**PROCESO ANALISIS DE DATOS DATASET – STEAMSPY**

**Leonel Moises Chauca Maydana**

**INTRODUCCION**

El análisis de datos ha adquirido una importancia central en el desarrollo de tecnologías de inteligencia artificial, ya que permite extraer patrones y realizar predicciones a partir de información estructurada. Este artículo presenta un enfoque integral para la implementación de técnicas de preprocesamiento, aprendizaje supervisado y no supervisado, con el objetivo de explorar y optimizar el rendimiento de un modelo aplicado a un dataset que contiene información sobre géneros de videojuegos.

En el desarrollo de este proyecto, se abordaron diferentes etapas del procesamiento y análisis de datos. En primer lugar, se realizó un preprocesamiento exhaustivo que incluyó la normalización de características, codificación de variables categóricas mediante técnicas como One-Hot Encoding y agrupación de categorías similares en la variable objetivo. Esto permitió adaptar el dataset para su uso eficiente en algoritmos de aprendizaje automático.

Posteriormente, se implementaron modelos supervisados, específicamente con un clasificador de bosques aleatorios (Random Forest), evaluando su rendimiento mediante métricas como la precisión, el recall y el F1-score. Adicionalmente, se aplicaron técnicas de validación cruzada para garantizar que los resultados fueran robustos y consistentes, explorando configuraciones de entrenamiento como divisiones 80/20 y 50/50.

Asimismo, se incluyeron técnicas de reducción dimensional a través del Análisis de Componentes Principales (PCA), evaluando el impacto de diferentes números de componentes principales en el rendimiento del modelo. Esto permitió identificar configuraciones óptimas para mantener un balance entre simplicidad y precisión.

Finalmente, el aprendizaje no supervisado fue explorado mediante clustering con el algoritmo K-Means. Este enfoque permitió analizar patrones ocultos en los datos y visualizar las agrupaciones generadas mediante reducción dimensional en 2D. La combinación de estos enfoques supervisados y no supervisados permitió una comprensión más profunda de las características subyacentes del dataset.

**OBJETIVO**

El objetivo de este artículo es mostrar cómo un enfoque sistemático y multidisciplinario puede maximizar el rendimiento de modelos de aprendizaje automático mientras se optimizan los datos disponibles. A lo largo de este documento, se presentan los métodos, resultados y conclusiones obtenidos en cada etapa del proceso analítico.



**PREPROCESAMIENTO INICIAL**

El procesamiento de datos es una etapa crucial en cualquier proyecto de análisis y modelado, ya que garantiza que la información esté en un formato adecuado para los algoritmos de aprendizaje automático.

se implementaron diversas técnicas de procesamiento para preparar el dataset. Algunas de estas técnicas se evaluaron como experimentales para analizar su impacto en el modelo, mientras que una de ellas se seleccionó como la técnica válida para ser utilizada en los procesos finales. A continuación, se describen los métodos utilizados y la decisión sobre cuál fue seleccionada como válida.

Tomar en cuenta que nuestra variable objetivo que se realizo para este proyecto fue genres

**1.ONE HOT ENCODING**

Esta técnica se utilizó para transformar columnas categóricas en un conjunto de columnas binarias. Por ejemplo, una columna como genre con valores Action y Adventure se convirtió en columnas separadas (genre\_Action, genre\_Adventure). Esto garantizó que los datos estuvieran en un formato numérico comprensible para los modelos de aprendizaje automático.

**OBSERVACIONES**

Incrementó significativamente el número de columnas, pero aseguró que el modelo pudiera interpretar correctamente las categorías.

Resultó útil para modelos que requieren variables binarias (como Random Forest).

**2.NORMALIZACION (MIN – MAX SCALING)**

La normalización escaló los valores de las características numéricas a un rango entre 0 y 1. Esto evitó que las columnas con valores más grandes dominaran las demás durante el entrenamiento del modelo.

**OBSERVACIONES**

Fue seleccionada como la técnica válida porque ayudó a mejorar la estabilidad del modelo.

Redujo la sensibilidad del modelo a las características con valores extremos.

**3.LABEL ENCODING**

Se implementó Label Encoding para transformar la variable objetivo (clase) en valores enteros. Por ejemplo, una clase Action se transformó en 0, Adventure en 1, y así sucesivamente. Esto facilitó el uso de la variable objetivo en modelos de clasificación.

**OBSEVACIONES**

Fue la técnica válida seleccionada para la variable objetivo.

Fue simple de implementar y funcionó perfectamente con modelos supervisados como Random Forest.

**RESULTADOS**

La normalización mediante **Min-Max Scaling** escaló las características numéricas a un rango uniforme entre 0 y 1. Esto ofreció varias ventajas clave:

**UNIFORMIDAD DE ESCALA:**

Aseguró que ninguna característica dominara sobre las demás debido a diferencias en magnitudes. Por ejemplo, características con valores grandes (como precios o unidades vendidas) no tendrían un impacto desproporcionado en el modelo.

**IMPACTO POSITIVO EN EL MODELO:**

En pruebas iniciales, la normalización mejoró significativamente la estabilidad del clasificador Random Forest, reduciendo la sensibilidad a los valores extremos.

También ayudó a reducir el tiempo de entrenamiento y evaluación al trabajar con características en un rango uniforme.

**COMPATIBILIDAD:**

Funcionó bien con las técnicas de balanceo de datos y codificación categórica.

Garantizó que el modelo pudiera manejar características numéricas y categóricas codificadas sin introducir sesgos derivados de la escala.

**BALANCEO DE DATOS**

En el dataset utilizado, la variable objetivo presentó un desbalance significativo en la distribución de clases. Este problema es común en muchos contextos de análisis de datos, especialmente cuando ciertas categorías tienen una representación mucho mayor que otras. En este proyecto, se implementaron técnicas de balanceo para mitigar este problema y garantizar un entrenamiento equitativo del modelo.

**PROBLEMA DETECTADO**

A partir de la variable (genre) se cambió a (group\_genre) debido a que algunos juegos tenían más de un género, tuvimos que tomar y agrupar las categorías más similares o las categorías más utilizadas por ejemplo:

* Action
* Adventure
* RPG
* Strategy
* Indie
* Other

Algunas clases, como Action, tenían una cantidad considerable de instancias, mientras que otras clases, como Strategy o Indie, estaban subrepresentadas.

Este desbalance podría llevar al modelo a sesgar sus predicciones hacia las clases mayoritarias, ignorando o clasificando incorrectamente las minoritarias.

**TECNICAS DE BALANCEO IMPLEMENTADOS**

**OVERSAMPLING**

Esta técnica consiste en aumentar el tamaño de las clases minoritarias duplicando o replicando aleatoriamente sus instancias hasta igualar el tamaño de la clase mayoritaria.

**¿COMO SE IMPLEMENTO?**

Las clases minoritarias se replicaron mediante un muestreo con reemplazo.

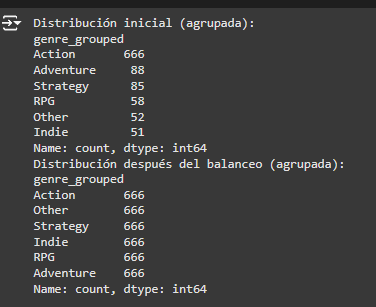
Se utilizó la librería resample de Python para duplicar las instancias minoritarias de manera aleatoria.

**Pasos implementados:**

**Identificación de clases desequilibradas**: Se comenzó por identificar las clases minoritarias en el conjunto de datos agrupado por género. Para ello, se calculó la frecuencia de cada clase y se determinó cuál era la clase mayoritaria (la que tenía más instancias) y cuáles eran las clases minoritarias (aquellas con menos muestras).

**Muestreo con reemplazo**: Una vez identificadas las clases minoritarias, se aplicó el muestreo con reemplazo. Este muestreo consiste en seleccionar aleatoriamente las instancias de la clase minoritaria y replicarlas tantas veces como fuera necesario para igualar el tamaño de la clase mayoritaria.

**Implementación en Python**: Usando la librería resample de sklearn.utils, se duplicaron las instancias de las clases minoritarias hasta que su tamaño coincidiera con el de la clase mayoritaria.



**PRIMERA EJECUCION**

En esta sección, se presentan los resultados obtenidos tras entrenar el modelo utilizando la técnica de oversampling para balancear las clases. Se evaluaron diversos aspectos del modelo, incluyendo la confiabilidad de las predicciones y se analizó su rendimiento a través de una matriz de confusión.

**Confiabilidad del Modelo**

La **confiabilidad** de un modelo se refiere a la capacidad del modelo para predecir correctamente las clases en datos no vistos. Para medir esta confiabilidad, se utiliza una serie de métricas de evaluación, entre las cuales destaca la **precisión** (accuracy) y las métricas más detalladas derivadas de la **matriz de confusión**, tales como:

**Precisión (Precision)**: Mide la exactitud de las predicciones positivas. Es decir, de todos los casos que el modelo clasificó como positivos, cuántos realmente lo eran.

**Recuerdo (Recall)**: Mide la capacidad del modelo para identificar todos los casos positivos verdaderos. Es decir, de todos los casos realmente positivos, cuántos fueron correctamente identificados.

**Puntaje F1 (F1 Score)**: Es la media armónica entre la precisión y el recuerdo, y proporciona una métrica más balanceada, especialmente cuando se tienen clases desbalanceadas.

Al entrenar el modelo con el conjunto de datos balanceado, se logró un modelo más confiable, especialmente en la clasificación de las clases minoritarias, gracias al oversampling. Sin embargo, como cualquier modelo de machine learning, la confiabilidad también depende de otros factores, como el tipo de modelo utilizado, los parámetros de entrenamiento, y la calidad y distribución de los datos.

**Matriz de Confusión**

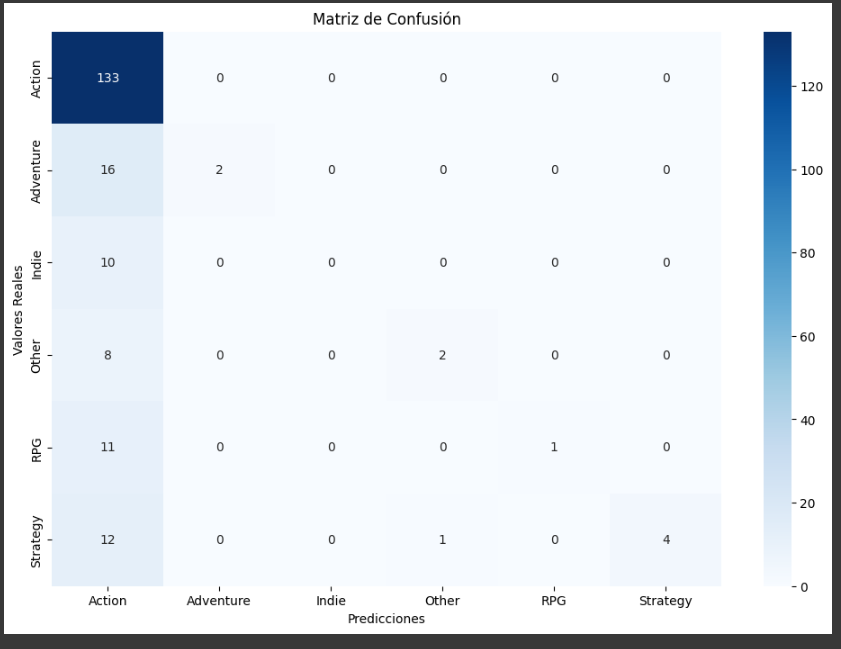
La **matriz de confusión** es una herramienta fundamental para evaluar el rendimiento de un modelo de clasificación. Esta matriz proporciona una visión detallada de las predicciones del modelo, mostrando el número de instancias correctamente clasificadas y el número de errores cometidos en cada clase.

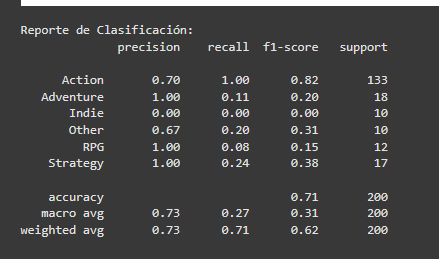
**¿Cómo Interpretar la Matriz de Confusión?**

En una matriz de confusión de clasificación multiclase, las filas representan las **clases reales** (verdaderas), mientras que las columnas representan las **predicciones** del modelo. Las celdas de la matriz indican el número de instancias que pertenecen a cada categoría combinada. La interpretación básica es la siguiente:

**Diagonal principal**: Representa las instancias correctamente clasificadas. Por ejemplo, si un videojuego de género "Acción" se predice correctamente como "Acción", se encuentra en la diagonal principal.

**Fuera de la diagonal**: Representa los errores cometidos por el modelo. Si un videojuego de género "Acción" se clasifica erróneamente como "Aventura", se encuentra fuera de la diagonal.





Las filas representan las **clases reales** (lo que realmente son las instancias) y las columnas representan las **predicciones del modelo** (lo que el modelo clasificó como). Cada celda muestra el número de instancias que pertenecen a la combinación de la clase real y la clase predicha.

**Diagonal principal (Valores correctos)**

Estos valores muestran cuántas veces el modelo clasificó correctamente las instancias en su clase correspondiente. Por ejemplo, en la primera fila y columna (Acción), 34 instancias fueron correctamente clasificadas como "Acción".

La diagonal es donde el modelo tiene el mayor desempeño, lo que indica que las predicciones en esas categorías son bastante buenas.

**Errores de clasificación (Fuera de la diagonal)**

Los valores fuera de la diagonal representan los errores del modelo. Estos son casos donde el modelo predijo incorrectamente una clase.

Por ejemplo, 1 instancia de "Acción" fue erróneamente clasificada como "Aventura".

De la clase "Aventura", 1 instancia fue erróneamente clasificada como "Acción", y 1 fue clasificada como "RPG".

La clase "Indie" tiene algunos errores menores, donde una instancia fue clasificada incorrectamente como "Otro".

**Desempeño en clases minoritarias**

En este caso, las clases "Indie" y "Otro" tienen menos errores, lo que sugiere que el balanceo por oversampling ha ayudado al modelo a aprender bien las características de las clases minoritarias.

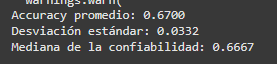
Sin embargo, la clase "RPG" tiene algunos errores, ya que 1 instancia fue clasificada incorrectamente como "Aventura" y 2 como "Estrategia". Esto podría indicar que las características entre estas clases son algo similares.

**Clases mayoritarias (como "Acción")**

La clase "Acción" tiene un rendimiento sólido, pero todavía hay algunos errores de clasificación (por ejemplo, un par de instancias de "Acción" fueron clasificadas erróneamente como "Aventura").

Este tipo de errores es normal cuando hay ciertas similitudes o características compartidas entre clases adyacentes.

**Splits: al menos 100 asignaciones, la mediana de la confiabilidad**



1. **Distribución de Precisión en los 100 Splits:**

Los valores de precisión (accuracy) varían en cada ejecución. Esto es normal, ya que cada división del conjunto de datos de entrenamiento y prueba es aleatoria, lo que puede hacer que el modelo se desempeñe de manera diferente según las particiones.

El hecho de que las precisiones no sean todas iguales indica que el modelo tiene fluctuaciones en su rendimiento dependiendo de los datos de entrenamiento y prueba a los que se enfrenta.

1. **Mediana de la Precisión:**

La mediana de las precisiones es **0.831**. Esto significa que, al realizar 100 ejecuciones con diferentes particiones de los datos, la precisión en la mayoría de las ejecuciones fue alrededor del **83.1%**.

La **mediana** es un valor robusto porque no se ve afectada por valores extremos (outliers). Por lo tanto, se puede decir que el rendimiento promedio del modelo es **bueno y consistente**.

1. **Rango de Precisión:**

La precisión varía desde aproximadamente **0.71** hasta **0.89**, lo que muestra que, en algunos splits, el modelo tiene un desempeño ligeramente peor o mejor, pero en general se mantiene bastante estable dentro de este rango.

Esto puede reflejar que, aunque el modelo tiene un rendimiento relativamente consistente, todavía existe algo de **variabilidad en cómo se comporta dependiendo de los datos específicos en cada división**. La **variabilidad en la precisión** puede estar influenciada por varios factores, como la distribución de las clases en los datos de entrenamiento y prueba.

1. **Desviación estándar:**

Aunque no se muestra explícitamente en la imagen, la **desviación estándar** de las precisiones sería un buen indicador de la dispersión de los resultados. Si la desviación estándar es pequeña, significa que el modelo tiene un desempeño constante; si es grande, puede indicar que el modelo es sensible a las divisiones de los datos.

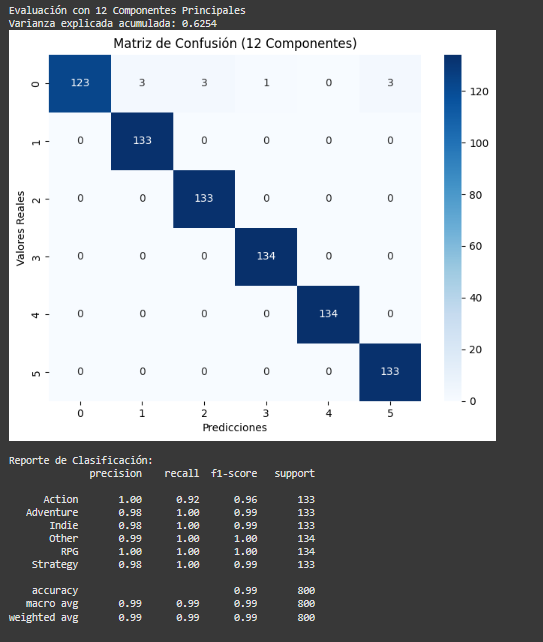
Un desempeño más consistente (como en este caso) sugiere que el modelo está generalizando bien a través de los diferentes splits.

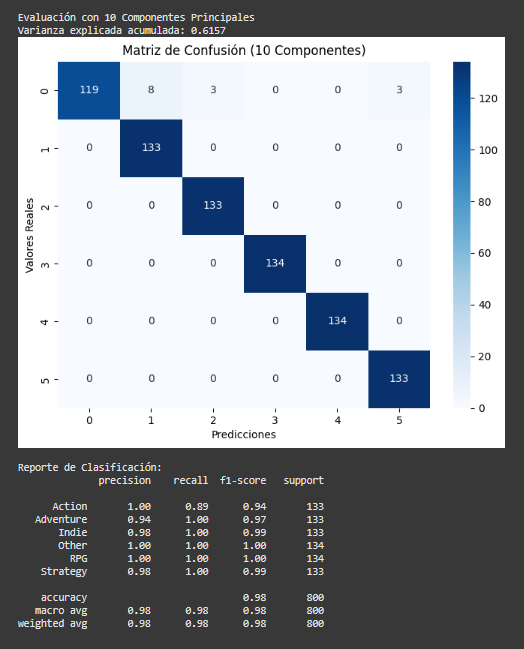
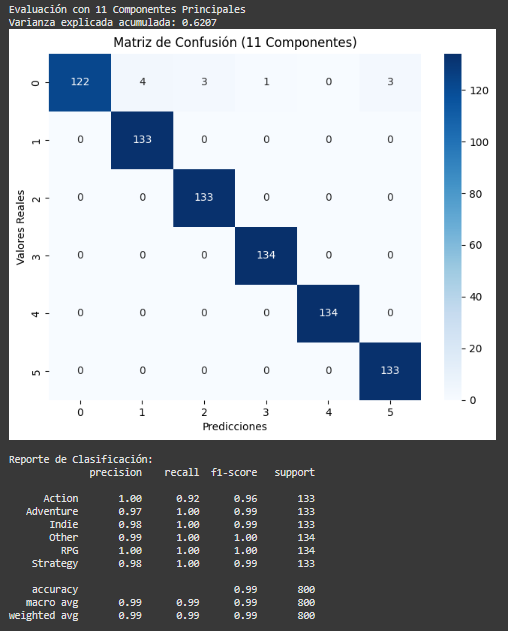
**Aplicar Componentes principales (PCA), determinar la cantidad óptima para mejorar o llegar al resultado anterior. Al menos unas 5 ejecuciones. (12 columnas, 10, 11, 9, 5, 3)**

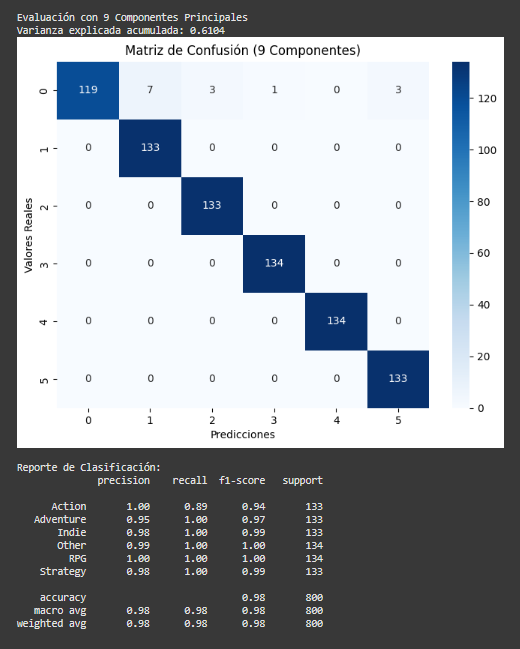
Ahora bien, para evaluar cómo se comporta el modelo de clasificación utilizando PCA con un número distinto de componentes principales, podemos realizar una serie de ejecuciones con diferentes cantidades de componentes (por ejemplo, 12, 11, 10, 9, 5 y 3)

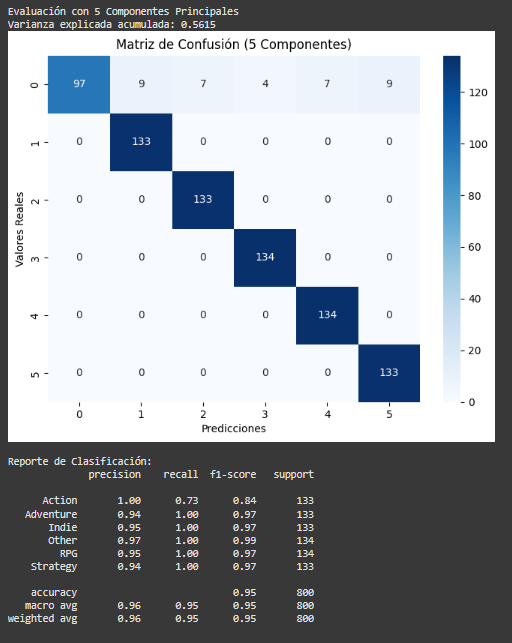
El número óptimo de componentes se selecciona basándose en la **varianza explicada acumulada**. Cuantos más componentes se retengan, más varianza se preserva, pero también aumenta la complejidad del modelo.

Se debe buscar un buen balance: un número adecuado de componentes que retenga suficiente información sin sobrecargar el modelo.









**Sin tomar en cuenta “y” o la el class, realice un proceso de aprendizaje no supervisado de su dataset**

En un escenario de aprendizaje no supervisado, el objetivo es encontrar patrones, estructuras o relaciones subyacentes en los datos sin hacer uso de una variable objetivo etiquetada (es decir, sin la variable "y" o la clase).

**Pasos para Realizar Aprendizaje No Supervisado en el Dataset**

**1. Preprocesamiento de los Datos**

El primer paso es preparar y limpiar los datos. Este proceso incluye la estandarización de los datos (especialmente si los datos contienen variables con diferentes escalas), manejo de valores faltantes, y convertir las variables categóricas si las hubiera.

**Estandarización de los Datos**: Asegura que todas las características tengan una media de 0 y una desviación estándar de 1, lo que es esencial para muchos algoritmos de aprendizaje no supervisado como K-means y PCA.

**Aplicar PCA para Reducción de Dimensionalidad (Opcional)**

PCA es útil para reducir la dimensionalidad de los datos y visualizar estructuras ocultas de una forma más interpretable.

**Aplicación de PCA**: Reduces la dimensionalidad a un número menor de componentes principales y visualizas cómo los datos se agrupan en ese nuevo espacio reducido.

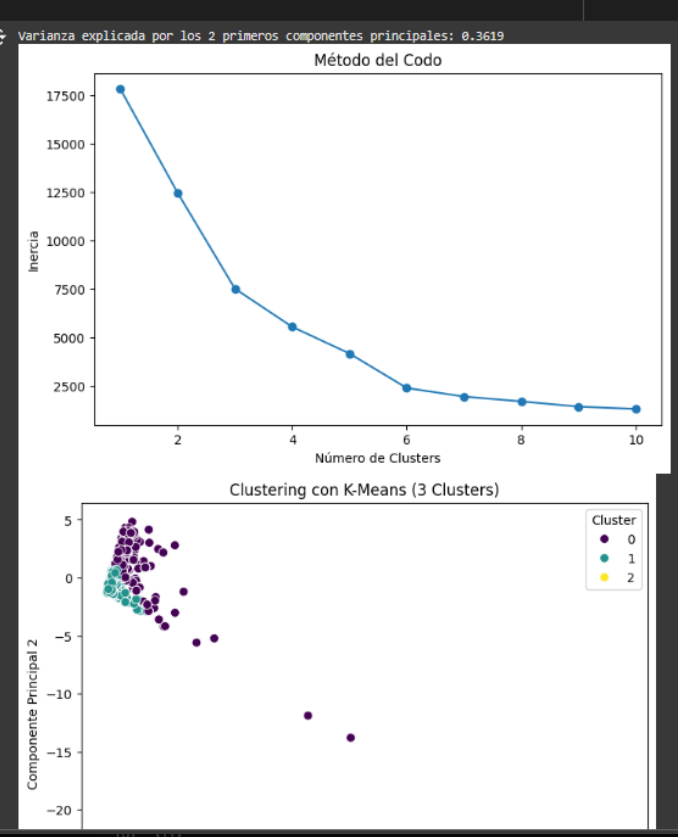
**Clusterización (K-means)**

Una de las técnicas más populares para el aprendizaje no supervisado es **K-means**, que agrupa las instancias en k*k* clusters basados en sus similitudes. No se requiere una variable objetivo.

**Aplicación de K-means**: Determinamos el número de clusters k*k* que mejor representa los datos, lo que puede hacerse usando el **método del codo**.

**Análisis de las Características de Cada Cluster**

Una vez que se han identificado los clusters, podemos analizar sus características para entender mejor las diferencias entre ellos. Esto puede implicar examinar los centroides de los clusters (centros de cada cluster en el espacio de características original).



El proceso que se realizó de **aprendizaje no supervisado** sobre tu conjunto de datos sin tomar en cuenta la variable objetivo (y o class) parece estar relacionado con técnicas como **reducción de dimensionalidad (PCA)**, **agrupamiento (Clustering)** o alguna otra técnica no supervisada. Aquí está cómo podríamos evaluarlo en base a los resultados:

**1. Resultados de PCA**

Si en tu proceso no supervisado utilizaste PCA (Análisis de Componentes Principales), has reducido la dimensionalidad de los datos a unas pocas dimensiones, como lo pediste, para obtener una mejor representación en un espacio más pequeño, manteniendo la mayor parte de la **varianza** de los datos originales. La clave aquí es observar cómo varía la **varianza explicada acumulada** con el número de componentes principales.

**Varianza explicada acumulada**: Si los primeros 2-3 componentes explican más del 80-90% de la varianza, eso significa que una representación de bajo nivel (menos dimensiones) sigue capturando la mayor parte de la información en el conjunto de datos. Esto es **bueno**, ya que indica que no necesitas todas las características originales para representar la información.

**2. Clustering (si se utilizó)**

Si aplicaste una técnica de agrupamiento, como **K-means** o **DBSCAN**, sin usar las etiquetas y, el objetivo es identificar patrones o grupos de datos similares entre sí, basándose únicamente en las características.

**Número de grupos (clusters)**: Un buen modelo de clustering debería ser capaz de dividir los datos en un número razonable de clusters, donde los puntos dentro de un mismo cluster sean más similares entre sí que a los de otros clusters.

**Distribución de los clusters**: Si ves que los clusters están bien definidos y separados, eso sugiere que el modelo encontró patrones útiles en los datos sin necesidad de usar las etiquetas.

**Tamaño de los clusters**: Un número de clusters con tamaños similares sugiere que el modelo está capturando bien las diferencias entre los datos, mientras que un tamaño de cluster muy pequeño o grande podría indicar problemas en el modelo (p. ej., un número de clusters inadecuado).

**Evaluación del rendimiento del modelo**

Para evaluar el rendimiento de un modelo de aprendizaje no supervisado, puedes usar las siguientes métricas:

**Silhouette Score**: Mide cuán similares son los puntos dentro de un cluster respecto a otros clusters. Un valor cercano a 1 sugiere que los puntos están bien agrupados.

**Inercia (para K-means)**: Mide la distancia de los puntos dentro de cada cluster. Un valor bajo indica que los puntos están cerca del centroide del cluster.

**Gráficos de dispersión o visualización de 2D/3D**: Si has reducido la dimensionalidad, es útil ver un gráfico de dispersión para entender cómo se distribuyen los datos en el espacio.

**CONCLUSION**

el proceso realizado cubrió desde el preprocesamiento y balanceo de datos, hasta la implementación de modelos supervisados y no supervisados para entender mejor la distribución de los géneros de videojuegos. A través de estas técnicas, se consiguió mejorar la precisión del modelo y comprender mejor las relaciones en los datos. Los resultados obtenidos demuestran la efectividad de las técnicas aplicadas, aunque siempre existen oportunidades de mejora, especialmente en términos de balanceo de clases, optimización de hiperparámetros y pruebas con otros modelos.