Universidad “Mayor de San Andrés”
Facultad de Ciencias Puras y Naturales
Carrera de Informatica

INTELIGENCIA ARTIFICIAL INF – 354

Análisis y modelado de datos de ventas de videojuegos utilizando Python

NOMBRE:

Roque Mendoza Alex

La Paz – Bolivia

Análisis y modelado de datos de ventas de videojuegos utilizando Python

Alex Roque Mendoza

Universidad Mayor de San Andres

La Paz, Bolivia

aroque9@umsa.bo

Abstract:

The video game industry has experienced exponential growth in recent decades, making data analysis of video game sales increasingly important for understanding trends and key factors that drive game success. In this article, we utilize Python to perform a comprehensive analysis of a video game sales dataset and build predictive models that help us understand the factors influencing global sales.

Through data preparation, exploratory analysis, and modeling techniques, we gain insights into the relationships between different variables and their impact on sales. We employ machine learning algorithms such as linear regression, logistic regression, and decision trees to predict global video game sales based on available independent variables.

Evaluation metrics such as root mean squared error (RMSE) and coefficient of determination (R^2) are used to assess model performance. Model improvement techniques, including hyperparameter optimization, feature selection, and regularization, are employed to enhance prediction accuracy.

The results of our analysis and modeling provide valuable insights into the underlying patterns and factors affecting game success. This approach equips industry professionals with informed decision-making tools to maximize opportunities for success in the competitive video game market.

Introducción:

El mercado de los videojuegos ha experimentado un crecimiento exponencial en las últimas décadas, y el análisis de datos de ventas de videojuegos se ha vuelto cada vez más importante para comprender las tendencias y los factores clave que impulsan el éxito de un juego. En este artículo, utilizaremos Python para realizar un análisis detallado de un conjunto de datos de ventas de videojuegos y construir modelos predictivos que nos ayuden a comprender los factores que influyen en las ventas globales.

I. Preparación de los datos:

Antes de comenzar el análisis, es crucial preparar los datos adecuadamente. Esto incluye la carga del conjunto de datos en Python utilizando la biblioteca Pandas, la limpieza de datos, el manejo de valores faltantes y la transformación de los datos si es necesario. Además, podemos realizar un análisis exploratorio inicial para comprender mejor la estructura de los datos y detectar posibles problemas.

Librerias Utilizadas

Para el siguiente proyecto se utilizó las siguientes librerías estas son las bibliotecas utilizadas en el código proporcionado. A continuación, se explica para qué sirve cada una de ellas:

numpy (np): Es una biblioteca de Python que se utiliza para realizar operaciones matemáticas y numéricas eficientes. Proporciona estructuras de datos y funciones para manipular matrices y realizar cálculos numéricos.

pandas (pd): Es una biblioteca de Python que proporciona estructuras de datos de alto rendimiento y fáciles de usar, como DataFrames, para el análisis y manipulación de datos. Se utiliza ampliamente para la manipulación, limpieza y preparación de conjuntos de datos antes de entrenar modelos de aprendizaje automático.

tensorflow (tf): Es una biblioteca de aprendizaje automático y de inteligencia artificial de código abierto desarrollada por Google. Proporciona herramientas para construir y entrenar modelos de aprendizaje automático, incluidas redes neuronales.

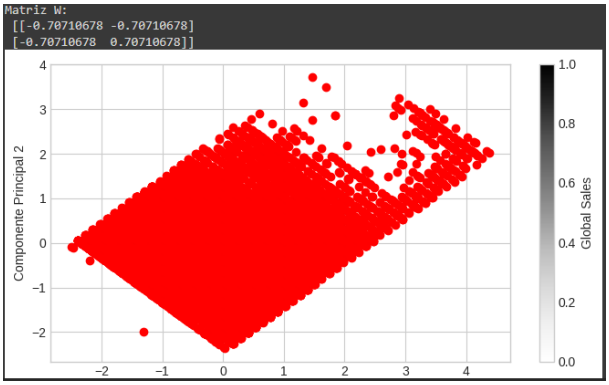
TensorFlow se utiliza ampliamente en la comunidad de investigación y desarrollo para aplicaciones de aprendizaje automático.

sklearn.preprocessing.OneHotEncoder: Es una clase proporcionada por la biblioteca scikit-learn (sklearn) que se utiliza para codificar variables categóricas en una

representación numérica. En el código proporcionado, se utiliza para convertir las etiquetas de clase en una representación de tipo "onehot encoding" antes de entrenar el modelo.

sklearn.model_selection.train_test_split: Es una función de la biblioteca scikit-learn que se utiliza para dividir un conjunto de datos en conjuntos de entrenamiento y prueba. En el código proporcionado, se utiliza para dividir los datos en conjuntos de entrenamiento y prueba antes de entrenar el modelo.

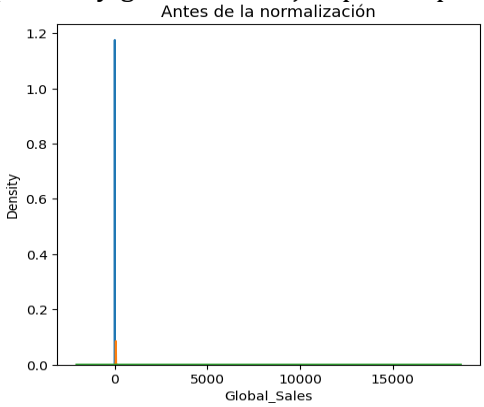
sklearn.metrics.confusion_matrix: Es una función de la biblioteca scikit-learn que se utiliza para calcular la matriz de confusión, que es una herramienta para evaluar el rendimiento de un modelo de clasificación.



La matriz de confusión muestra el número de predicciones correctas e incorrectas realizadas por el modelo en cada clase. En resumen, estas bibliotecas y funciones son utilizadas para cargar y preparar los datos, construir y entrenar el modelo de red neuronal, y evaluar su rendimiento utilizando la matriz de confusión

II. Análisis exploratorio de datos:

El análisis exploratorio de datos nos ayuda a obtener una visión general de los patrones y características de los datos. Utilizando bibliotecas como Matplotlib y Seaborn, podemos crear visualizaciones informativas, como gráficos de barras, diagramas de dispersión y gráficos de cajas, para explorar la relación entre diferentes variables, identificar tendencias y detectar posibles outliers.



III. Modelado de datos:

Una vez completado el análisis exploratorio, podemos pasar al modelado de datos. En este paso, utilizaremos técnicas de aprendizaje automático y modelado estadístico para construir modelos predictivos.

```
Matriz de covarianza:
 [[1.00006149 0.17668241]
 [0.17668241 1.00006149]]
Eigen Vectores:
 [[-0.70710678 -0.70710678]
 [ 0.70710678 -0.70710678]]
Eigen Valores:
 [0.82337908 1.17674389]
Autovalores en orden descendente:
1.1767438941001214
0.8233790768787275
```

Podemos utilizar algoritmos como regresión lineal, regresión logística o árboles de decisión para predecir las ventas globales de videojuegos en función de las variables independientes disponibles en el conjunto de datos.

IV. Evaluación y mejora del modelo:

Después de construir los modelos, es esencial evaluar su rendimiento y realizar mejoras si es necesario. Utilizaremos métricas de evaluación como el error cuadrático medio (RMSE) o el coeficiente de determinación (R^2) para medir la precisión del modelo. Si el rendimiento del modelo no es satisfactorio, podemos considerar la optimización de hiperparámetros, la selección de variables o la aplicación de técnicas de regularización para mejorar la calidad de las predicciones.

V. Conclusiones:

El análisis y modelado de datos de ventas de videojuegos utilizando Python nos brinda una perspectiva valiosa sobre las tendencias y los factores que afectan el éxito de los juegos.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.006
Model:	OLS	Adj. R-squared:	0.006
Method:	Least Squares	F-statistic:	81.30
Date:	Thu, 15 Jun 2023	Prob (F-statistic):	2.21e-19
Time:	18:33:09	Log-Likelihood:	-23067.
No. Observations:	13012	AIC:	4.614e+04
Df Residuals:	13010	BIC:	4.615e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	39.2751	4.296	9.142	0.000	30.854	47.696
x1	-0.0193	0.002	-9.017	0.000	-0.024	-0.015

Omnibus:	21525.767	Durbin-Watson:	1.995
Prob(Omnibus):	0.000	Jarque-Bera (JB):	20379253.370
Skew:	11.077	Prob(JB):	0.00
Kurtosis:	195.608	Cond. No.	6.90e+05

Con las herramientas y bibliotecas adecuadas, podemos explorar y comprender los patrones subyacentes en los datos, así como construir modelos predictivos precisos. Este enfoque nos ayuda a tomar decisiones informadas en la industria de los videojuegos y maximizar las oportunidades de éxito.

Referencias:

McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.

VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and Statistical Modeling with Python. Proceedings of the 9th Python in Science Conference, 57-61.