

Análisis de datos relacionados a la venta de videojuegos aplicando herramientas de Aprendizaje Automático.

Matías, Astore; Martin, Ceccotti; Simón, Risso; Mateo, Rivero

*UNRaf, Universidad Nacional de Rafaela, Santa Fe, Argentina.
Inteligencia Artificial y Aprendizaje Automático (AGRO / BIO)
Aprendizaje Automático y Grandes Datos (IC)*

Autor Corresp: teoorivero15@gmail.com

Resumen

El presente informe abarca el análisis detallado de un conjunto de datos de videojuegos con ventas superiores a 100,000 copias, obtenido de la página “Kaggle”, la cual proporciona diversos conjuntos de datos para aplicar cualquier técnica de análisis. Los datos fueron extraídos mediante la técnica de web scrapping del sitio “vgchartz.com” el cual contiene datos acumulados hasta 2016. El mismo cuenta con 16,598 filas y 11 columnas, aborda aspectos como ranking, nombre, plataforma, año, género, editor y ventas por región. La elección del conjunto se basa en su relevancia para la industria, permitiendo identificar tendencias, tomar decisiones informadas y explorar cambios a lo largo del tiempo y en diversas regiones.

El análisis inicia con un Análisis Exploratorio de Datos (EDA) detallado, utilizando librerías de Python tales como Pandas y Profiling, seguido del Data Wrangling para coherencia. Se aborda un análisis estadístico y visual para responder preguntas específicas y lograr una mayor comprensión del conjunto de datos. Se plantean y responden dos objetivos claves que involucran técnicas de agrupación y regresión. La agrupación utiliza algoritmos como k-Means para generar grupos basados en características. Para la regresión, se emplean algoritmos como Ridge, Regresión Lineal, etc. El trabajo concluye con la evaluación de algoritmos y predicciones específicas para futuros juegos, brindando una base sólida para decisiones en la industria de videojuegos.

Palabras claves: Videojuegos, Scrapping, EDA, Data Wrangling, Predicción, Regresión, Agrupación, Algoritmos, Machine Learning.

1. Introducción

A lo largo de este informe se detalla el trabajo realizado con el conjunto de datos de ventas de videojuegos [1]. La elección de este conjunto de datos se justifica con la gran importancia que tienen las ventas de videojuegos nuevos debido a la gran demanda de miles de juegos en distintos géneros. Este exhaustivo registro de videojuegos con ventas superiores a 100000 copias se posiciona como una fuente valiosa de información sobre la evolución del mercado. La exclusión deliberada de juegos móviles e independientes, así como de descargas gratuitas, garantiza un enfoque preciso en las ventas tangibles.

La relevancia de este conjunto de datos se evidencia en su capacidad para desvelar tendencias de ventas, géneros populares y plataformas exitosas en la industria de los videojuegos. Además, facilita la toma de decisiones comerciales y la exploración de cómo las preferencias de los jugadores influyen en las ventas a nivel mundial, desglosadas por regiones geográficas clave.

Como punto de partida, se llevó a cabo el Análisis Exploratorio de Datos (EDA) utilizando Pandas y Profiling. Este enfoque proporciona una comprensión detallada de la estructura y calidad de los datos, permitiendo identificar posibles desafíos y oportunidades. Posteriormente, el proceso de Data Wrangling aseguró la coherencia y limpieza de los datos, con ajustes específicos, como la eliminación de celdas faltantes y la corrección de tipos de datos incompatibles [2].

Además, se incorpora la perspectiva de investigaciones previas, como la realizada por Muhammed Ali Acikgoz, que trata sobre el análisis de datos en la industria de los videojuegos, destacando la creciente importancia de utilizar datos para comprender el comportamiento y las preferencias de los jugadores [3]. Su enfoque se centra en cómo dicho análisis puede proporcionar información valiosa para la participación de los jugadores, la identificación de tendencias y el diseño de estrategias de marketing específicas en el desarrollo y la comercialización de juegos. La investigación de Muhammed Ali Acikgoz proporciona una base sólida para comprender el panorama de la industria de videojuegos, centrándose en ventas, preferencias de género y la influencia de la cultura en las elecciones de las personas, lo cual se

puede observar gracias a los increíbles gráficos realizados por el autor.

Antes de sumergirse en los objetivos específicos de este informe, es crucial contextualizar la investigación. Si bien se realizó un análisis previo, el enfoque o proyecto se distingue al incorporar una metodología más avanzada y específica. La aplicación de algoritmos de regresión como Random Forest, Regresión Lineal, Regresión Ridge y SVR, junto con algoritmos de agrupación como DBSCAN y K-Means, representa una mejora sustancial en la capacidad predictiva y de segmentación en comparación con enfoques más convencionales. La mayoría de los estudios se centran en tendencias generales y preferencias de género, pero pocos han explorado a fondo la capacidad predictiva y de agrupación de características específicas de los videojuegos. Esto no solo añade profundidad a la comprensión del panorama de la industria de videojuegos, sino que también establece un estándar más alto para futuras investigaciones en este campo.

El presente informe va más allá al plantear nuevos objetivos y preguntas clave. A diferencia de investigaciones anteriores, se propusieron abordar dos objetivos específicos que se mencionan más adelante. Estos objetivos no solo contribuirán al avance de la investigación actual, sino que también revelarán nuevas perspectivas en el panorama de la industria de videojuegos, permitiendo una toma de decisiones más informada y estratégica.

2. Metodología

2.1 Análisis Exploratorio de Datos (EDA)

La fase inicial del estudio consta de un detallado Análisis Exploratorio de Datos (EDA), empleando una variedad de herramientas esenciales, entre ellas, Pandas, Profiling y Matplotlib. Esta fase es fundamental para comprender la complejidad y diversidad del conjunto de datos, sentando las bases necesarias para la implementación de algoritmos en las siguientes etapas del estudio.

El EDA abarca una serie de pasos cruciales, cada uno de los cuales se abordó minuciosamente para obtener una visión holística de los datos. A continuación, se detallan las actividades realizadas en cada etapa.

1. Preparación de datos

El primer paso fundamental en un Análisis Exploratorio de Datos (EDA) es asegurar que los datos sean accesibles y estén listos para cualquier técnica estadística. Dicho esto, el conjunto de datos se presentó en formato CSV (Valores Separados por Comas), lo cual facilitó su manipulación y procesamiento, utilizando las

herramientas recomendadas por la cátedra, mencionadas anteriormente.

Profiling

El Profiling desempeñó un papel crucial en esta etapa al proporcionar una visión panorámica del conjunto de datos. En este reporte se identificaron 563 celdas faltantes, se confirmaron la ausencia de filas duplicadas y clasificó las variables según su tipo. Este análisis exhaustivo permitió descubrir patrones, tendencias y desafíos potenciales, estableciendo una base sólida para abordar los objetivos de agrupación y predicción mediante algoritmos.

Data Wrangling

Tras este análisis inicial, se prosiguió en la fase de Data Wrangling, también conocido como la manipulación de datos, es un proceso esencial en la investigación. Este proceso implica la limpieza y unificación de conjuntos de datos complejos y desordenados, transformándolos en un formato más accesible y adecuado para análisis y modelado. Aquí se detallan las etapas clave del Data Wrangling implementado:

- **Descubrimiento:** Antes de sumergirse en cualquier análisis, se dedicó tiempo a comprender a fondo los datos, explorando su estructura, tipos y cantidad. También se buscó entender el propósito y la utilidad de los datos en el contexto de la investigación.
- **Estructuración:** La estandarización del formato de los datos fue una prioridad durante la fase de estructuración. Dada la diversidad de fuentes u orígenes posibles, se enfrentaron datos en diferentes formatos y estructuras. La estandarización garantiza coherencia y uniformidad, facilitando la posterior manipulación y análisis de los datos.
- **Limpieza:** La eliminación de datos redundantes y no informativos fue un paso crítico en la estrategia de limpieza. Se detectaron y eliminaron duplicados, así como celdas con información faltante, totalizando 563 eliminaciones de un conjunto de 16,598 filas. Además, se estandarizó el formato de las columnas, abordando incompatibilidades y asegurando coherencia en la presentación de datos.
- **Enriquecimiento (no aplicado):** Aunque el proceso de enriquecimiento estaba contemplado, no fue necesario en este caso. La riqueza inherente del conjunto de datos de videojuegos limitó la necesidad de agregar información adicional.
- **Validación:** La validación de datos fue una etapa crucial para garantizar la precisión y la integridad. Se aseguró de que los datos no se vieran afectados durante el proceso de manipulación y eliminación, buscando fiabilidad, credibilidad y calidad en los datos limpios. Este paso es fundamental, ya que los

datos se utilizarán para tomar decisiones informadas.

- **Publicación:** La etapa de publicación implica compartir los datos preparados para su uso. Estos datos limpios y estructurados son esenciales para realizar análisis exploratorios, entrenar modelos y tomar decisiones fundamentadas en el contexto de la investigación sobre videojuegos con ventas significativas. En este caso, no se publicaron los datos, sino que se guardó el nuevo conjunto de datos con las modificaciones realizadas.

En resumen, durante las diversas etapas de este proceso, se realizó lo siguiente:

- Se eliminaron 563 celdas faltantes distribuidas en todo el conjunto de datos, garantizando coherencia y limpieza.
- Se corrigieron los tipos de datos de las columnas 'Name' y 'Publisher' a texto y categórico respectivamente, abordando incompatibilidades identificadas en el informe.
- La columna 'Year', inicialmente de tipo 'float64', se ajustó al tipo de dato 'int32' para mayor coherencia.
- Se gestionaron las filas con valores "unknown", reduciéndose de 203 a 100 después de la eliminación de celdas faltantes. Dada su proporción insignificante, se optó por mantenerlas como "unknown", asegurando así la integridad del conjunto de datos.

Este proceso de limpieza y preparación no solo es un requisito previo para abordar preguntas específicas, sino que también establece las condiciones ideales para la correcta implementación de algoritmos en las fases posteriores del estudio. Al comprender y limpiar a fondo el conjunto de datos, se preparó para enfrentar con confianza los desafíos de agrupación y predicción, asegurando resultados sólidos y confiables en el análisis de videojuegos con ventas significativas.

II. Análisis estadístico

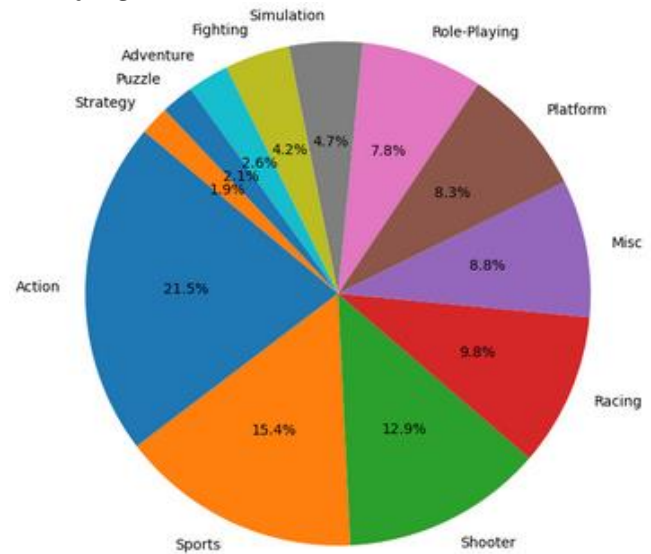
El segundo paso del Análisis Exploratorio de Datos (EDA) implica realizar un análisis estadístico, tanto gráfico como numérico, de las variables en el conjunto de datos. El objetivo es obtener una comprensión inicial de la información contenida y detectar posibles errores de codificación. En esta fase, se utilizaron herramientas como Matplotlib para la creación de gráficos, proporcionando así una representación visual de las respuestas a las preguntas planteadas.

Para abordar este paso, se plantearon una serie de preguntas claves, pero en el informe se

mostrarán las 5 más relevantes ya que las demás eran similares entre sí, también se utilizaron gráficos generados con Matplotlib para proporcionar respuestas basadas en variables específicas del conjunto de datos. Matplotlib es una biblioteca de visualización en Python elegida por su versatilidad y capacidad para crear gráficos claros y comprensibles.

A continuación, se presentan algunas de las preguntas planteadas:

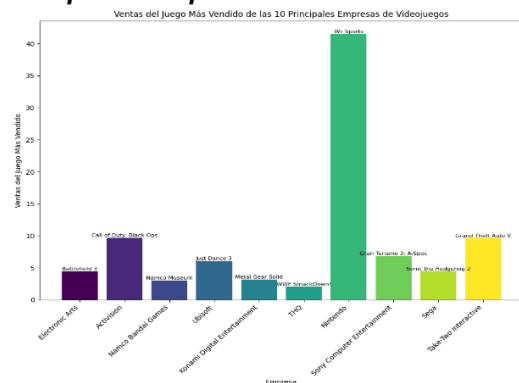
¿Cuál es el género más vendido en base a las ventas de una región según la empresa de videojuegos?



1. Distribución de ventas por género en EE.UU.

A partir del gráfico de pastel, se observa claramente que el género más vendido para la empresa Nintendo en Estados Unidos es el "Action" (Acción), seguido de cerca por el género "Sports" (Deportes). Estos dos géneros dominan significativamente las ventas, mientras que los demás géneros tienen una presencia mucho menor en comparación. Esta información es crucial para entender las preferencias de los jugadores en Estados Unidos en relación con los productos ofrecidos por Nintendo, lo que puede ser útil para futuras estrategias de marketing y desarrollo de juegos.

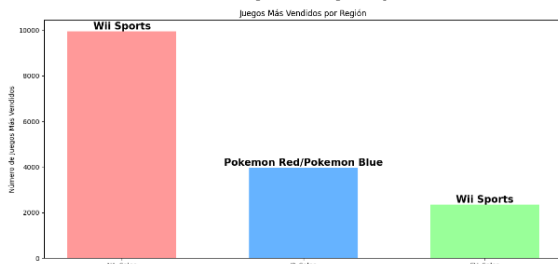
¿Cuál es el juego más vendido de cada una de las principales empresas?



2. Videojuegos más vendidos por las principales empresas.

A partir del gráfico de ranking realizado se identificaron los juegos más vendidos de las principales empresas de videojuegos hasta 2016. Para Nintendo, el juego más vendido fue 'Wii Sports', destacándose como líder en ventas para la empresa. En el caso de Take-Two Interactive, 'Grand Theft Auto V' se posicionó como el título más vendido, subrayando el impacto duradero de esta franquicia en la industria. Para Activision, 'Call of Duty: Black Ops' se destacó como el juego más vendido, mostrando la popularidad continua de la serie 'Call of Duty'. Este análisis revela los títulos emblemáticos que han contribuido significativamente a las ventas de estas empresas, ofreciendo una visión clara de los juegos que han capturado la atención y el interés de los jugadores a lo largo de los años.

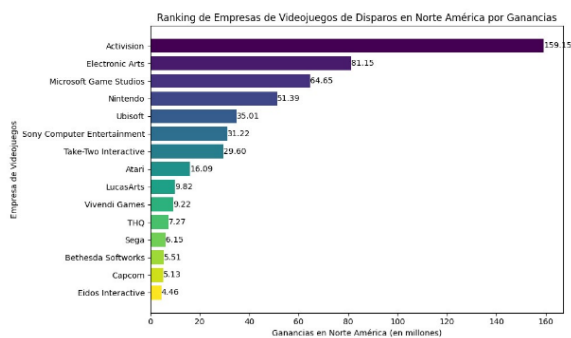
¿Cuáles son los juegos más vendidos en cada región geográfica (por ejemplo, Norteamérica, Europa, Japón)?



3. Videojuegos más vendidos por región.

Para la presente interrogación, se puede decir que la cantidad de ediciones que se vendieron para el juego Wii Sports está por encima de los demás videojuegos que se muestran en el gráfico. Dejando a la región de Norte América como una de las regiones donde más se pueden llegar a vender videojuegos y obtener buenas ganancias.

¿Cuál es la empresa que obtuvo la mayor cantidad de ganancias en la región de Norte América para videojuegos tipo "Shooter"?

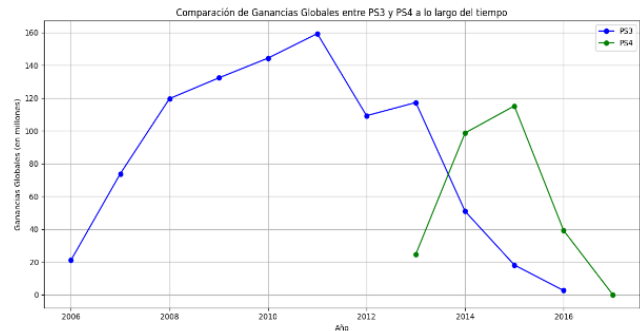


4. Principales empresas que obtienen grandes ganancias en videojuegos de tipo Shooter.

Para responder a esta pregunta, se analizó y se creó un gráfico de ranking que muestra las

empresas de videojuegos de disparos ordenadas por ganancias en Norte América. A partir del gráfico de ranking, se identificó que la empresa "Activision" fue la que obtuvo la mayor cantidad de ganancias en la región de Norte América para videojuegos tipo "Shooter".

Después del lanzamiento de la PS4, en base a las ganancias globales, ¿tuvo algún impacto negativo la plataforma PS3?



5. Impacto de PS4 sobre PS3.

Se realizó un análisis de cambio a lo largo del tiempo (Gráfico: Change v Time) que comparó las ganancias globales de las plataformas PS3 y PS4 después del lanzamiento de la PS4. A través del gráfico, se observó que las ganancias de la plataforma PS3 comenzaron a disminuir gradualmente después del lanzamiento de la PS4, lo que sugiere un posible impacto negativo en las ganancias globales de la PS3.

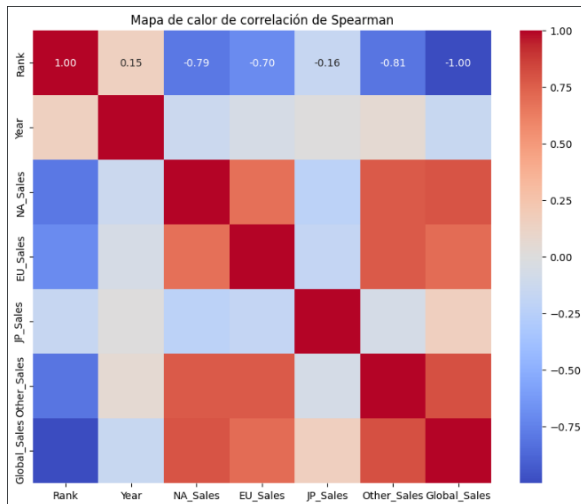
Este enfoque sistemático permitió explorar diversas dimensiones del conjunto de datos, revelando patrones, tendencias y relaciones clave que servirán como base para pasos posteriores del análisis. En base a este paso se realizaron una serie de conclusiones como por ejemplo que las preferencias de género varían según la región y la empresa de videojuegos. Por ejemplo, en Estados Unidos, Nintendo tiene éxito en géneros como "Action" y "Sports". Esto sugiere que las empresas deben adaptar sus estrategias de desarrollo y marketing según la región y el género de juego. También se puede decir que durante el periodo de 2000 a 2010, Electronic Arts, Nintendo y Activision fueron las empresas líderes en ganancias a nivel mundial. Esto indica que estas empresas tuvieron un impacto significativo en la industria durante esa década. Por otra parte, se agrega que los juegos más vendidos de cada empresa son títulos emblemáticos que han contribuido de manera significativa a las ventas, por ejemplo, "Wii Sports" para Nintendo y "Grand Theft Auto V" para Take-Two Interactive. Estos datos pueden orientar estrategias de desarrollo, marketing y toma de decisiones en la industria de los videojuegos.

III. Correlaciones independientes

La Etapa 3 del Análisis Exploratorio de Datos (EDA), centrada en Correlaciones y Dependencias, busca desentrañar las relaciones intrínsecas entre las variables numéricas presentes en el conjunto de datos sobre videojuegos con ventas significativas. Este proceso desempeña un papel crucial al proporcionar

percepciones detalladas sobre posibles patrones y asociaciones que puedan influir en las ventas globales de videojuegos.

En el análisis de correlaciones, se empleó la matriz de correlación de Spearman, una herramienta robusta que evalúa relaciones no lineales entre variables, revelando posibles dependencias ocultas. El enfoque se dirigió a variables numéricas clave, como el ranking de juegos, el año de lanzamiento y las ventas en diversas regiones. Con el propósito de ofrecer una representación visual más clara, se elaboró un gráfico de correlación, proporcionando una visualización gráfica de las relaciones numéricas. Este gráfico facilita la identificación de patrones y tendencias que podrían no ser evidentes al examinar sólo la matriz de correlación.



6. Correlaciones entre variables.

Matriz de correlación de Spearman:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.148835	-0.794823	-0.696730	-0.156597	0.811410	-1.000000
Year	0.148835	1.000000	-0.129967	-0.055209	0.007654	0.057738	-0.099627
NA_Sales	-0.794823	-0.129967	1.000000	0.682682	-0.224943	0.772111	-0.148554
EU_Sales	-0.696730	-0.055209	0.682682	1.000000	-0.174661	0.767558	0.794859
JP_Sales	-0.156597	0.007654	-0.224943	-0.174661	1.000000	-0.068726	0.696465
Other_Sales	0.811410	0.057738	0.772111	0.767558	-0.068726	1.000000	0.156717
Global_Sales	-1.000000	-0.099627	-0.148554	0.794859	0.696465	0.156717	1.000000

7. Matriz de correlación de Spearman

A partir de la gráfica y la matriz de correlación de Spearman se obtuvieron las siguientes conclusiones:

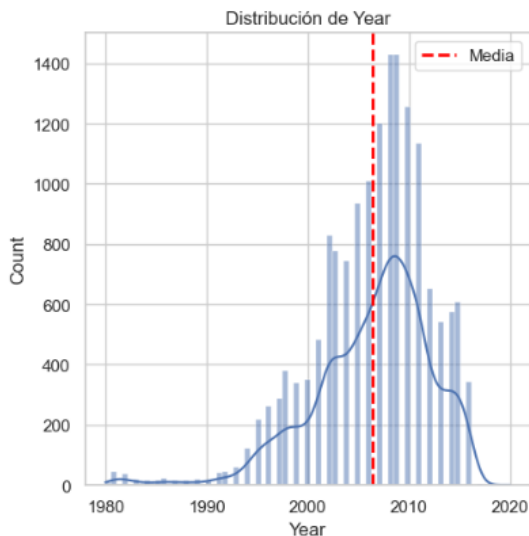
- **Rango del Juego y Ventas:** Se observa una fuerte correlación negativa entre el rango del juego y las ventas globales, indicando que juegos con rangos más bajos tienden a tener mayores ventas. Este hallazgo es coherente con la intuición, ya que los juegos populares tienden a posicionarse más alto en el ranking.

- **Año de Lanzamiento y Ventas:** La correlación negativa entre el año de lanzamiento y las ventas sugiere que los juegos más recientes tienden a tener un rendimiento comercial superior. Esto puede deberse a la evolución tecnológica, cambios en las preferencias de los jugadores y estrategias de marketing actualizadas.
- **Ventas por Región:** Existe una correlación positiva significativa entre las ventas en Norteamérica y Europa, indicando que los juegos que tienen un buen desempeño en una región también tienden a destacarse en la otra. Este patrón puede deberse a similitudes en preferencias culturales y de consumo.
- **Ventas en Japón:** La correlación positiva más débil con las ventas en Japón sugiere que el mercado japonés puede tener preferencias diferentes en comparación con Norteamérica y Europa. La relación menos pronunciada indica que el rendimiento en Japón puede depender de otros factores específicos.
- **Ventas en otras Regiones:** La correlación positiva entre las ventas en otras regiones y las ventas globales destaca la importancia de los mercados menos destacados individualmente. Las ventas en regiones más pequeñas contribuyen significativamente al rendimiento global.
- **Correlación entre Ventas Globales y Regionales:** La fuerte correlación positiva entre las ventas globales y las ventas en Norteamérica, Europa y otras regiones resalta la interconexión de estos mercados en términos de rendimiento comercial.

En conclusión, el análisis detallado de correlaciones proporcionó una visión profunda de las relaciones entre las variables numéricas clave en el conjunto de datos de videojuegos con ventas significativas. Se destacó la importancia del rango del juego, el año de lanzamiento y las ventas por región, revelando patrones valiosos que impactan en las ventas globales. Estos hallazgos contribuirán significativamente a la comprensión de los factores que influyen en el rendimiento comercial de los videojuegos, estableciendo así las bases para futuros análisis y toma de decisiones informadas.

IV. Distribución de las variables

La etapa 4 del Análisis Exploratorio de Datos (EDA) se centra en la distribución de simetría de las variables. La simetría en una distribución se refiere a la igualdad de las formas en ambos lados de su centro. A fin de lograr una mayor interpretación de las variables correspondientes, se procedió a realizar las gráficas de distribución de simetría de cada variable numérica. A continuación, se presenta a modo de ejemplo un gráfico de simetría de la variable "year".



8. Distribución de la característica Year.

Este gráfico de distribución proporciona una visión detallada de cómo están distribuidos los años en el conjunto de datos. Cada barra en el histograma representa un intervalo de años, y la altura de la barra indica la frecuencia o cantidad de registros en ese intervalo.

V. Valores perdidos

En el proceso de preparación de datos para el análisis, se detectaron valores faltantes en el conjunto de datos. Estos valores pueden surgir por diversas razones, como errores de entrada, problemas durante la recolección de datos o simplemente porque la información no estaba disponible en ese momento. Se implementó la eliminación de 563 celdas faltantes distribuidas en todo el conjunto de datos. Esta acción garantiza coherencia y limpieza en el análisis posterior, al eliminar aquellos puntos de datos que no contenían información completa. Este enfoque es crucial para asegurar que las conclusiones derivadas del análisis estén basadas en datos completos y confiables. Estas acciones se llevaron a cabo para asegurar que el conjunto de datos esté libre de valores faltantes o inconsistentes, sentando así una base sólida para los análisis subsiguientes y los algoritmos aplicados en el informe.

2.2 Objetivos y herramientas analíticas aplicadas

Una vez concluido por completo el EDA, después de analizar las relaciones entre variables y comprender la dinámica de la industria, se plantea el desafío de predecir las ventas futuras. La diversidad y la complejidad de los datos demandan algoritmos predictivos para anticipar el rendimiento de los videojuegos en función de atributos específicos, lo que brinda a las empresas una ventaja estratégica para la

toma de decisiones. Por lo tanto, se plantea el siguiente objetivo:

¿Cuáles serán las ventas globales de un videojuego específico en función de sus características, como el nombre, la plataforma, el año de lanzamiento, el género y el editor? Los algoritmos que se utilizaron para responder al mismo, fueron: ridge, regresión lineal, árboles aleatorios y SVR.

Por otro lado, el EDA reveló la existencia de patrones complejos y variaciones en las características de los videojuegos. La agrupación busca categorizar los juegos según similitudes en sus atributos, permitiendo una comprensión más profunda de las preferencias del mercado y segmentando de manera efectiva el vasto espectro de productos. Por lo tanto, se planteó el siguiente objetivo a responder:

A partir de un algoritmo de agrupación, se necesita generar n cantidad de grupos según las características del dataset para luego caracterizar cada grupo en base a sus atributos.

Los algoritmos que se utilizaron para responder al mismo, fueron: DBSCAN y K-Means.

I. Objetivo de regresión

Entrenamiento de los modelos

Para empezar, es importante destacar que se importan las librerías necesarias, incluyendo las librerías de los 4 algoritmos a aplicar, luego, se realiza la carga de los datos, es decir, se lee un conjunto de datos desde un archivo Excel llamado "df_copia_sin_nan.xlsx" y se almacena los datos en un DataFrame llamado "data". Asimismo, se identifican las columnas categóricas "Platform", "Genre" y "Publisher" que necesitan ser codificadas.

```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

# Leer el conjunto de datos
data = pd.read_excel('df_copia_sin_nan.xlsx')

# Variables categóricas a codificar
categorical_features = ['Platform', 'Genre', 'Publisher']
```

9. Importación de métodos y conjunto de datos.

Luego, se crea un objeto "OneHotEncoder" y se lo utiliza para transformar las variables categóricas en representaciones numéricas binarias. El resultado se almacena en "data_encoded". Después se convierten las variables categóricas codificadas en un DataFrame denominado "data_encoded_df" y consiguiente se combina este DataFrame con el conjunto de datos original "data_final".

Además, se seleccionan las características, es decir, se identifican las columnas resultantes después de la codificación One-Hot y las selecciona como características relevantes. Por último, la última línea de

código es la encargada de realizar la división de los datos, es decir, se divide los datos en conjuntos de entrenamiento “X_train” y “y_train” y prueba “X_test” y “y_test” utilizando la función “train_test_split” de scikit-learn.

```

# Variables de entrenamiento y prueba
categorical_features = ['Platform', 'Genre', 'Publisher']

# Crear un codificador One-hot
encoder = OneHotEncoder(sparse_output=False, handle_unknown='ignore')

# Ajustar y transformar las variables categóricas
data_encoded = encoder.fit_transform(data[categorical_features])

# Crear un DataFrame con las variables codificadas
data_encoded_df = pd.DataFrame(data_encoded, columns=encoder.get_feature_names_out(categorical_features))

# Combinar las características numéricas con las columnas transformadas One-hot
data_final = pd.concat([data.drop(columns=categorical_features), data_encoded_df], axis=1)

# Obtener todas las columnas después de aplicar One-hot Encoding
encoded_columns = [col for col in data_encoded_df.columns if col not in categorical_features]

# Seleccionar características relevantes
features = encoded_columns + lista las columnas transformadas One-hot

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(data_final[features], data_final[global_sales], test_size=0.2, random_state=0)

```

10. Transformación de variables categóricas a numéricas.

Aunque en el principio se importan varios algoritmos de regresión (Ridge, SVR, Linear Regresión Y RandomForestRegressor), este código en particular, se centra en la preparación de datos, específicamente en la codificación One-Hot de variables categóricas y la división de los datos en conjuntos de entrenamiento y prueba. La parte del código relacionada con los algoritmos se aplicaría a continuación en los pasos siguientes del análisis, después de haber preparado los datos de esta manera [4].

Evaluación y error (Regresión Ridge)

En este apartado, se aplica la codificación para realizar el algoritmo de regresión Ridge y se evalúa su rendimiento utilizando la métrica de error.

Para empezar, se realiza la inicialización y ajuste del modelo mencionado, para ello se crea una instancia del modelo de regresión Ridge especificando el parámetro “alpha” como 10. Dicho parámetro controla la fuerza de regularización en el modelo, que ayuda a prevenir el sobreajuste. Luego, el modelo se ajusta (entrena) utilizando los datos de entrenamiento “X_train” y “y_train”. Una vez que el modelo ha sido ajustado, se utilizan los datos de prueba “X_test” para realizar predicciones de las ventas globales “predictions_ridge”.

Por otra parte, se realiza como ya se dijo, el cálculo del error de Ridge, se utiliza la métrica de error “mean_squared_error” (error cuadrático medio) de scikit-learn para calcular la raíz del error cuadrático medio entre las predicciones y los valores reales de las ventas globales. La raíz cuadrada del error cuadrático medio (RMSE) se calcula colocando el parámetro “squared=False” en la función “mean_squared_error”, el resultado se imprime para evaluar el rendimiento del modelo.

```

# Inicializar y ajustar el modelo de regresión Ridge
ridge_model = Ridge(alpha=10)
ridge_model.fit(X_train, y_train)

# Realizar predicciones con Ridge
predictions_ridge = ridge_model.predict(X_test).ravel()

rmse_ridge = mean_squared_error(y_test, predictions_ridge, squared=False)
print("Error de Ridge:", rmse_ridge)

```

11. Evaluación y error de Regresión Ridge.

Evaluación y error (SVR)

Como en el caso anterior, se crea una instancia, pero en este caso para el modelo SVR especificando el tipo de kernel (en este caso “linear”) y otros parámetros como “C” que controla la penalización de error. Luego, el modelo SVR se ajusta (entrena) utilizando los datos de entrenamiento “X_train” y “y_train”. Después de ajustar el modelo, se utilizan los datos de prueba “X_test” para realizar predicciones de las ventas globales “predictions_svr”.

Por último, al igual que con el modelo Ridge, se utiliza la métrica de error “mean_squared_error” para calcular la raíz del error cuadrático medio entre las predicciones y los valores reales de las ventas globales.

```

# Inicializar y ajustar el modelo de Máquinas de Soporte Vectorial para la regresión (SVR)
svr_model = SVR(kernel='linear', C=1.0)
svr_model.fit(X_train, y_train)

# Realizar predicciones usando SVR
predictions_svr = svr_model.predict(X_test)
rmse_svr = mean_squared_error(y_test, predictions_svr, squared=False)
print("Error de SVR:", rmse_svr)

```

12. Evaluación y error de SVR.

Evaluación y error (Regresión Lineal)

Para dicho algoritmo, de nuevo, se inicializa una instancia del modelo mencionado utilizando la clase “LinearRegression()” y luego el modelo se entrena con los datos de entrenamiento. Después de ajustar el modelo, se utilizan los datos de prueba para realizar predicciones de las ventas globales “predictions_linear”.

Al igual que en los casos anteriores, se utiliza la métrica de error “mean_squared_error” para calcular la raíz del error cuadrático medio.

```

# Inicializar y ajustar el modelo de Regresión Lineal
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)

# Realizar predicciones usando regresión lineal
predictions_linear = linear_model.predict(X_test)
rmse_linear = mean_squared_error(y_test, predictions_linear, squared=False)
print("Error de Regresión Lineal:", rmse_linear)

```

13. Evaluación y error de Regresión Lineal.

Evaluación y error (Bosques Aleatorios)

Para dicho algoritmo, como en todos los casos, se crea una instancia del modelo de Random Forest para regresión utilizando la clase “RandomForestRegressor”, luego se especifica el número de estimadores (árboles) en el bosque mediante el parámetro “n_estimators” (en este caso, se usan 100 árboles) y también se fija la semilla aleatoria mediante “random_state=42” para garantizar reproducibilidad. Nuevamente, después de ajustar el modelo, se utilizan los datos de prueba para realizar predicciones de ventas globales.

```
# Inicializar y ajustar el modelo de Random Forest para regresión
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Realizar predicciones usando Bosques Aleatorios
predictions_rf = rf_model.predict(X_test)
rmse_rf = mean_squared_error(y_test, predictions_rf, squared=False)
print("Error de Random Forest:", rmse_rf)
```

14. Evaluación y error de Bosques Aleatorios.

Además, similar a los casos anteriores, se utiliza nuevamente la métrica de error "mean_squared_error" para calcular la raíz del error cuadrático medio entre las predicciones y los valores reales de las ventas globales.

Como se pudo observar, en esta sección del informe, se ha abordado la implementación de estos cuatro algoritmos de regresión. La elección de estos algoritmos se basó en la necesidad de abordar la complejidad del problema de predicción de ventas de videojuegos, donde las relaciones entre las características y la variable objetivo pueden ser no lineales y variadas. Por ejemplo, Random Forest es eficaz para modelar relaciones no lineales entre estas características, lo cual es crucial en problemas complejos para el conjuntos de datos elegido, además, la regresión Lineal y regresión Ridge proporciona modelos lineales más simples y fácilmente interpretables, lo cual es útil para comprender la importancia relativa de las características, por otra parte, la máquina de soporte vectorial para regresión (SVR) proporciona flexibilidad y puede adaptarse a relaciones no lineales mediante el uso de funciones kernel, esto le permite capturar patrones más complejos en los datos.

La elección de varios algoritmos ofrece un enfoque integral para abordar el problema de predicción. Diferentes algoritmos tienen fortalezas en diferentes situaciones, y utilizar una variedad como en este caso, permite evaluar y comparar su rendimiento en el conjunto de datos específico.

Por otra parte, cabe mencionar que la métrica de error utilizada para evaluar el rendimiento de los modelos fue la raíz del error cuadrático medio (RMSE). Se eligió RMSE debido a su capacidad para cuantificar la diferencia entre las predicciones y los valores reales de manera significativa y fácilmente interpretable, la elección de esta métrica se fundamenta en su capacidad para penalizar de manera proporcional los errores, proporcionando así una evaluación equitativa de los modelos. Aunque los cuatro algoritmos comparten la misma métrica de error (RMSE), los resultados varían significativamente entre ellos.

A fin de poder comparar el resultado de los errores obtenidos entre los diversos modelos y comprender mejor el rendimiento de los mismos, se realizó una gráfica por cada error para poder contemplar de forma más visual los mismos. El

código para esto fue el que se presenta a continuación.

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))

# Gráfico para Random Forest
plt.subplot(1, 2, 1)
plt.scatter(y_test, predictions_rf)
plt.plot([0, max(y_test)], [0, max(y_test)], 'k--', lw=2)
plt.title("Predicciones Random Forest vs Valores Reales")
plt.xlabel("Valores reales")
plt.ylabel("Predicciones Random Forest")
```

15. Generación de grafico de dispersión para Bosques Aleatorios.

```
plt.figure(figsize=(12, 6))

# Gráfico para SVR
plt.subplot(1, 2, 1)
plt.scatter(y_test, predictions_svr)
plt.plot([0, max(y_test)], [0, max(y_test)], 'k--', lw=2)
plt.title("Predicciones SVR vs Valores Reales")
plt.xlabel("Valores reales")
plt.ylabel("Predicciones SVR")

plt.tight_layout()
plt.show()
```

16. Generación de grafico de dispersión para SVR.

```
plt.figure(figsize=(12, 6))

# Gráfico para Regresión Lineal
plt.subplot(1, 2, 1)
plt.scatter(y_test, predictions_linear)
plt.plot([0, max(y_test)], [0, max(y_test)], 'k--', lw=2)
plt.title("Predicciones Regresión Lineal vs Valores Reales")
plt.xlabel("Valores reales")
plt.ylabel("Predicciones Regresión Lineal")
```

17. Generación de grafico de dispersión para Regresión Lineal.

```
# Gráfico para Ridge
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 2)
plt.scatter(y_test, predictions_ridge)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], 'k--', lw=2)
plt.title("Predicciones Ridge vs Valores Reales")
plt.xlabel("Valores reales")
plt.ylabel("Predicciones Ridge")

plt.tight_layout()
plt.show()
```

18. Generación de grafico de dispersión para Regresión Ridge.

Estos códigos generan gráficos de dispersión que comparan las predicciones de ventas globales de diferentes modelos de regresión con los valores reales. Cada conjunto de gráficos corresponde a un modelo específico (Random Forest, SVR, Regresión Lineal y Ridge) y sigue un formato similar. La línea punteada representa la línea de referencia donde las predicciones serían iguales a los valores reales.

En el primer conjunto de gráficos, se comparan las predicciones del modelo Random Forest con los valores reales. En el segundo conjunto, se hace lo mismo para el modelo SVR. En el tercer conjunto, se

visualizan las predicciones del modelo de Regresión Lineal, y finalmente, en el último conjunto, se presenta la comparación para el modelo Ridge.

Estos gráficos permiten una evaluación visual de la precisión de cada modelo al observar cómo se alinean las predicciones con los valores reales. La consistencia con la línea de referencia indica una mayor precisión en las predicciones. Este análisis visual es complementario a las métricas de error previamente calculadas y proporciona una comprensión más completa del rendimiento de cada modelo.

Generalización de nuevos datos

Una vez realizado el entrenamiento de los modelos y el cálculo de sus errores para comparar sus rendimientos, se prosiguió con la generalización de nuevos datos, es decir, la predicción de cuantas ventas globales tendrá ciertos nuevos juegos en torno a sus características. Para esto se realizó el siguiente código, en donde las predicciones fueron efectuadas por cada uno de los algoritmos para comparar los resultados.

```
# Crear un nuevo juego para predicción
new_game = pd.DataFrame({
    'Platform': [''],
    'Year': [],
    'Genre': [''],
    'Publisher': ['']
})

# Ajustar las características del nuevo juego al formato del entrenamiento
new_game_encoded = encoder.transform(new_game[categorical_features])

# Crear un DataFrame con las variables codificadas
new_game_encoded_df = pd.DataFrame(new_game_encoded, columns=encoder.get_feature_names_out(categorical_features))

# Combinar las características numéricas con las columnas transformadas One-Hot
new_game_final = pd.concat([new_game.drop(columns=categorical_features), new_game_encoded_df], axis=1)
```

19. Código utilizado para predecir las ventas globales de nuevos videojuegos.

Este código simula la predicción de ventas globales para un nuevo juego. Primero, se crea un DataFrame (new_game) con las características del nuevo juego. Luego, se utiliza el codificador One-Hot previamente ajustado (encoder) para transformar las características categóricas a numéricas y se crea un nuevo DataFrame con las variables codificadas (new_game_encoded_df). Finalmente, se combinan las características numéricas y las columnas codificadas para obtener un conjunto completo de características en el formato adecuado (new_game_final). Este conjunto se utilizará en los modelos entrenados para predecir las ventas globales del nuevo juego.

```
# Realizar la predicción con el modelo de Ridge
prediction_for_new_game = ridge_model.predict(new_game_final[features])

# Imprimir la predicción
print("Predicción de ventas globales para el nuevo juego:", prediction_for_new_game)

# Realizar la predicción con el modelo de SVR
prediction_for_new_game = svr_model.predict(new_game_final[features])

# Imprimir la predicción
print("Predicción de ventas globales para el nuevo juego (SVR):", prediction_for_new_game)

# Realizar la predicción con el modelo de Regresión Lineal
prediction_for_new_game = linear_model.predict(new_game_final[features])

# Imprimir la predicción
print("Predicción de ventas globales para el nuevo juego (Regresión Lineal):", prediction_for_new_game)

# Realizar la predicción con el modelo de Random Forest
prediction_for_new_game = rf_model.predict(new_game_final[features])

# Imprimir la predicción
print("Predicción de ventas globales para el nuevo juego (Random Forest):", prediction_for_new_game)
```

20. Código utilizado para predecir las ventas globales de un nuevo videojuego utilizando cada modelo entrenado.

Una vez preparado el conjunto con las nuevas características, se llevaron a cabo los siguientes códigos, los mismos realizan predicciones de ventas globales para un nuevo juego utilizando modelos entrenados con diferentes algoritmos de regresión. En resumen, se sigue un proceso similar para cada algoritmo:

- Ridge: Se utiliza el modelo de regresión Ridge previamente ajustado para realizar la predicción del nuevo juego.
- SVR (Support Vector Regression): Se utiliza el modelo de máquinas de soporte vectorial para regresión (SVR) para realizar la predicción del nuevo juego.
- Regresión Lineal: Se efectúa el modelo de regresión lineal previamente ajustado para realizar la predicción del nuevo juego.
- Random Forest: Se emplea el modelo de Random Forest para regresión previamente ajustado para realizar la predicción del nuevo juego.

En cada caso, se imprime la predicción de ventas globales para el nuevo juego. Aunque los códigos son similares, cada algoritmo tiene su propia lógica interna y consideraciones, lo que lleva a resultados ligeramente diferentes. Estas predicciones permiten estimar las ventas globales de un nuevo juego según las características proporcionadas.

```
# Crear un gráfico de barras con colores vibrantes
plt.figure(figsize=(10, 6))
colors = ['skyblue', 'lightgreen', 'lightcoral', 'gold']
plt.bar(modelos, predictions_for_new_game, color=colors)
plt.title('Predicciones de Ventas Globales para el Nuevo Juego')
plt.xlabel('Modelo')
plt.ylabel('Ventas Globales Predichas')

# Añadir un fondo de cuadrícula
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Mostrar el gráfico
plt.show()
```

21. Estimación de las ventas globales de un nuevo videojuego según las características proporcionadas.

Una vez obtenidas las predicciones de ventas globales de un nuevo juego generado por cuatro modelos de regresión: Ridge, SVR, Regresión Lineal y Random Forest, utilizando la biblioteca Matplotlib, se crea un gráfico de barras para visualizar de mejor

manera y comparar. El tamaño y los colores del gráfico se configuran para una presentación atractiva. Cada barra representa las ventas predichas por un modelo específico. El título del gráfico y las etiquetas en los ejes x e y proporcionan contexto, mientras que la cuadrícula de fondo facilita la interpretación. Al ejecutar el código, se obtiene un gráfico que permite comparar de manera clara las predicciones de cada modelo para el nuevo juego.

Dichos resultados de los distintos modelos (sus errores y predicciones) se presentarán a continuación en el apartado denominado "resultados" con sus respectivas gráficas.

II. Objetivo de agrupación

Para la resolución de este objetivo se utilizaron dos algoritmos de clustering: K-Means y DBSCAN [5]. Para ambos se realizó el mismo preprocesamiento, el cual se presenta a continuación:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df = pd.read_excel('df_copia_sin_nan.xlsx')
dataset = df[['Genre', 'NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
```

22. Importación del conjunto de datos y librerías.

Se importan pandas, matplotlib y numpy para el manejo y la representación de los datos. Luego, se lee el conjunto de datos y se almacenan en "df". Resulta importante resaltar que, de la totalidad del dataset, se seleccionan las columnas "Genre", "NA_Sales", "EU_Sales", "JP_Sales" y "Other_Sale", esto se debe a que el análisis se enfocó en encontrar patrones y relaciones entre los géneros de los videojuegos y las distintas regiones con el fin de identificar las preferencias de los jugadores según estas características.

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler

# Variables categoricas a codificar
categorical_features = ['Genre']

label_encoder = LabelEncoder()
dataset['Genre_Label'] = label_encoder.fit_transform(dataset['Genre'])
features = dataset[['Genre_Label', 'NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]

scaler = StandardScaler()
scaled_df = scaler.fit_transform(features)
```

23. Utilización de métodos para transformar datos categóricos a numéricos.

Luego, se importan "LabelEncoder" y "StandardScaler" del módulo de preprocesamiento de scikit-learn. En segundo lugar, se almacena la columna "Genre" en la variable "categorical_features" con el fin de transformar los datos categóricos en numéricos. Luego, se crea un objeto "LabelEncoder()" y se aplica el método ".fit_transform" para realizar la conversión. Finalmente, se rearma el dataset pero esta vez con la columna "Genre_Label" la cual contiene los datos transformados.

Para finalizar, se escalan los datos creando un objeto "StandardScaler()" y aplicando el método ".fit_transform".

K-Means

Este algoritmo de agrupación se basa en el hiperparámetro "k", el cual determina el número de clusters. El algoritmo trabaja de la siguiente manera: primero se seleccionan de manera aleatoria k puntos en el espacio de características los cuales se toman inicialmente como los centros de los clústeres (centroides). Luego, se calculan las distancias de los puntos a cada centroide con una medida específica y se asignan los puntos al clúster más cercano. El siguiente paso es reajustar los centros de los grupos en función de los valores promedios de los puntos en cada clúster. Este proceso se repite hasta que la variación de los centroides sea pequeña.

```
from sklearn.cluster import KMeans

km = KMeans()
km.fit(scaled_df)

k_values = list(range(2, 20))
inertias = []
for k in k_values:
    km = KMeans(n_clusters=k)
    km.fit(scaled_df)
    inertias.append(km.inertia_)
```

24. Importación de KMeans.

En un principio, se importa K-Means desde scikit-learn y, con el fin de optimizar la cantidad de clúster y determinar cuál es el mejor valor de k para nuestro modelo, se generó un gráfico de "codo" a través del siguiente código.

```
import matplotlib.pyplot as plt

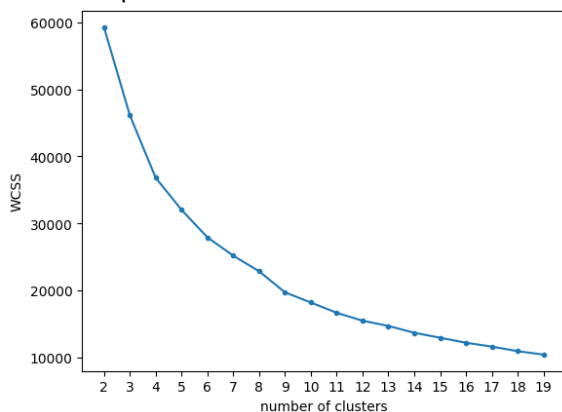
plt.plot(k_values, inertias, marker='.')
plt.xticks(k_values)
plt.xlabel('number of clusters')
plt.ylabel('WCSS')
```

25. Generación del gráfico de "codo".

En este caso, se hace uso de una de las principales métricas de clustering denominada "Suma de cuadrados dentro del grupo" por sus siglas en inglés WCSS (Within cluster sum of squares) o más conocida como "Inercia". En general, las métricas utilizadas en algoritmos de agrupación apuntan a medir que tan bien están agrupados los clústeres. Para el caso específico de la inercia, es un método relativamente simple que calcula la suma de las distancias desde cada punto al centro del grupo.

Otra de las principales métricas de clúster se llama "Coeficiente de Silueta". Este método mide una relación que involucra la distancia promedio entre un solo punto y todos los puntos en el mismo grupo y, por otro lado, la distancia promedio entre un solo punto y todos los puntos en el siguiente grupo más cercano. La puntuación del coeficiente de silueta puede variar de -1 a +1, si obtenemos un valor que se aproxima a 1 significa que los clústeres se encuentran bien formados, si obtenemos un número que se aproxima a 0 quiere decir que los clústeres se encuentran superpuestos y, por último, si se obtiene un resultado que se aproxima a -1, la asignación de los clústeres es errónea.

El gráfico de codo obtenido para la métrica inercia se presenta a continuación.



26. Grafica de "codo" generada.

En un gráfico de "codo", idealmente se puede ver un punto donde la tasa de mejora del WCSS u otra métrica se ralentiza y la línea se aplanan rápidamente. En el caso anterior, no hay un codo muy claro, lo que sugiere una dispersión más continua de los datos.

DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) es un algoritmo de agrupación completamente diferente a K-Means. Con DBSCAN, los grupos se componen de puntos centrales y puntos no centrales. Todos los puntos centrales están dentro de una distancia épsilon (eps en los parámetros de scikit-learn), de al menos n puntos en el mismo grupo (n es el parámetro min_samples en la función de). Entonces, cualquier otro punto dentro de la distancia épsilon de los puntos centrales también está en el grupo. Si algún punto no está dentro de la distancia épsilon de algún punto central, estos son valores atípicos.

```
from sklearn.cluster import DBSCAN
db = DBSCAN(eps=0.4, min_samples=10, n_jobs=-1).fit(scaled_df)
```

27. Inicialización de DBSCAN.

El código anterior muestra cómo se importa el algoritmo desde scikit-learn y se entrena el modelo con los siguientes valores de hiperparámetros: eps=0.4 y min_samples=10. El hiperparámetro min_samples generalmente debe estar entre la cantidad de características y el doble de esta cantidad, con un valor más alto para datos más ruidosos. Y, en cuanto a eps, se entrenó con una variedad de valores de este hiperparámetro y se determinó cuál era el más adecuado.

Luego, con el código que se muestra a continuación, se imprimieron la cantidad de clústeres y el número estimado de puntos de ruido, dando como resultado 4 y 1191, respectivamente.

```
labels = db.labels_

n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)
```

28. Cantidad de clústeres y puntos de ruido.

Finalmente, con el objetivo de poder comparar los clústeres obtenidos y debido a que la cantidad de características imposibilita la posibilidad de graficar los resultados, lo que se hizo con el código que se presenta a continuación fue imprimir los valores medios de cada característica para cada clúster.

Primero se crea una copia de nuestro DataFrame original y se agregan las etiquetas al mismo. Luego se recorre cada etiqueta de grupo e imprime el valor promedio de los puntos en ese grupo. También se agregó una nueva línea ("\n") para que cada grupo de resultados esté separado por una línea en blanco.

```
df_labels = dataset.copy()
df_labels['label'] = db.labels_
for label in range(n_clusters_):
    print(f'cluster {label}:')
    print(df_labels[df_labels['label'] == label].mean(), '\n')
```

29. Código encargado de mostrar en pantalla los valores medios de cada característica para cada clúster.

3. Resultados

A continuación, en esta sección se exhiben los resultados tanto del objetivo de predicción como el de agrupación. Aquí se presentan los resultados de los errores y de los distintos modelos, junto con las visualizaciones necesarias para su correcto entendimiento.

3.1 Resultados de regresión

I. Resultados de errores

En esta sección, se presentan los resultados de los errores obtenidos mediante la aplicación de los diferentes algoritmos de regresión. Estos errores, medidos en términos de la Raíz del Error Cuadrático Medio (RMSE), proporcionan una evaluación cuantitativa de la precisión de las predicciones realizadas por cada algoritmo. La interpretación de estos resultados es crucial para determinar la eficacia y la idoneidad de cada modelo predictivo en el contexto

específico de la industria de videojuegos. A continuación, se detallarán los errores RMSE para cada algoritmo utilizado.

Algoritmos	Valor del RMSE
Ridge	1.9840159398067003
SVR	2.046756025988088
Lineal	2051372370447.3508
Bosques Aleatorios	1.9964432291671563

Tras evaluar los errores de los diferentes algoritmos de predicción, se pueden extraer las siguientes conclusiones significativas sobre su desempeño en la estimación de las ventas globales de videojuegos:

- **Ridge:** El algoritmo Ridge se destaca como el mejor en términos de precisión, evidenciado por su bajo error de RMSE. Esta capacidad para generalizar de manera efectiva sugiere que el modelo Ridge es altamente competente en la predicción de las ventas globales.
- **Máquina de Soporte Vectorial para Regresión (SVR):** Aunque ligeramente superado por Ridge, SVR exhibe un rendimiento sólido, proporcionando predicciones precisas. Su capacidad para manejar relaciones no lineales contribuye a su eficacia en la tarea de predicción.
- **Regresión Lineal:** La Regresión Lineal presenta un error extremadamente alto, indicando que este modelo podría no ser apropiado para el conjunto de datos específico. La simplicidad de la Regresión Lineal puede limitar su capacidad para capturar la complejidad de las relaciones en los datos.
- **Bosques Aleatorios:** El modelo de Bosques Aleatorios muestra un rendimiento cercano al de Ridge, demostrando una sólida capacidad para predecir las ventas globales con precisión. Su capacidad para manejar la complejidad y las interacciones en los datos contribuye a su eficacia.

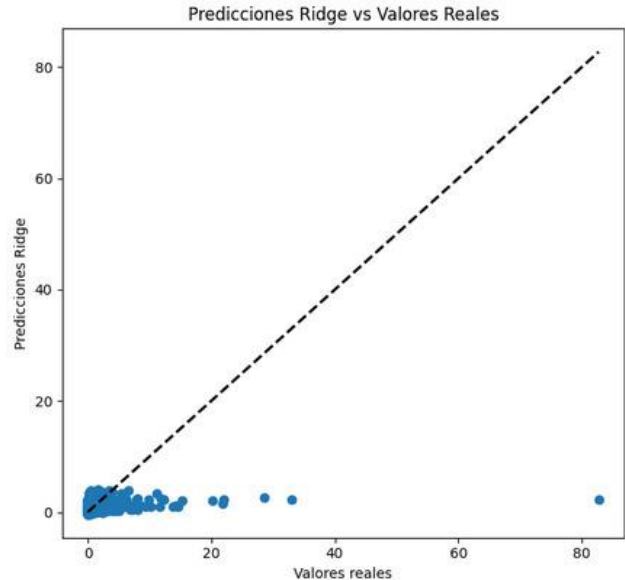
Estas conclusiones respaldan la elección del modelo Ridge como el más adecuado para la tarea de predicción en este contexto específico, brindando insights valiosos para la toma de decisiones en la industria de videojuegos.

II. Visualización de errores

En este apartado, se presenta una serie de gráficos que detallan la relación entre las predicciones de ventas globales realizadas por los distintos algoritmos de regresión (Random Forest, SVR, Regresión Lineal y Ridge) y los

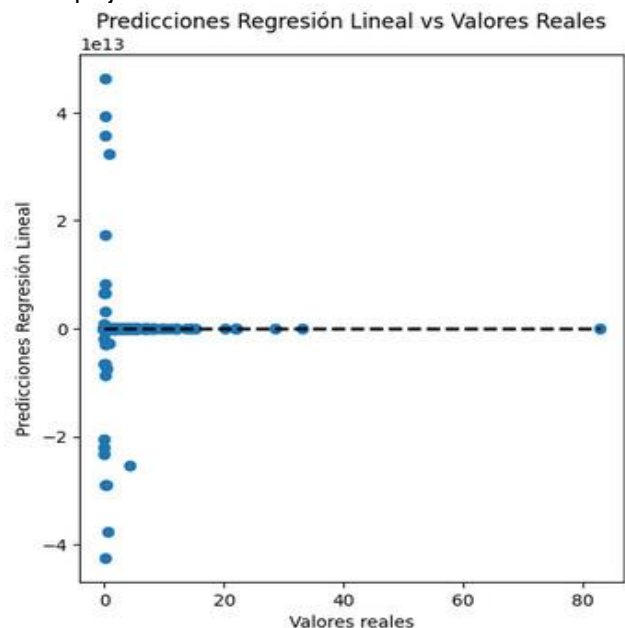
valores reales. Estas visualizaciones proporcionan una evaluación más intuitiva del desempeño de cada modelo, permitiendo identificar patrones y tendencias en los errores de predicción.

Comenzando con el gráfico correspondiente al algoritmo de Ridge, se observa una alineación más cercana de los puntos azules con la línea diagonal, indicando que las predicciones están más próximas a los valores reales. Este patrón refuerza la conclusión obtenida previamente sobre la eficacia de Ridge en la tarea de predicción.



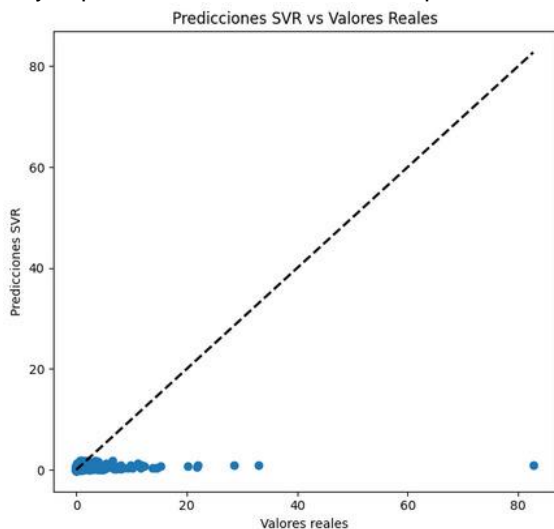
30. Desempeño del modelo de Regresión Ridge.

En el caso de la Regresión Lineal, la dispersión de los puntos azules revela una mayor variabilidad en las predicciones, con muchos puntos alejados de la línea ideal. Esta observación sugiere que la Regresión Lineal puede no ser la elección más adecuada para modelar la complejidad de los datos.

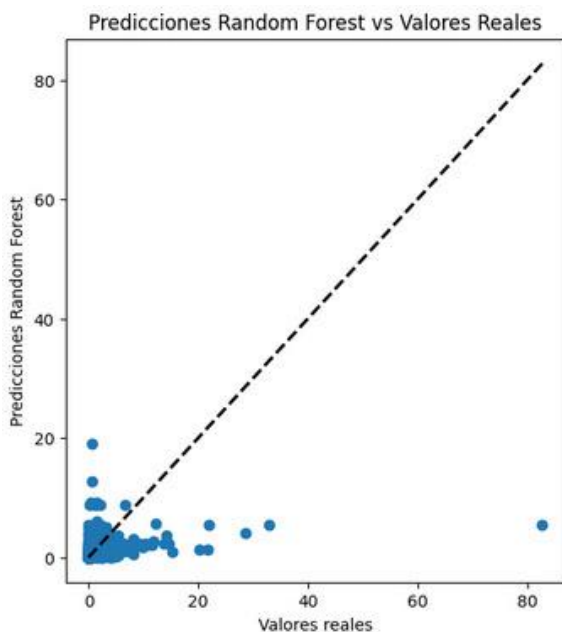


31. Desempeño del modelo de Regresión Lineal.

La comparación entre Random Forest y SVR muestra que ambos presentan puntos cercanos a la línea de referencia, pero la dispersión de los puntos en el gráfico de SVR es ligeramente mayor, indicando una menor consistencia en las predicciones. Aunque ambos modelos son efectivos, Random Forest parece ofrecer una mayor precisión en este contexto específico.



32. Desempeño del modelo SVR.



33. Desempeño del modelo Bosques Aleatorios.

Estas visualizaciones complementan las métricas de error previamente calculadas, proporcionando una perspectiva más completa del rendimiento de cada algoritmo. La interpretación de estos gráficos permite destacar las fortalezas y debilidades de cada modelo, brindando información valiosa para la toma de decisiones en el desarrollo y la implementación de estrategias en la industria de videojuegos.

III. Predicción con nuevos juegos

En esta sección, se llevó a cabo un ejercicio predictivo mediante la aplicación de los modelos de regresión entrenados previamente. El objetivo fue realizar predicciones de ventas globales para juegos futuros en función de sus características específicas (género, publisher, año y plataforma). Se abordaron tres escenarios distintos, cada uno formulado como una pregunta específica:

- Predicción N°1: ¿Cuáles serán las ventas para un juego de la empresa "Take Two Interactive" de género "Action" para la plataforma PC?
- Predicción N°2: ¿Cuáles serán las ventas para un juego de la empresa "Nintendo" de género "Platform" para la consola Wii?
- Predicción N°3: ¿Cuáles serán las ventas para un juego de la empresa "Ubisoft" de género "Sports" para la plataforma PC?

Como ya se mencionó en la explicación de los códigos de predicción, se desarrollaron códigos específicos para cada algoritmo de regresión utilizado (Ridge, SVR, Regresión Lineal y Random Forest). Estos códigos aplican los modelos entrenados a un conjunto de nuevas características correspondientes a los juegos planteados en las predicciones. Las ventas globales predichas para cada juego fueron registradas y se presentan en una tabla a continuación. Estos resultados ofrecen una visión anticipada de las posibles ventas de juegos futuros, proporcionando información valiosa para la planificación y estrategias de la industria de videojuegos.

Casos analizados	Ridge	SVR	Lineal	Random Forest
Predicción 1	0.74	0.18	0.73	0.33
Predicción 2	2.61	0.91	2.64	4.15
Predicción 3	0.13	0.07	0.10	0.05

Respecto al primer caso, la predicción para un juego de la empresa Take Two Interactive, de género "Action" para la plataforma PC, sugiere que las ventas estimadas de alrededor de 700,000 unidades son consistentes con el historial de la empresa en la creación de juegos de acción para PC, según el modelo Ridge, el cual se destaca por su precisión.

En cuanto al segundo caso, que involucra a la empresa Nintendo, género "Platform" y plataforma Wii, los resultados indican que un nuevo juego con estas características podría generar ganancias significativas. Aunque el valor generalizado por Random Forest es elevado, se destaca que el modelo Ridge, con un error ligeramente menor, ofrece una predicción más precisa, sugiriendo un potencial éxito en el mercado.

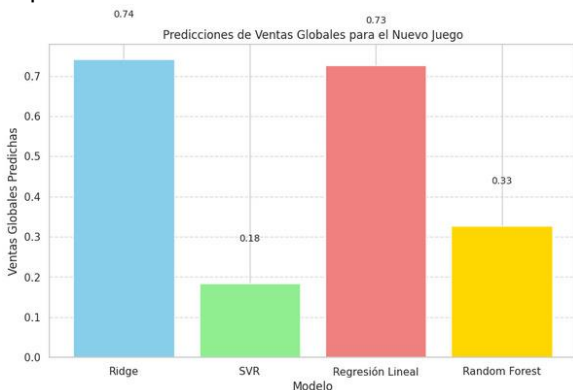
Para el tercer caso, relacionado con la empresa Ubisoft, género "Sports" y plataforma PC, las predicciones revelan valores mínimos de ventas en todos los modelos. Esta situación se interpreta

considerando que Ubisoft no tiene antecedentes de desarrollar juegos de este género, sugiriendo que lanzar un nuevo juego con estas características podría resultar en ganancias mínimas o incluso pérdidas.

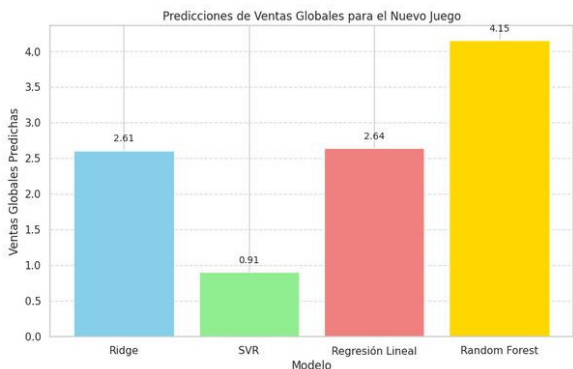
Estas conclusiones proporcionan información valiosa para la toma de decisiones estratégicas en la industria de videojuegos, destacando la importancia de conocer el historial y las especialidades de cada empresa al planificar nuevos lanzamientos.

IV. Visualizaciones de nuevos juegos

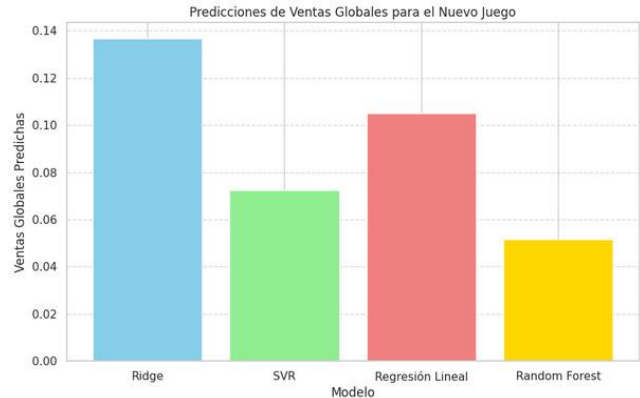
A continuación, se presentan las gráficas de barras correspondientes a cada caso mencionado anteriormente. Estas visualizaciones permiten una comparación directa de los resultados de las predicciones generadas por los diferentes algoritmos. Cada gráfico proporciona una representación clara de cómo se desempeñan los modelos en términos de predicciones de ventas globales para los nuevos juegos propuestos, facilitando la evaluación de su rendimiento en cada escenario específico.



34. Grafica para predicción N°1.



35. Grafica para predicción N°2.



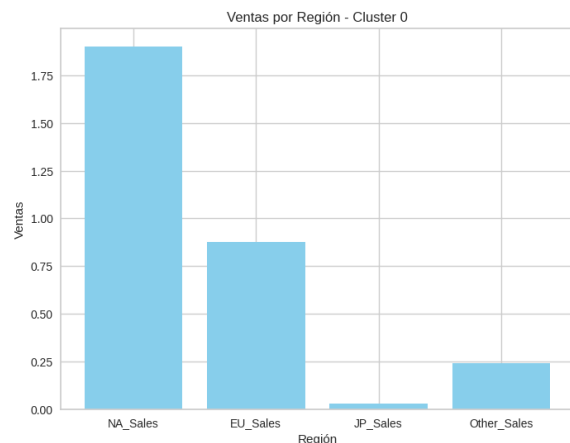
36. Grafica para predicción N°3.

3.2 Resultados de Agrupación

A continuación, se presentan los resultados obtenidos al aplicar el algoritmo de clustering DBSCAN. Con el fin de poder analizar y comparar cada clúster, lo que se hizo fue calcular la media de cada característica para cada clúster, obteniendo los siguientes valores.

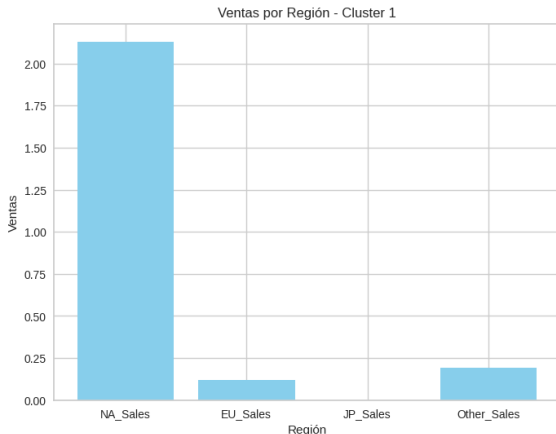
	Clúster 0	Clúster 1	Clúster 2	Clúster 3
NA_Sales	1.902	2.130	0.150	0.051
EU_Sales	0.878	0.122	0.073	0.023
JP_Sales	0.030	0.0006	0.039	1.226
Other_Sales	0.243	0.190	0.023	0.008
Genre_Label	7.7	10	4.9	7.2

A partir de estos resultados, se generaron cuatro gráficos de barras, uno por cada clúster y, teniendo en cuenta el contexto del mercado, se obtuvieron posibles deducciones de por qué los clústeres están formados de esa manera y qué información podemos obtener de esto.



37. Gráfico de barras del clúster 0.

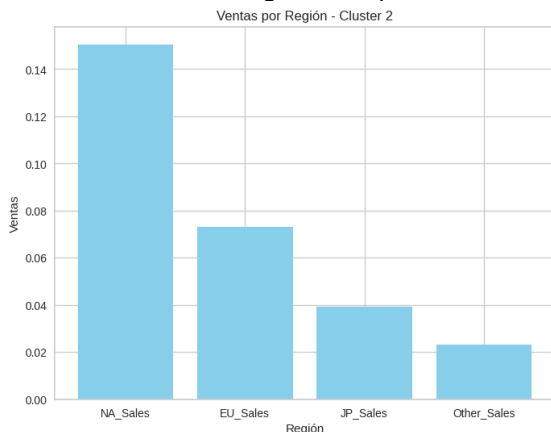
Este es el gráfico obtenido para el clúster 0, el cual corresponde al género "Shooter". Este es un resultado esperado para la característica "NA_Sales" debido a que, según lo analizado en el EDA, empresas norteamericanas tales como Electronic Arts y Activision, creadoras de populares juegos como Battlefield y Call Of Duty se encuentran en el top 5 de ventas globales.



38. Gráfico de barras del clúster 1.

El clúster 1 corresponde al género "Sports". En este caso se puede observar una gran diferencia entre "NA_Sales" con respecto a las demás regiones. Si bien Nintendo, empresa japonesa, se encuentra entre los primeros puestos de este género con el juego "Wii Sports", los valores obtenidos en este gráfico no resultan sorprendentes.

En América del Norte, deportes como el fútbol americano, básquet y béisbol son muy populares. Entre los videojuegos más famosos, se puede destacar el "NBA 2K", el cual se lanza año a año. Este resultado da cuenta que las preferencias y gustos del resto de regiones no están enfocadas en el género "Sports".

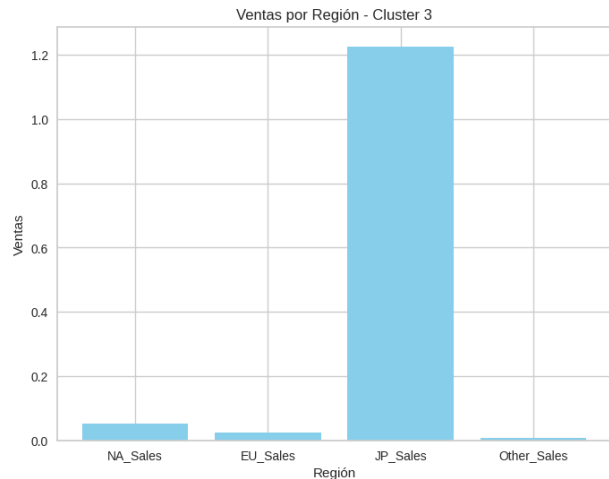


39. Gráfico de barras del clúster 2.

Correspondiente al género "Puzzle", el clúster 2 arroja el gráfico menos esperado de todos.

Este es el resultado más sorprendente ya que el género puzzle obtuvo un porcentaje bastante bajo a la hora de realizar el análisis estadístico.

Si bien las ventas en América del Norte siguen siendo predominantes y, en general, las ventas son muy bajas comparadas a los gráficos anteriores, se podría considerar al género puzzle un nicho de mercado a tener en cuenta y digno de explorar por las empresas.



40. Gráfico de barras del clúster 3.

Finalmente, tenemos el género "Role-Playing" para el clúster 3. Como ejemplos de videojuegos RPG (Role-Playing Game) podemos mencionar World of Warcraft, The Witcher, Final Fantasy, Diablo, Dark Souls, Pokemon Red/Blue, entre otros.

Este resultado también era de esperarse, debido a que, por ejemplo, el juego más vendido para la región de Japón es el Pokémon Red/Blue, lo que marca una clara preferencia de los jugadores de esta región con este tipo de videojuegos.

4. Discusiones

La integración de técnicas de Análisis Exploratorio de Datos (EDA) y algoritmos de aprendizaje automático en el contexto de la industria de videojuegos ha generado una serie de observaciones y conclusiones significativas. Al analizar los resultados obtenidos, es crucial evaluar el impacto tanto teórico como práctico de estos hallazgos.

Desde una perspectiva teórica, la efectividad del EDA para abordar valores faltantes y revelar patrones intrínsecos destaca la importancia de una preparación de datos exhaustiva. La segmentación identificada mediante algoritmos de agrupación proporciona una visión detallada de la diversidad en el mercado de videojuegos, subrayando la complejidad de las preferencias de los consumidores.

En términos de aplicaciones prácticas, las predicciones de ventas globales generadas por modelos de regresión tienen implicaciones significativas para la toma de decisiones en la industria. Es particularmente notable que el algoritmo Ridge haya

demostrado ser el más preciso, sugiriendo su utilidad para estimaciones futuras. Este hallazgo abre oportunidades concretas para las empresas de videojuegos al orientar estrategias de desarrollo y marketing hacia géneros, plataformas y editoras específicas.

La comparación entre diferentes algoritmos y la presentación visual de las predicciones permiten una evaluación integral de su desempeño. Las gráficas de dispersión ilustran claramente cómo las predicciones se alinean con los valores reales, ofreciendo una perspectiva visual que complementa las métricas de error.

Además, otra conclusión relevante se deriva del análisis de la viabilidad de nuevos juegos por parte de desarrolladores independientes o pequeños. Se observa que la probabilidad de éxito para estos nuevos juegos es reducida, ya que las grandes empresas dominan gran parte del mercado y son responsables de los juegos de mayor éxito a nivel mundial. En este contexto, se sugiere que los desarrolladores más pequeños podrían considerar la posibilidad de unirse a estas grandes empresas para aumentar sus posibilidades de éxito.

5. Conclusiones

I. Resumen de los principales resultados

El análisis exploratorio de datos (EDA) y la aplicación de algoritmos de aprendizaje automático en la industria de videojuegos han proporcionado resultados significativos. Durante el EDA, se identificaron patrones clave y se abordaron los valores faltantes, garantizando la integridad de los análisis subsiguientes. Los algoritmos de agrupación revelaron segmentos distintos en el conjunto de datos, mientras que los modelos de regresión destacaron el rendimiento preciso del algoritmo Ridge en la predicción de ventas globales.

II. Implicaciones (teóricas y/o prácticas)

Los hallazgos teóricos y prácticos de este estudio tienen implicaciones sustanciales en la industria de videojuegos. Desde una perspectiva teórica, la aplicación efectiva de técnicas de EDA y aprendizaje automático destaca la importancia de abordar datos complejos de manera integral. En términos prácticos, las predicciones de ventas y la segmentación de juegos según características específicas ofrecen información valiosa para la toma de decisiones estratégicas en el desarrollo y marketing de videojuegos.

III. Planes para futuras investigaciones

Para futuras investigaciones, se sugiere explorar la incorporación de más variables en el análisis, como la influencia de críticas y reseñas en las ventas. Además, la evaluación de

algoritmos adicionales y la comparación de su rendimiento podrían enriquecer aún más la comprensión de la predicción de ventas en la industria de videojuegos. Estas expansiones podrían contribuir a refinamientos adicionales en las estrategias de desarrollo y comercialización de videojuegos.

6. Referencias

- [1] G. Smith, «Kaggle,» 2016. Available: <https://www.kaggle.com/datasets/gregorut/video-gamesales>.
- [2] R. J. Python, «Limpieza de datos,» de *Hands On Data Preprocessing in Python*, p. Capítulos 9 y 11.
- [3] M. A. Acikgoz, «The Use of Data for Understanding the Video Game Market,» Streamlit. Available: <https://maliackgoz-py4ds-notes-streamlit-app-0p7r5c.streamlit.app/>.
- [4] A. Géron, «Training Models,» de *Hands On Machine Learning with Scikit Learn, Keras, and TensorFlow*, O'Reilly Media, 2022.
- [5] A. Géron, «Unsupervised Learning Techniques,» de *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, 2022.