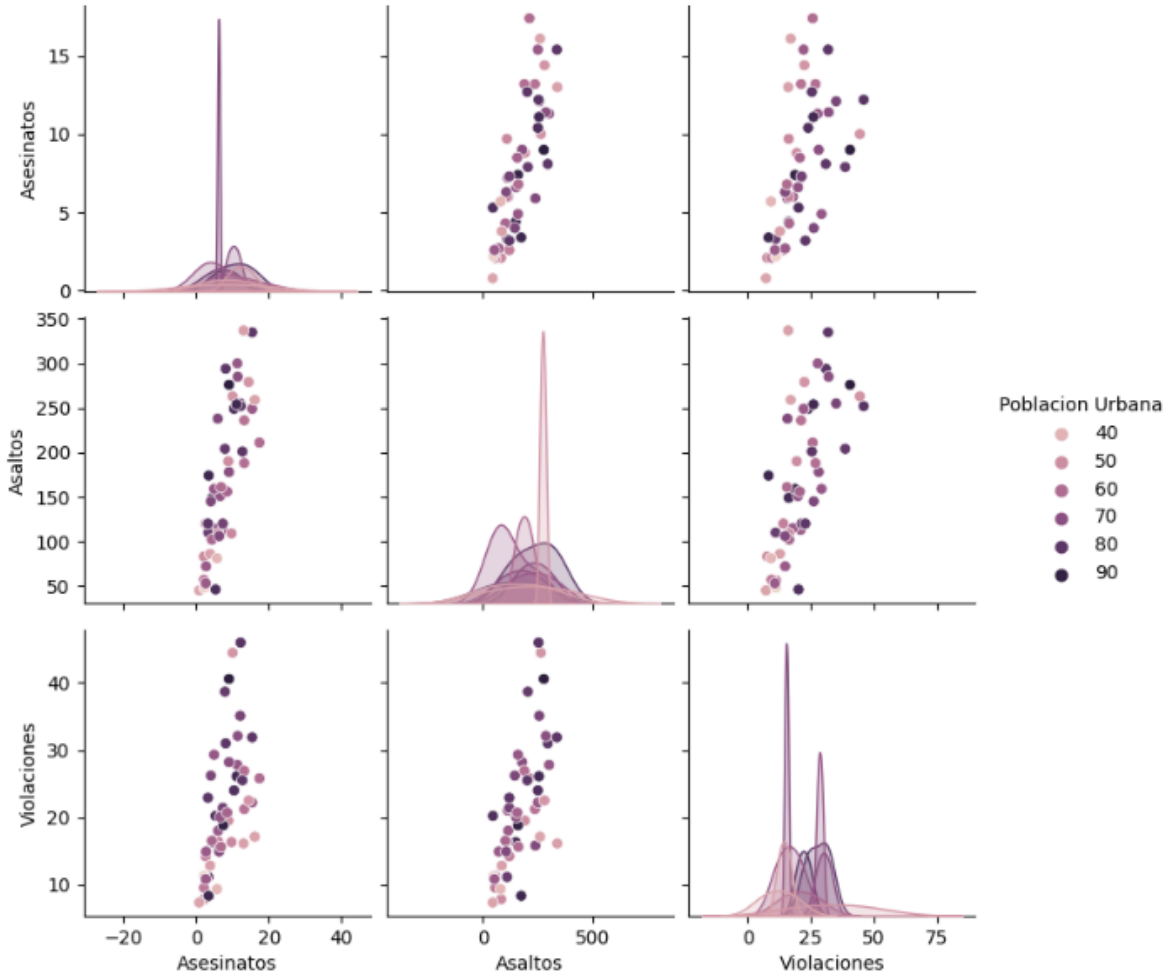

Figura 3: Gráfico pairplot con influencia de Población Urbana.

Estos gráficos de dispersión proporcionan una visión gráfica de las distribuciones y relaciones entre las variables seleccionadas. El análisis visual es crucial para identificar posibles patrones, tendencias y anomalías en los datos, y sienta las bases para decisiones posteriores en el proceso de modelado de machine learning. En las secciones siguientes, nos sumergiremos más profundamente en el análisis y construcción de nuestro modelo predictivo.

3. Creación del Modelo KMeans de SKLearn:

En esta etapa del proyecto, avanzamos en la creación y aplicación del modelo de agrupamiento KMeans utilizando la implementación proporcionada por la librería SKLearn.

nos propusimos identificar el número más adecuado de clusters para nuestro modelo de KMeans, utilizando una técnica conocida como el "Método del Codo" (Elbow Method). El

objetivo es encontrar el punto en el que la suma de los errores cuadrados (SSE) disminuye significativamente, indicando el número óptimo de clusters.

3.1 Aplicación del Método del Codo:

Se empleó el algoritmo KMeans del módulo sklearn.cluster para realizar agrupamientos en el conjunto de datos. Iteramos sobre un rango de posibles números de clusters (k) y calculamos la suma de errores cuadrados (SSE) para cada k. Los resultados se visualizaron en un gráfico, donde se observa la relación entre el número de clusters y la SSE.

Este proceso nos permitió identificar un punto en la curva donde la SSE comienza a estabilizarse, indicando el número óptimo de clusters. En el ejemplo proporcionado ver Figura 4, la curva del codo sugirió que el número ideal de clusters para nuestro conjunto de datos es 3.

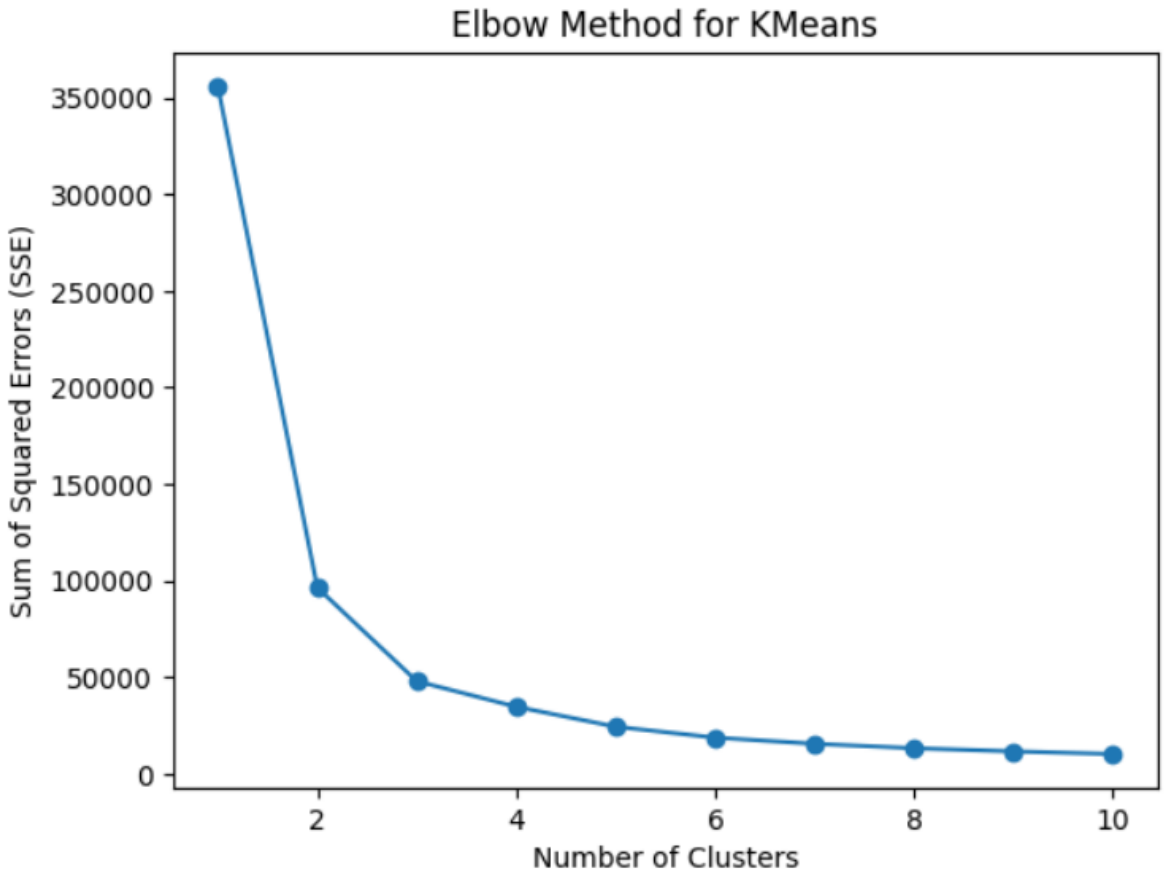

Figura 4: Curva del codo, ayuda a determinar el número de Clusters a utilizar.

Este análisis es crucial para determinar el nivel de partición más apropiado en los datos, proporcionando información valiosa para la fase siguiente del proyecto, donde se implementarán y evaluarán los modelos de agrupamiento. En las siguientes secciones, profundizaremos en la aplicación y análisis de los resultados de KMeans con el número óptimo de clusters identificado

3.2 Implementación del Modelo KMeans con Número Óptimo de Clusters:

Con el número óptimo de clusters identificado como 3 mediante el Método del Codo, procedimos a la creación y aplicación del modelo KMeans a nuestro conjunto de datos.

3.2.1 Instanciación del Modelo KMeans:

Creamos una instancia del modelo KMeans de la librería SKLearn, especificando el número óptimo de clusters identificado anteriormente, que en este caso fue 3. La inicialización se llevó a cabo con una semilla aleatoria (random_state=42) para garantizar reproducibilidad.

3.2.2 Ajuste del Modelo a los Datos:

Aplicamos el modelo KMeans al conjunto de datos utilizando la función fit_predict, lo que nos permitió asignar cada punto de datos a un cluster particular. La información de los clusters resultantes se incorporó al DataFrame original para futuros análisis.

3.2.3 Etiquetado de Niveles de Seguridad:

Para facilitar la interpretación de los resultados, asignamos etiquetas descriptivas a cada cluster. En este caso, se utilizaron las etiquetas "Precaución," "Seguro," y "Peligroso" para representar los clusters 0, 1 y 2, respectivamente. Estas etiquetas se añadieron como una nueva columna llamada 'Nivel_Seguridad' al DataFrame.

3.2.4 Visualización de los Resultados:

Creamos visualizaciones para entender y comunicar efectivamente los resultados del modelo KMeans. Utilizamos un gráfico de dispersión para representar la agrupación de datos en función de las características 'Asesinatos' y 'Asaltos', pudiéndose observar la relación entre estos (más asesinatos equivale a más asaltos), ver Figura 5. Se incluyeron los centroides de los clusters como puntos destacados en rojo para indicar los puntos centrales de cada grupo.

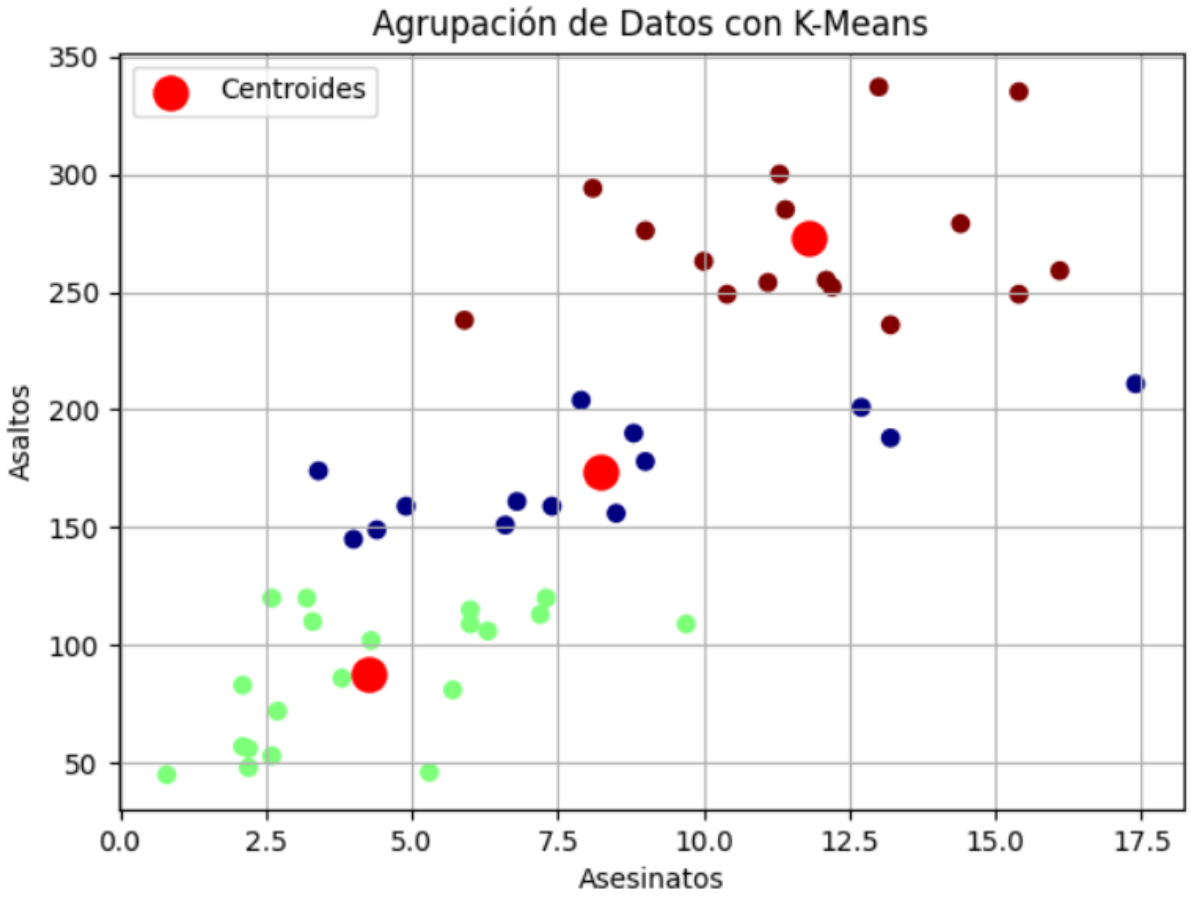

Figura 5: Gráfico de dispersión, representando asesinatos y asaltos.

3.2.5 Análisis Adicional con Pairplot:

Para una comprensión más profunda de la distribución de datos en los clusters, se generó un pairplot que muestra relaciones variadas entre las características. Cada punto se coloreó según su nivel de seguridad asignado para proporcionar una representación visual de la separación de clusters, ver Figura 6.

Este análisis visual es esencial para evaluar la eficacia del modelo en la agrupación de datos y proporciona información útil para las fases posteriores del proyecto.

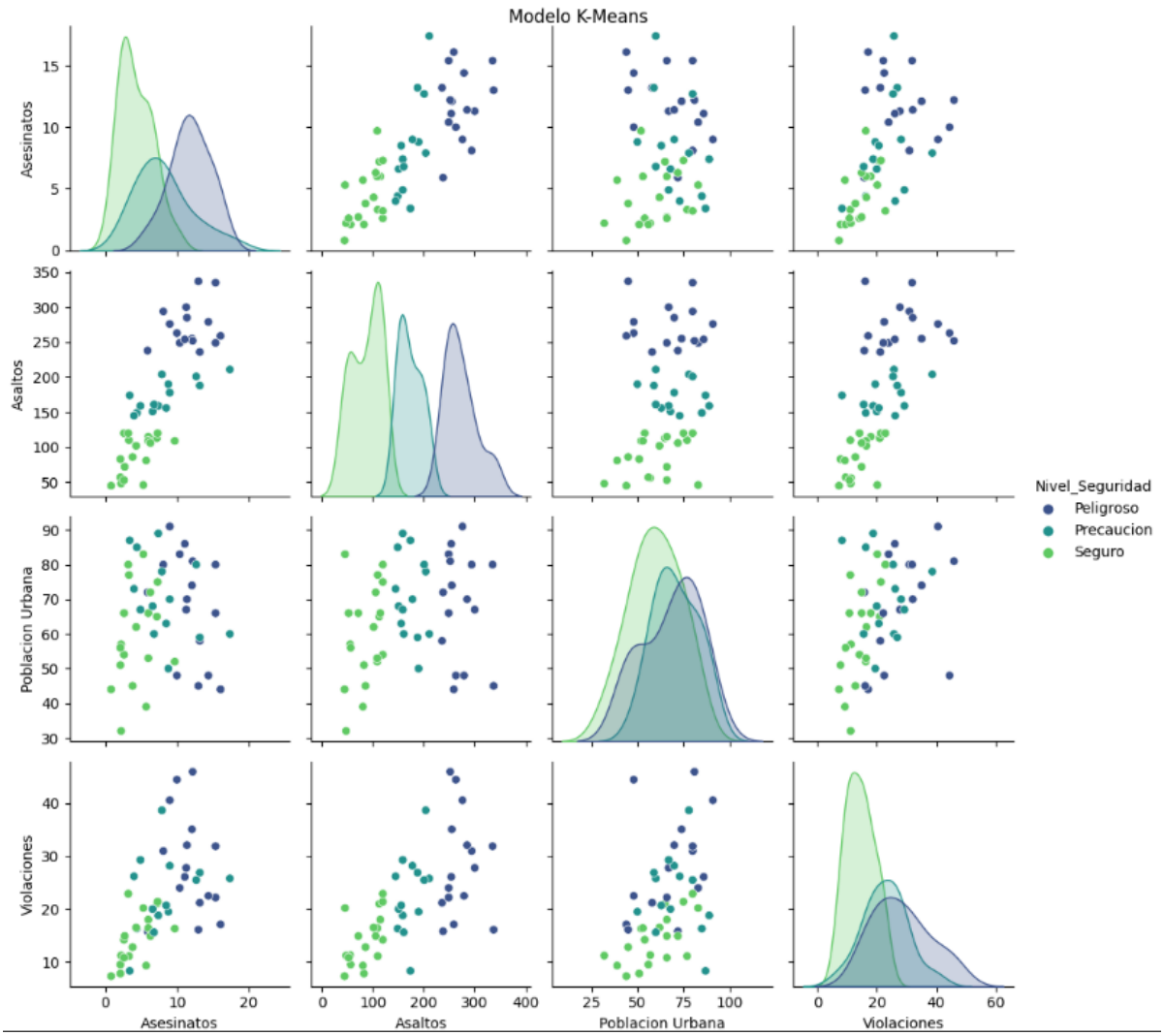

Figura 6: Gráfico con pairplot, relaciones varias entre variables modelo KMeans.

4. Creación del Modelo Mean Shift de SKLearn:

En esta fase del proyecto, nos enfocamos en la creación e implementación del modelo de agrupamiento Mean Shift utilizando la librería SKLearn. El objetivo principal fue explorar la estructura de los datos y identificar patrones naturales sin predefinir la cantidad de clusters.

4.1 Estimación del Ancho de Banda Óptimo:

Para aplicar el modelo Mean Shift de manera efectiva, es crucial determinar el ancho de banda (bandwidth) óptimo. Empleamos la función estimate_bandwidth de SKLearn, variando el parámetro de cuantiles y evaluando la puntuación de silueta para cada configuración. El gráfico resultante nos permitió identificar el ancho de banda adecuado para nuestro conjunto de datos ver Figura 7. En este caso, determinamos y comprobamos que un ancho de banda de 52 era óptimo para nuestro modelo.

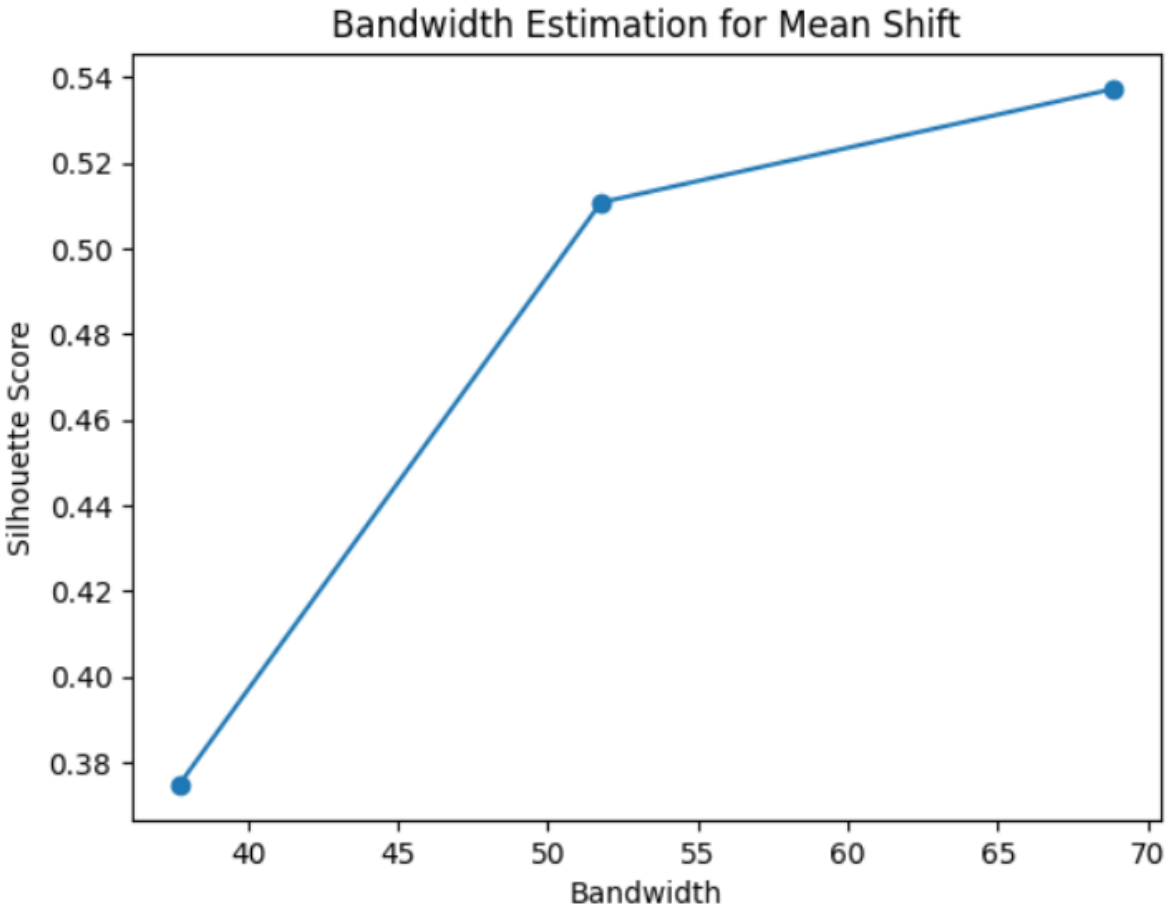

Figura 7: Resultados de la función ‘estimate_bandwidth_meanshift’.

4.2 Implementación del Modelo Mean Shift:

Instanciamos y entrenamos el modelo Mean Shift utilizando el ancho de banda óptimo identificado anteriormente. Visualizamos los resultados utilizando un gráfico de dispersión que representa la agrupación de datos en función de las características 'Asaltos' y 'Asesinatos'. Observándose que tenemos 3 grupos es decir tres clusters, ver Figura 8.

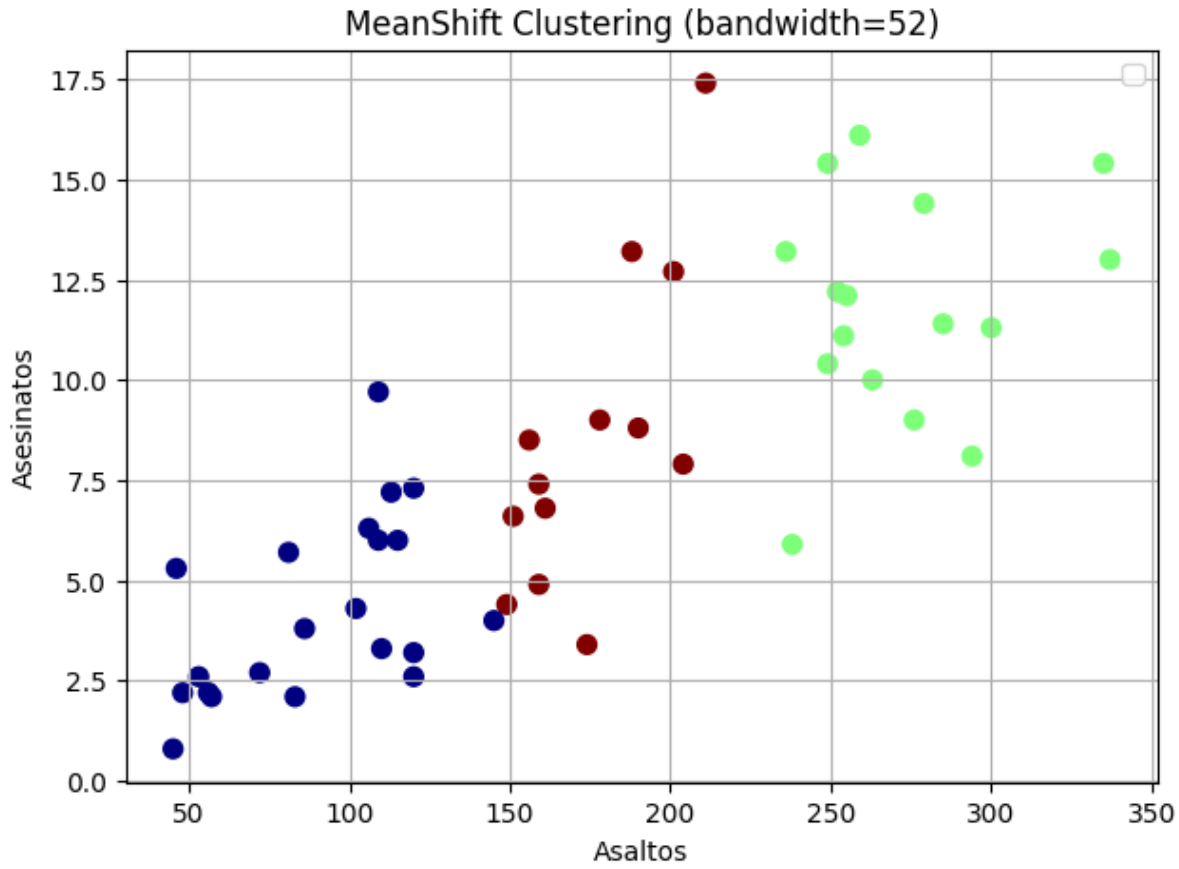

Figura 8: Gráfico de dispersión del modelo MeanShift.

4.3 Análisis Adicional con Pairplot:

Generamos un pairplot que muestra relaciones entre las características para comprender mejor la distribución de datos en los tres clusters identificados por el modelo Mean Shift. Este análisis visual es esencial para evaluar la eficacia del modelo y proporciona información útil para interpretar los resultados, ver Figura 9.

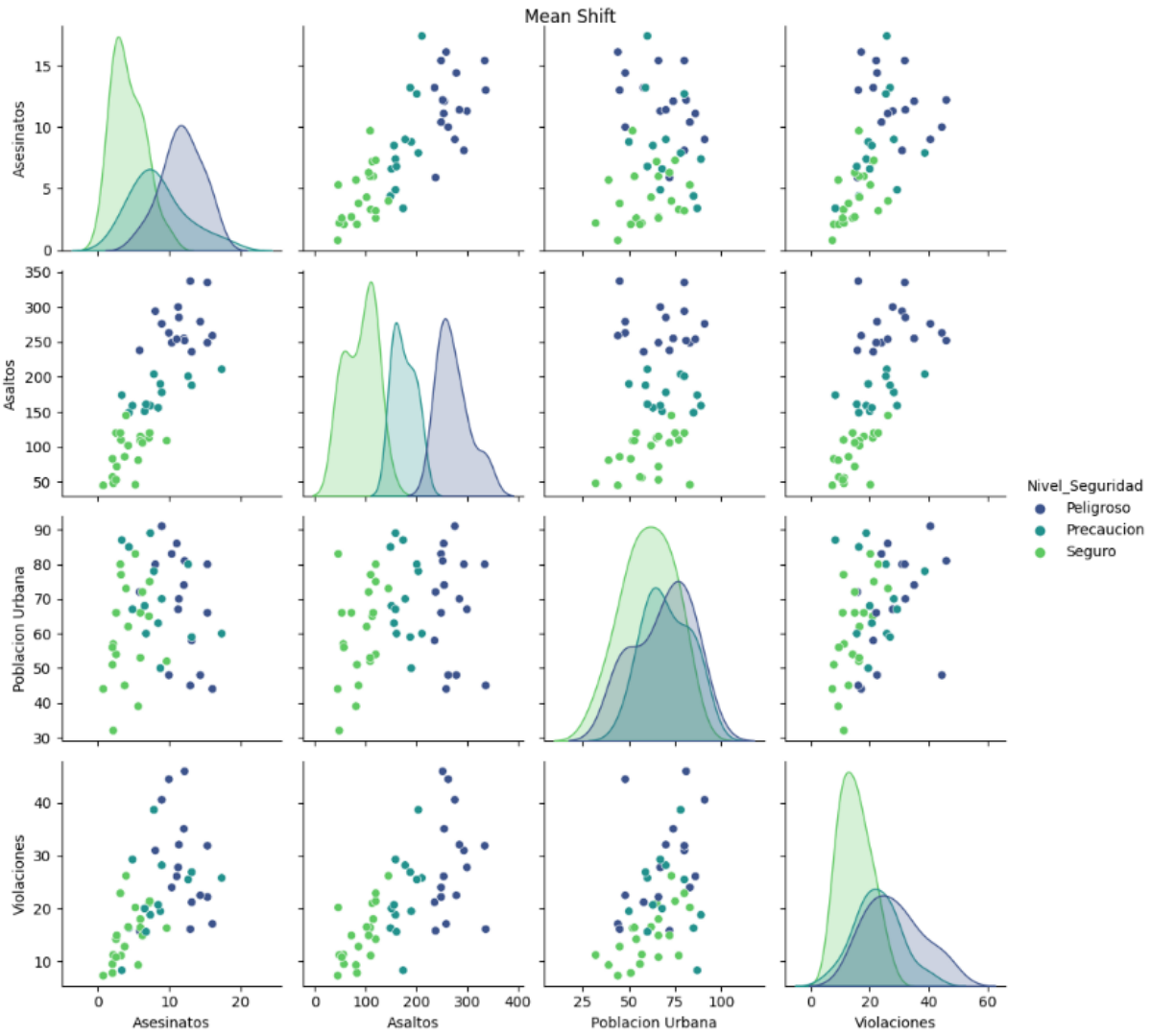

Figura 9: Gráfico pairplot, relaciones varias entre variables del modelo MeanShift.

5. Creación del Modelo DBSCAN de SKLearn:

En esta fase del proyecto, nos centramos en la implementación del modelo de agrupamiento DBSCAN (Density-Based Spatial Clustering of Applications with Noise) mediante la librería SKLearn.

5.1 Identificación del Parámetro Óptimo EPS:

Antes de aplicar DBSCAN, es esencial determinar el valor óptimo del parámetro EPS, que representa la distancia máxima entre dos muestras para ser consideradas en el mismo vecindario. Desarrollamos una función ‘find_optimal_eps_dbscan’ para encontrar este valor mediante la observación de las distancias al vecino más cercano. El gráfico resultante nos proporcionó información crucial para la configuración de DBSCAN, ver Figura 10.

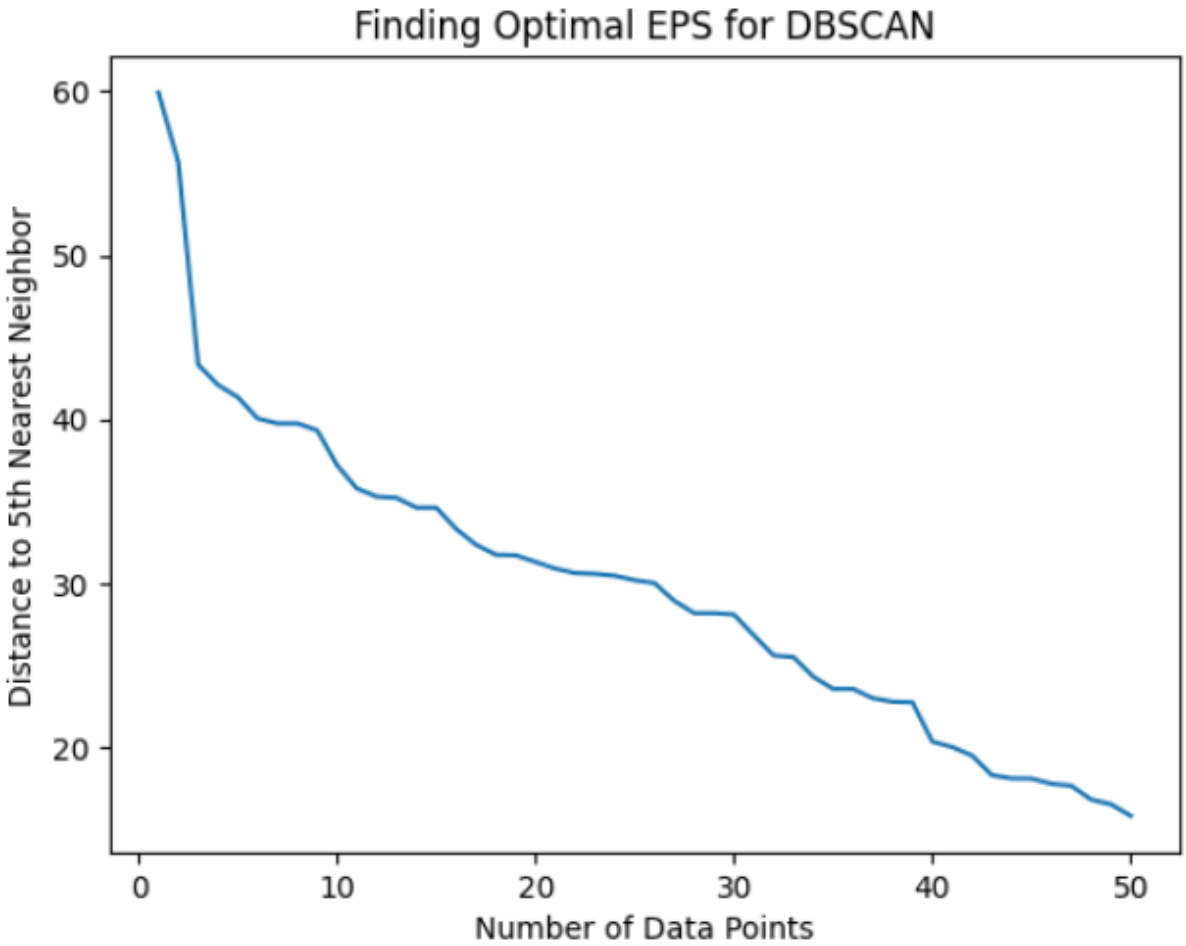

Figura 10: Gráfico lineal para determinar el EPS óptimo.

5.2 Aplicación del Modelo DBSCAN:

Ajustamos el modelo DBSCAN utilizando el valor de EPS identificado en la sección anterior. El resultado fue un conjunto de etiquetas de cluster, donde el valor -1 indica puntos considerados como ruido o atípicos. Visualizamos los resultados mediante un gráfico de dispersión que representa los datos en función de las características 'Asaltos' y 'Asesinatos', ver Figura 11, coloreando cada punto según el cluster asignado por el modelo.

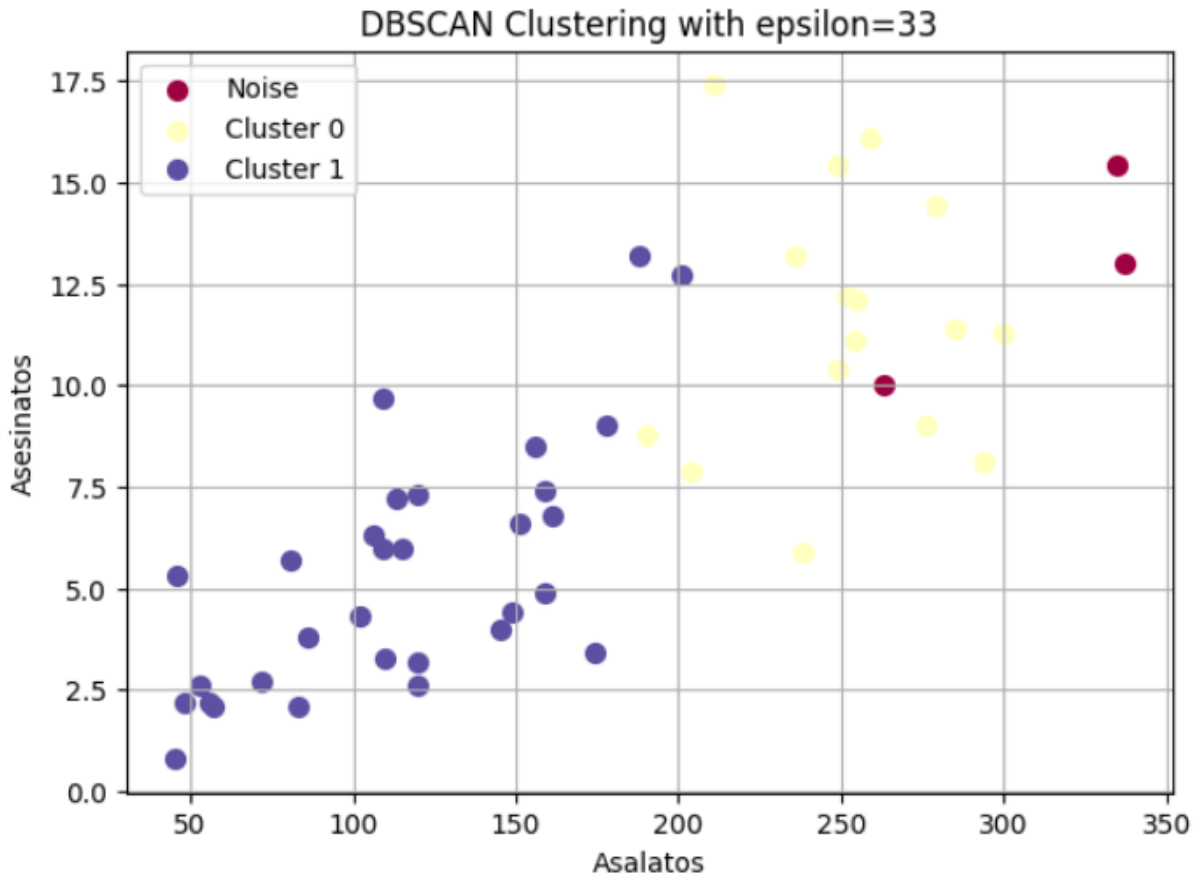

Figura 11: Gráfico de dispersión, asaltos y asesinatos del modelo DBSCAN.

5.3 Categorización y Análisis Adicional:

Categorizamos los clusters asignando etiquetas descriptivas, como 'Peligro' y 'Seguro', y clasificamos los puntos no asignados como 'Indefinido'. Luego, generamos un pairplot para visualizar las relaciones bivariadas entre las características, coloreadas según las categorías definidas. Este análisis reveló dos grupos claramente distinguibles que representan los niveles de seguridad 'Peligro' y 'Seguro'. Además, identificamos un pequeño número de datos indefinidos que no se asignaron claramente a ningún cluster, ver Figura 12.

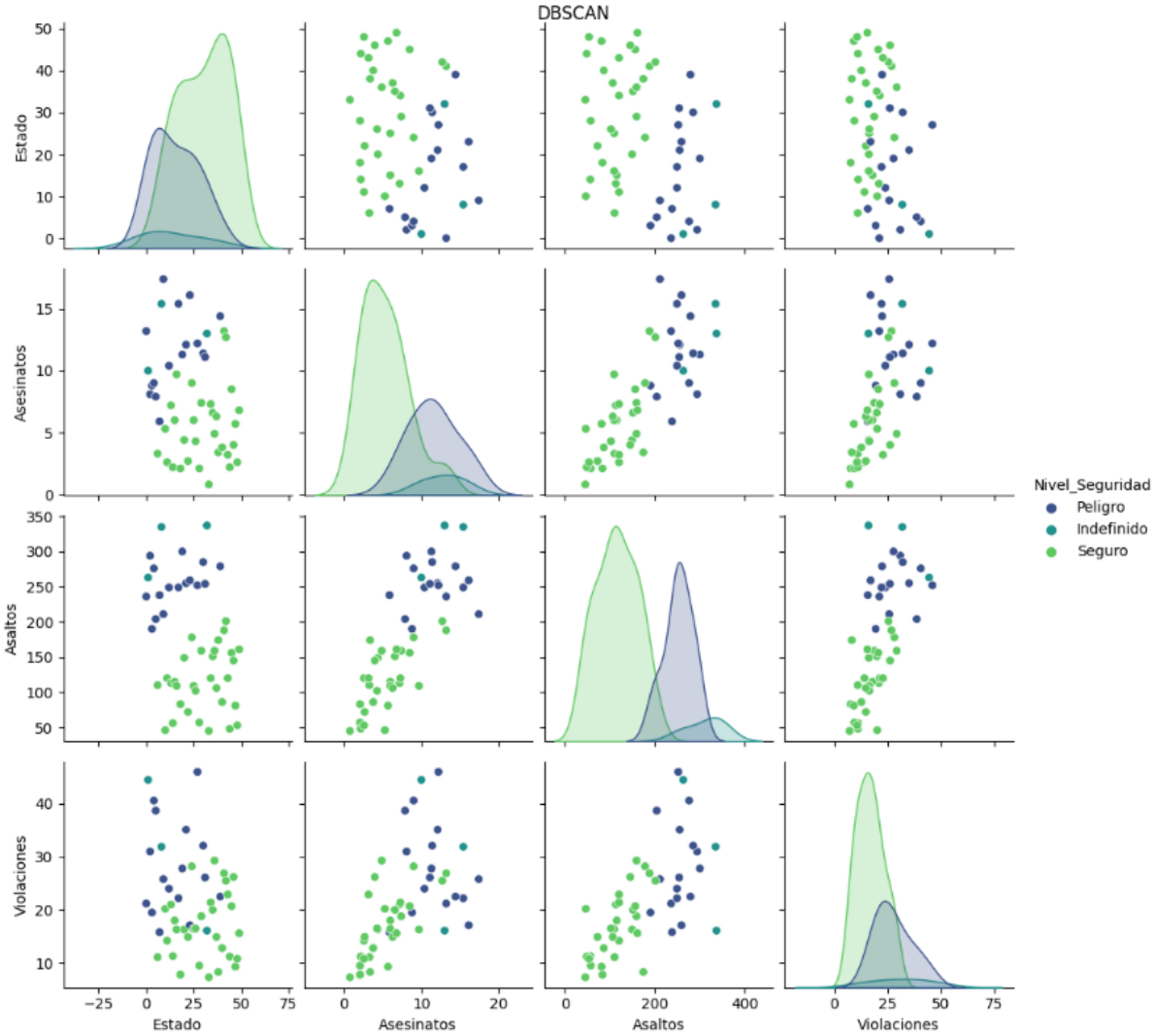

Figura 12: Gráfico pairplot, relaciones varias entre variables del modelo DBSCAN.

Estos resultados son fundamentales para comprender la capacidad de DBSCAN para capturar patrones en el conjunto de datos de crímenes y proporcionan información valiosa para la comparación con los modelos anteriores de KMeans y Mean Shift.

6. Conclusión:

En este proyecto de Clustering exploramos a fondo un conjunto de datos que abarca registros de crímenes y el porcentaje de población urbana en diversos estados de EE. UU. Nuestro objetivo principal fue clasificar los estados en diferentes categorías basadas en estos factores determinantes.

Iniciamos nuestro proyecto con la importación y preprocesamiento de datos, realizando una selección de características relevantes y la codificación para entrenar nuestros modelos. A través de visualizaciones detalladas utilizando gráficos pairplot, analizamos las relaciones entre las variables, pudiendo observar patrones y tendencias para nuestro análisis.

Posteriormente, realizamos la creación de tres modelos de clustering: KMeans, Mean Shift y DBSCAN. Aplicamos el Método del Codo para determinar el número óptimo de clusters en KMeans, logrando una clasificación efectiva de los estados en tres niveles de seguridad

‘Peligroso’, ‘Precaución’ y ‘Seguro’. Mean Shift, sin necesidad de predefinir clusters, reveló patrones naturales en los datos agrupando también en tres grupos como lo realizamos con KMeans, mientras que DBSCAN demostró su capacidad para identificar claramente dos grupos distintivos, 'Peligro' y 'Seguro', junto con algunos datos indefinidos.

En resumen, cada modelo nos ofreció perspectivas diferentes sobre la estructura de los datos. La elección del modelo depende de los objetivos específicos y la naturaleza del conjunto de datos. Este proyecto no solo brindó una clasificación efectiva de los niveles de seguridad de los dfierentes estados, sino que también proporcionó una comprensión de las relaciones y patrones en los datos de crímenes, observándose que, a mayores asaltos, mayores asesinatos, y a mayores violaciones también mayores asesinatos.