INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

UNIDAD ACADÉMICA:

ESCUELA SUPERIOR DE CÓMPUTO

PROGRAMA ACADÉMICO:

Licenciatura en Ciencia de Datos

UNIDAD DE APRENDIZAJE:

Series de Tiempo

NOMBRE DEL PROFESOR:

Daniel Jiménez Alcantar

NOMBRE DE LA ACTIVIDAD:

Practica 3 “Regresión lineal en contexto de series de tiempo”.

INTEGRANTES

Aguilar Ramírez Carlos Francisco

Arista Romero Juan Ismael

Jiménez Flores Luis Arturo

Vazquez Martin Marlene Gabriela

FECHA: 14 de marzo de 2025 **Grupo**:6AV1

Introducción

Las series de tiempo son fundamentales en el análisis de datos, ya que permiten modelar y predecir eventos basados en su comportamiento pasado. En esta práctica, se aplicaron técnicas estadísticas y computacionales para analizar la evolución de los streams de canciones en la plataforma Spotify, con el objetivo de identificar tendencias y realizar predicciones sobre su popularidad.

El enfoque principal de este análisis fue el uso de regresión lineal y modelos de series de tiempo, como el modelo de Media Móvil Autorregresiva Integrada (ARIMA), para examinar cómo evolucionan las reproducciones de una canción a lo largo del tiempo. Se consideraron aspectos clave como la dependencia temporal, la autocorrelación y la estacionariedad, los cuales diferencian el análisis de series de tiempo de una regresión estándar.

Para este estudio, se utilizó un dataset que contiene información sobre canciones, su fecha de lanzamiento y la cantidad de streams acumulados. A partir de estos datos, se realizaron los siguientes pasos:

- **Preprocesamiento y exploración del dataset** para adecuarlo al análisis de series de tiempo.
- **Análisis de estacionariedad y transformación de la serie** para cumplir con los requisitos de modelado.
- **Aplicación de regresión lineal** para identificar tendencias en los streams a lo largo del tiempo.
- **Implementación del modelo ARIMA**, ajustando sus parámetros para mejorar la precisión de la predicción.
- **Evaluación del modelo** mediante métricas de error como RMSE para validar su desempeño.

El propósito de esta práctica es comprender el comportamiento de los streams de canciones y evaluar qué modelo logra capturar mejor su evolución. Los resultados obtenidos permitirán conocer las ventajas y limitaciones de cada enfoque, lo que es de gran utilidad en contextos donde se requiere pronosticar tendencias en datos temporales.

Problemática

En la actualidad, las plataformas de streaming como **Spotify** juegan un papel fundamental en la industria musical, ya que permiten a los artistas distribuir su música y alcanzar grandes audiencias. Sin embargo, la popularidad de una canción no es estática, sino que cambia con el tiempo debido a múltiples factores, como tendencias musicales, estrategias de marketing y eventos virales en redes sociales.

Uno de los principales desafíos para la industria es predecir el comportamiento de los streams de una canción después de su lanzamiento. ¿Cómo se puede determinar si una canción tendrá un crecimiento sostenido o si su popularidad disminuirá rápidamente? ¿Existen patrones que permitan prever el éxito de una canción en función de su historial de streams? Estas preguntas son clave tanto para artistas como para productores musicales que buscan optimizar sus estrategias de promoción.

El problema radica en que los datos de streams forman una serie de tiempo, lo que significa que están influenciados por factores temporales y no pueden analizarse con métodos convencionales de regresión sin considerar aspectos como la dependencia temporal, la autocorrelación y la estacionariedad.

Por ello, esta práctica busca abordar el problema desde un enfoque de **modelado** de series de tiempo, explorando distintos métodos como:

Regresión Lineal: Para analizar la tendencia general en los streams.

ARIMA: Para modelar patrones temporales y hacer predicciones.

El objetivo es encontrar el modelo que mejor se ajuste a la evolución de los streams, permitiendo una mejor comprensión de su comportamiento y facilitando la toma de decisiones en la industria musical.

Modelo Estadístico

Para modelar la evolución de los streams de canciones a lo largo del tiempo, se utilizaron dos enfoques estadísticos principales: Regresión Lineal y Modelos Autorregresivos de Series de Tiempo (ARIMA).

Regresión Lineal

La Regresión Lineal es un modelo estadístico que permite analizar la relación entre una variable dependiente (streams) y una variable independiente (fecha de lanzamiento). Sin embargo, en series de tiempo, la regresión estándar tiene ciertas limitaciones, ya que asume independencia entre las observaciones, lo cual no es válido en datos temporales.

La ecuación de la regresión lineal utilizada en este análisis es la siguiente:

$$y_t = \beta_0 + \beta_1 t + \epsilon_t$$

Donde:

- y_t representa la cantidad de streams en el tiempo t.
- β_0 es la intersección (valor de streams cuando t=0).

- β1 es la pendiente (tasa de cambio de los streams con el tiempo).
- ϵt es el término de error aleatorio.

Para mejorar el ajuste del modelo, se aplicó una **transformación logarítmica** en la variable de streams, con el objetivo de reducir la variabilidad y capturar mejor la tendencia.

Modelos Autorregresivos y ARIMA

Dado que las series de tiempo tienen dependencia temporal, se utilizó el modelo ARIMA (AutoRegressive Integrated Moving Average), que combina tres componentes:
ARIMA(p,d,q)

Donde:

- **p**: Número de retardos (lags) en la parte autoregresiva (AR).
- **d**: Número de diferenciaciones para hacer la serie estacionaria (Integrated-I).
- **q**: Número de términos de media móvil (Moving Average - MA).

Para determinar los mejores valores de p y q, se analizaron las funciones de autocorrelación (ACF) y autocorrelación parcial (PACF). Además, se utilizó AutoARIMA, que seleccionó automáticamente ARIMA(0,1,2) como el modelo óptimo.

Prueba de Estacionariedad

Antes de aplicar ARIMA, se realizó la prueba de Dickey-Fuller aumentada (ADF) para verificar si la serie era estacionaria.

- La prueba inicial indicó que la serie no era estacionaria (p-valor > 0.05).
- Se aplicó una diferenciación de primer orden, lo que permitió hacer la serie estacionaria (p-valor < 0.05).

Modelo Computacional

Regresión Lineal

El primer modelo que utilizamos fue la regresión lineal, para identificar características de la función. Y la graficamos para entenderla visualmente también.

Regresión Lineal

```python
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm

# Convertir la fecha en un número ordinal
df_time_series['date_ordinal'] = df_time_series['release_date'].map(lambda x: x.toordinal())

# Definir variables para regresión
X = df_time_series['date_ordinal']  # Variable independiente (fecha)
y = df_time_series['streams']       # Variable dependiente (streams)

# Agregar una constante para la intersección en la regresión
X = sm.add_constant(X)

# Ajustar el modelo de regresión lineal
model = sm.OLS(y, X).fit()

# Obtener la línea de tendencia
df_time_series['trend'] = model.predict(X)

# Graficar los datos originales y la regresión
plt.figure(figsize=(12,6))
plt.scatter(df_time_series['release_date'], df_time_series['streams'], label="Datos Reales", alpha=0.5)
plt.plot(df_time_series['release_date'], df_time_series['trend'], color='red', linewidth=2, label="Regresión Lineal")

# Etiquetas y título
plt.xlabel("Fecha de Lanzamiento")
plt.ylabel("Streams")
plt.title("Regresión Lineal de Streams en el Tiempo")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)

# Mostrar el gráfico
plt.show()

# Mostrar resultados del modelo
print(model.summary())
```

Figura 1. Modelo de regresión Lineal

Regresión Lineal con Transformación Logarítmica

Transformamos la regresión para visualizar la aproximación de la forma funcional de los datos cuando no tienen una relación lineal, que si es el caso.

Regresión con Transformación Logarítmica

```python
import numpy as np

# Aplicar transformación logarítmica a los streams
df_time_series['log_streams'] = np.log1p(df_time_series['streams'])

# Definir variables para regresión
X = df_time_series['date_ordinal']
y = df_time_series['log_streams']

# Agregar una constante para la intersección
X = sm.add_constant(X)

# Ajustar modelo de regresión
model_log = sm.OLS(y, X).fit()

# Obtener la nueva tendencia
df_time_series['log_trend'] = model_log.predict(X)

# Graficar los datos con la transformación logarítmica
plt.figure(figsize=(12,6))
plt.scatter(df_time_series['release_date'], df_time_series['log_streams'], label="Datos Reales (Log)", alpha=0.5)
plt.plot(df_time_series['release_date'], df_time_series['log_trend'], color='red', linewidth=2, label="Regresión Lineal (Log)")

# Etiquetas y título
plt.xlabel("Fecha de Lanzamiento")
plt.ylabel("Log(Streams)")
plt.title("Regresión Lineal con Transformación Logarítmica")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)

# Mostrar el gráfico
plt.show()

# Mostrar los resultados del modelo
print(model_log.summary())
```

Figura 2. Modelo de regresión lineal con Transformación logarítmica

Modelo ARIMA

Aplicamos el modelo ARIMA primero para graficar e identificar visualmente la autocorrelación de nuestra variable objetivo “streams_diff”.

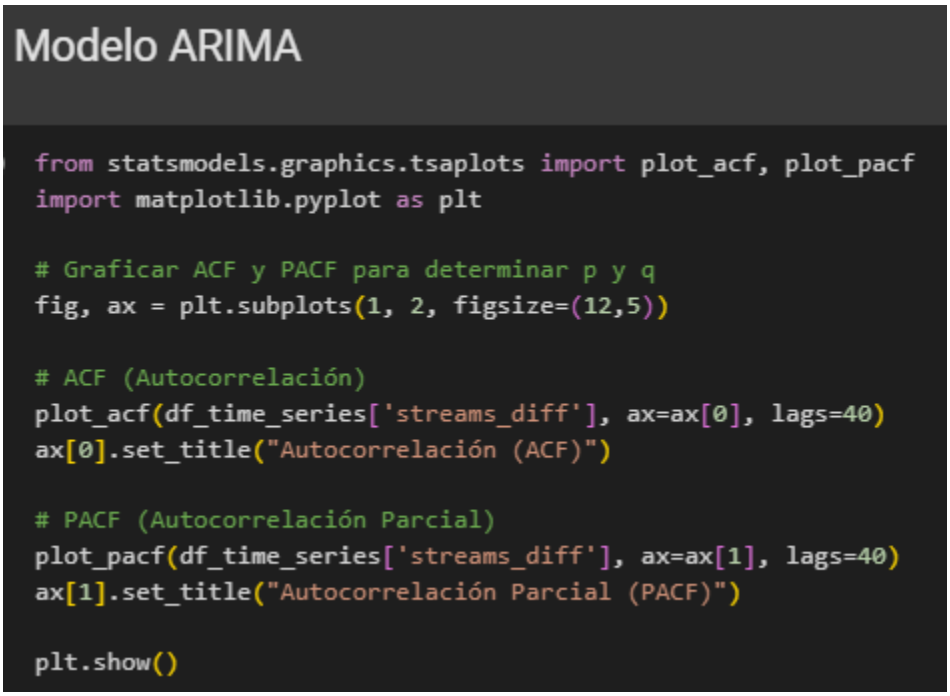

Figura 3. Código de gráficos de autocorrelación ARIMA

Posterior a identificar la autocorrelación, ya aplicamos el modelo como tal y ajustamos los hiperparámetros del entrenamiento, manualmente, para que quede adecuado a nuestras necesidades.

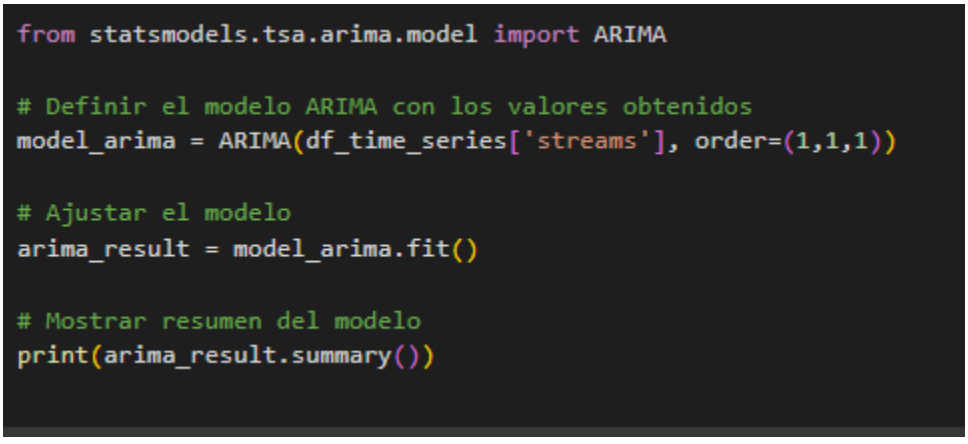

Figura 4. Ajuste de hiperparámetros en el modelo ARIMA

Posteriormente, generamos las predicciones que el modelo ARIMA arroja, previamente entrenado.

```python
# Generar predicciones dentro del conjunto de datos
df_time_series['arima_pred'] = arima_result.predict()

# Graficar datos reales y predicciones
plt.figure(figsize=(12,6))
plt.plot(df_time_series['release_date'], df_time_series['streams'], label="Datos Reales", alpha=0.5)
plt.plot(df_time_series['release_date'], df_time_series['arima_pred'], color='red', label="Predicción ARIMA")

# Etiquetas y título
plt.xlabel("Fecha de Lanzamiento")
plt.ylabel("Streams")
plt.title("Modelo ARIMA: Predicciones vs Datos Reales")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)

# Mostrar el gráfico
plt.show()
```

Figura 5. Código de las predicciones del modelo ARIMA

En sí ya tendríamos el modelo, sin embargo, hicimos un paso extra que es aplicar la biblioteca “pmdarima” para encontrar los mejores hiperparámetros para el modelo previo que usamos.

```python
!pip install pmdarima
from pmdarima import auto_arima

# Encontrar los mejores parámetros automáticamente
auto_model = auto_arima(df_time_series['streams'], seasonal=False, trace=True, stepwise=True)

# Mostrar los mejores parámetros encontrados
print("Mejor modelo encontrado:", auto_model.order)
```

Figura 6. Ajuste de hiperparámetros automáticos en el modelo ARIMA

Para finalizar esta sección, graficamos el modelo.

```python
from statsmodels.tsa.arima.model import ARIMA

# Definir el modelo ARIMA con los valores obtenidos
model_arima = ARIMA(df_time_series['streams'], order=(0,1,2))

# Ajustar el modelo
arima_result = model_arima.fit()

# Mostrar resumen del modelo
print(arima_result.summary())

# Generar predicciones dentro del conjunto de datos
df_time_series['arima_pred'] = arima_result.predict()

# Graficar datos reales y predicciones
plt.figure(figsize=(12,6))
plt.plot(df_time_series['release_date'], df_time_series['streams'], label="Datos Reales", alpha=0.5)
plt.plot(df_time_series['release_date'], df_time_series['arima_pred'], color='red', label="Predicción ARIMA")

# Etiquetas y título
plt.xlabel("Fecha de Lanzamiento")
plt.ylabel("Streams")
plt.title("Modelo ARIMA: Predicciones vs Datos Reales")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.5)

# Mostrar el gráfico
plt.show()
```

Figura 7. Código del gráfico de los mejores hiperparámetros datos reales vs predicciones

Metodología

Las series de tiempo, son conjuntos de datos los cuales son recopilados a lo largo del tiempo. El análisis de estas, se convierte en una técnica estadística para predecir y comprender el comportamiento de los datos a través del tiempo. Estas deben cumplir unos supuestos básicos:

- **Normalidad**: La cual se refiere a la distribución de los datos en la serie de tiempo, se dice que existe normalidad si estos se distribuyen alrededor de su media, siguiendo la forma de la campana de Gauss.
- **Independencia**: Los valores de la serie en algún punto no deben mostrar correlación con los valores anteriores o posteriores, si existe alguna relación temporal en los datos, entonces hay autocorrelación presente en la serie. Se observa por medio de la función de autocorrelación (ACF).
- **Homocedasticidad**: la varianza de los errores debe ser constante a lo largo del tiempo, es decir, la dispersión de los errores no debe depender del tiempo.
- **Linealidad**: la mayoría de modelos de series de tiempo, suelen asumir linealidad ya que implica que los cambios en una variable están directamente relacionados con los cambios en otra variable de manera proporcional.

Para cumplir con estos supuestos, aplicamos la prueba de Dickey-Fuller aumentada para evaluar la normalidad de los datos y posteriormente aplicamos la metodología de Box-Jenkins.

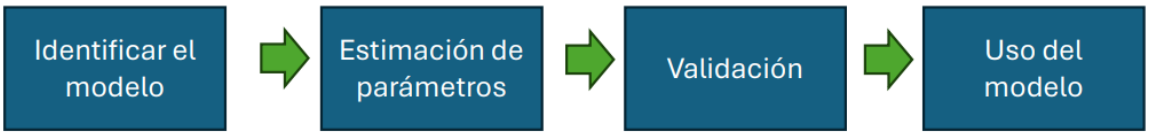

Figura 8. Proceso metodológico de Box-Jenkins

La cual se divide generalmente en tres etapas y en ocasiones cuenta con una cuarta:

1. **Identificación del modelo:** Se utilizan funciones de autocorrelación y autocorrelación parcial para determinar el orden apropiado del modelo ARIMA.
2. **Estimación de parámetros:** Se ajusta el modelo seleccionado a la serie temporal observada.
3. **Diagnóstico del modelo:** Se valida el modelo para asegurar que representa con precisión el proceso de generación de datos.
4. **Uso (pronóstico):** Esta etapa se trata de realizar pronósticos de la serie en el futuro.

Propuesta de solución para el análisis de una serie de tiempo con regresión lineal

a. Desarrolla las consideraciones clave que la diferencian del uso estándar de la regresión lineal: Dependencia temporal, la autocorrelación, la estacionariedad y la predicción.

En las series de tiempo, los datos están ordenados cronológicamente, lo que implica que los valores en un momento dado pueden depender de los valores en momentos anteriores. Esta **dependencia temporal** es una característica distintiva de las series de tiempo que no se encuentra en los análisis de regresión estándar, donde se asume que las observaciones son independientes. En el código se puede visualizar este gráfico que denota dependencia temporal.

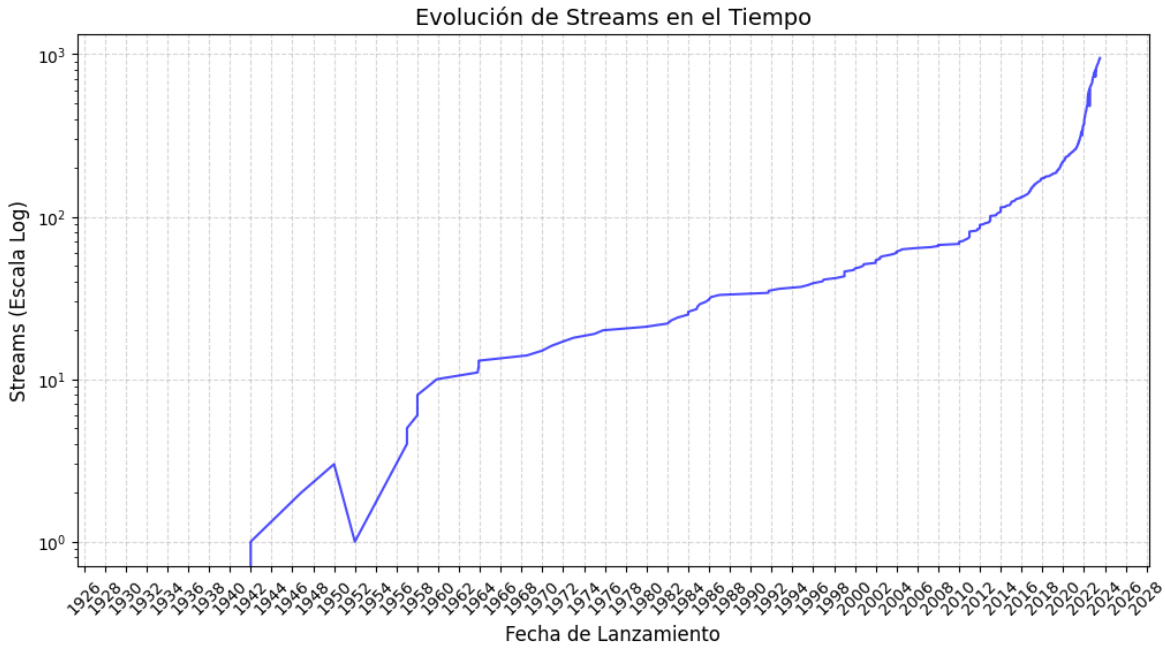

Figura 9. Demostración gráfica de la dependencia temporal.

Por otra parte, la **autocorrelación** se refiere a la correlación entre los valores de una serie de tiempo y sus propios valores rezagados. En otras palabras, es la medida de cuánto se relacionan los valores de la serie consigo mismos a lo largo del tiempo. Una alta autocorrelación puede violar el supuesto de **independencia** de los residuos en la regresión lineal estándar, lo que puede llevar a inferencias estadísticas incorrectas.

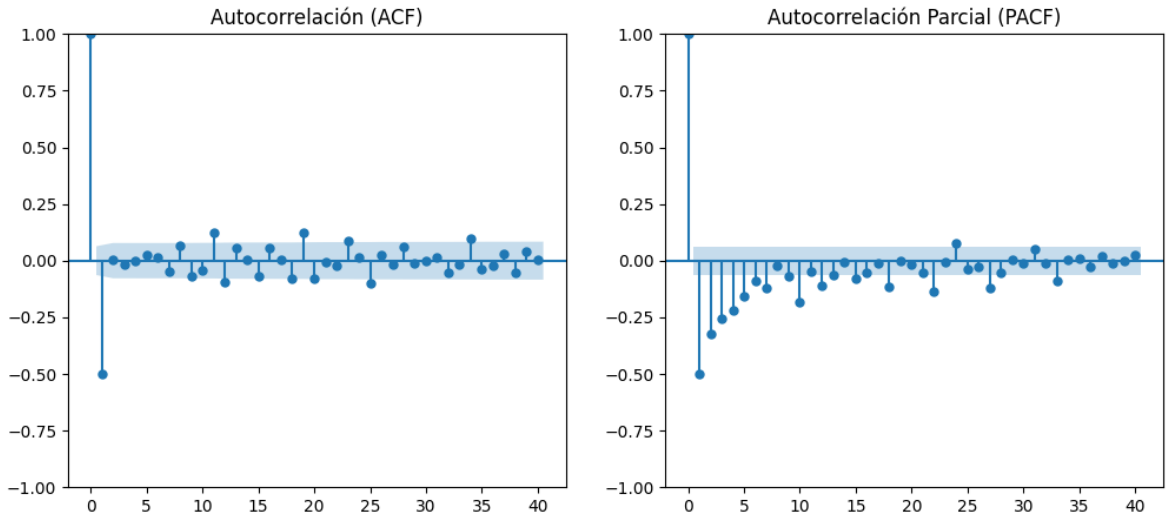

Figura 10. Demostración gráfica de la autocorrelación.

La **estacionariedad**, es una propiedad estadística de las series de tiempo que implica que las propiedades estadísticas de la serie, como la media y la varianza, permanecen constantes a lo largo del tiempo. Muchas técnicas de series de tiempo, incluida la regresión, asumen que la serie es estacionaria o puede transformarse en estacionaria.

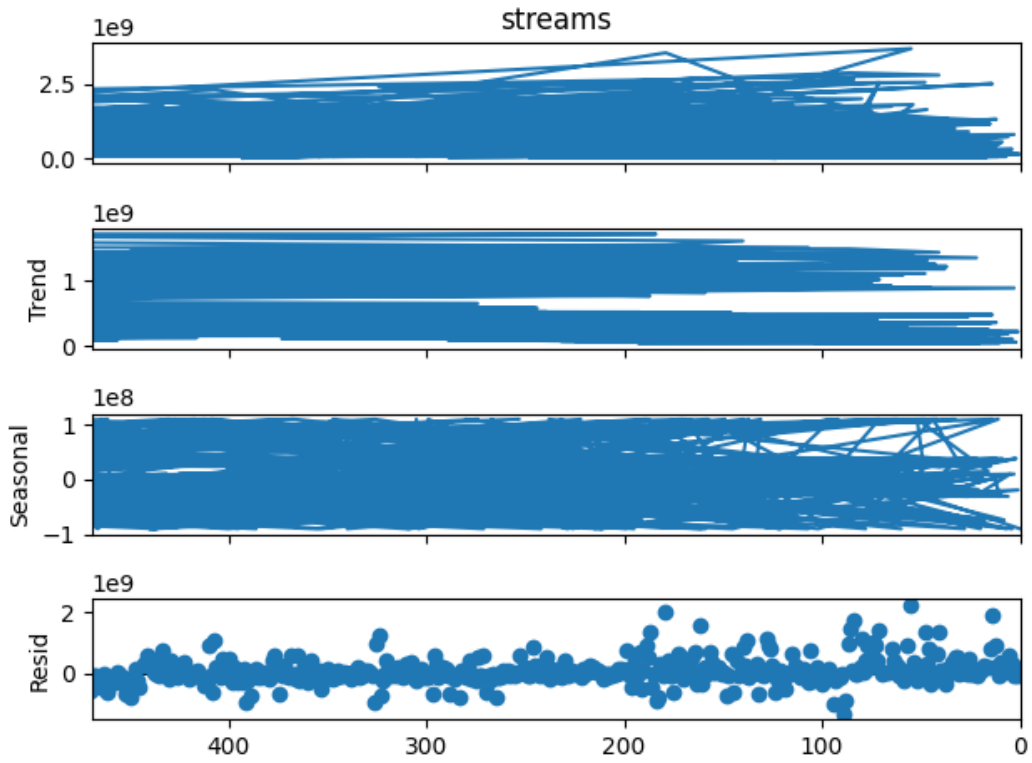

Figura 11. Demostración gráfica de la estacionalidad.

Finalmente, la regresión lineal y en el modelo ARIMA, dentro del contexto de las series de tiempo, a menudo se utiliza con el propósito de la **predicción**, es decir, **prever valores futuros** de la serie. La dependencia temporal y otras características específicas de las series de tiempo influyen en cómo se construye y evalúa un modelo de regresión para la predicción. Estos hallazgos se pueden ver en la figura 12 y 13.

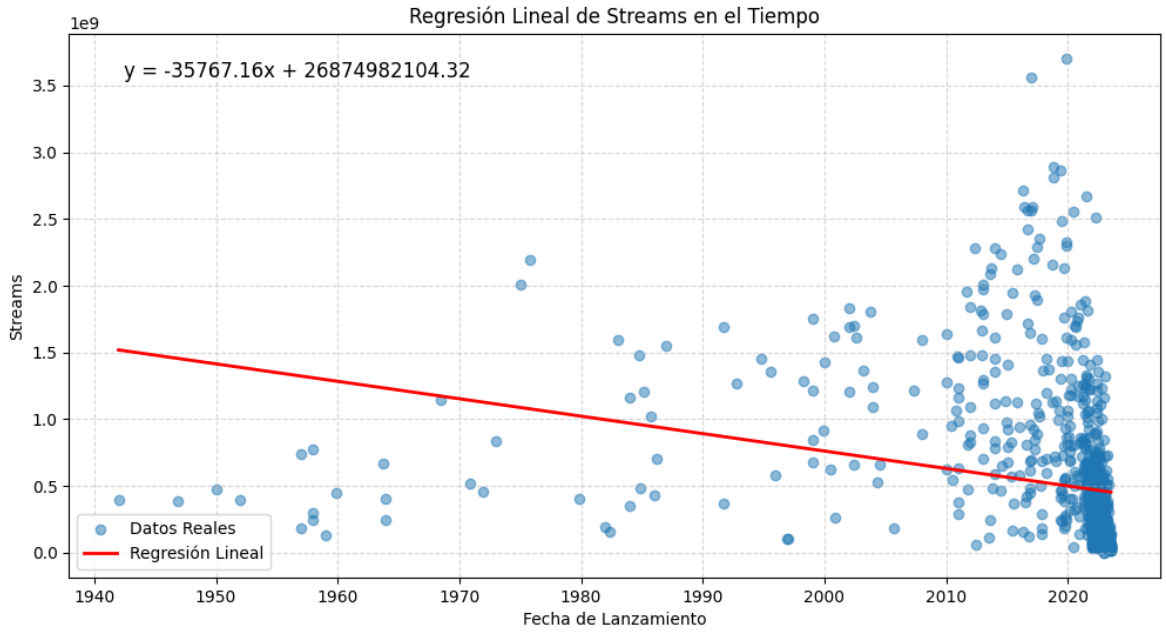

Figura 12. Demostración gráfica de la predicción en la regresión lineal.

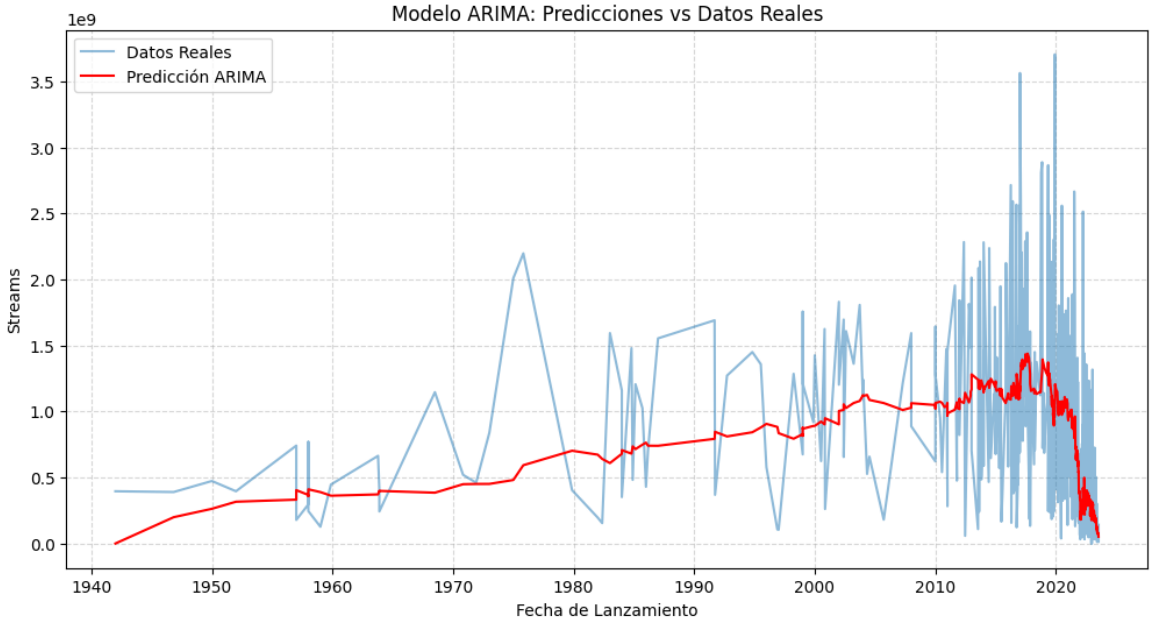

Figura 13. Demostración gráfica de la predicción en el modelo ARIMA.

b. Modela la ecuación de regresión del dataset.

Regresión Lineal Simple:

Ecuación: y = -35767.16x + 26874982104.32

Donde:

- y representa los streams.
- x representa la fecha en formato ordinal.

Regresión Lineal con Transformación Logarítmica:

Ecuación: y = -0.00x + 78.01

Donde:

- y representa el logaritmo de los streams.
- x representa la fecha en formato ordinal.

c. Aplica los modelos de series de tiempo autorregresivos y el modelo ARIMA.

Primero realizamos un análisis de autocorrelación (ACF) y autocorrelación parcial (PACF) para determinar los parámetros del modelo ARIMA.

Donde:

- Se define un modelo ARIMA con los valores obtenidos de p=1, d=1 y q=1.
  - ARIMA (1,1,1)
- Luego, se ajusta el modelo a los datos y se muestra un resumen de los resultados.
- Se generan predicciones dentro del conjunto de datos utilizando el modelo ARIMA.
- Finalmente, se grafica los datos reales y las predicciones del modelo ARIMA para visualizar el ajuste del modelo.
- Se utiliza la función auto_arima para encontrar los mejores parámetros automáticamente.
  - El mejor modelo encontrado es ARIMA (0,1,2).
- Se define un modelo ARIMA con los parámetros encontrados automáticamente.
- Se ajusta el modelo y se muestran los resultados.
- Se generan las predicciones y se grafican junto con los datos reales.

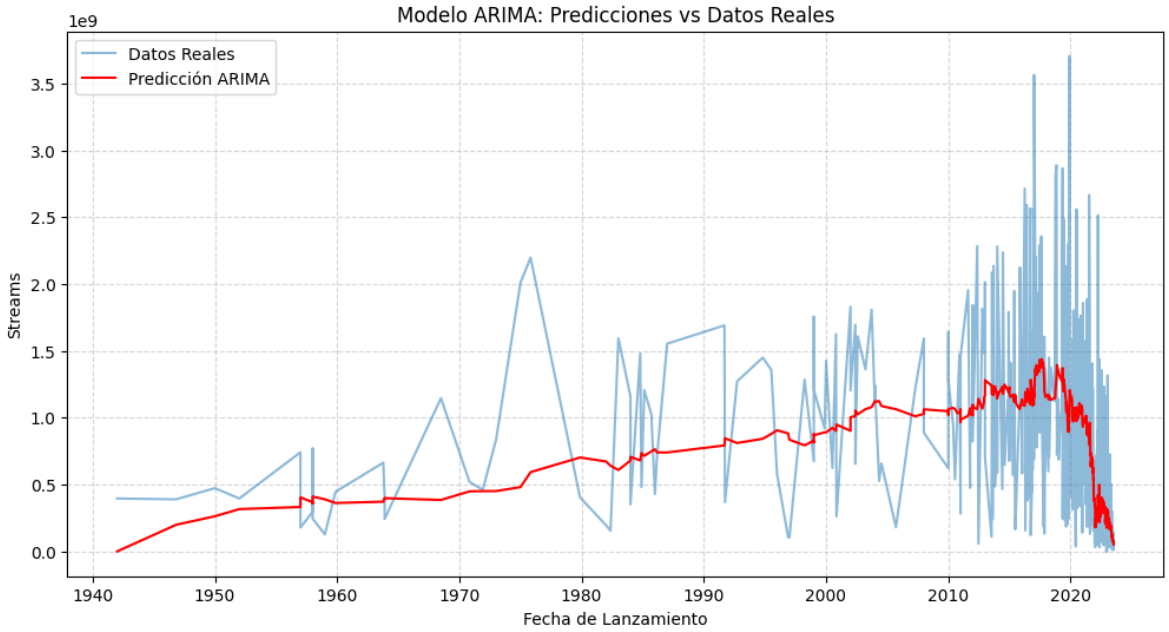

Figura 14. Demostración gráfica de la predicción en el modelo ARIMA (1,1,1).

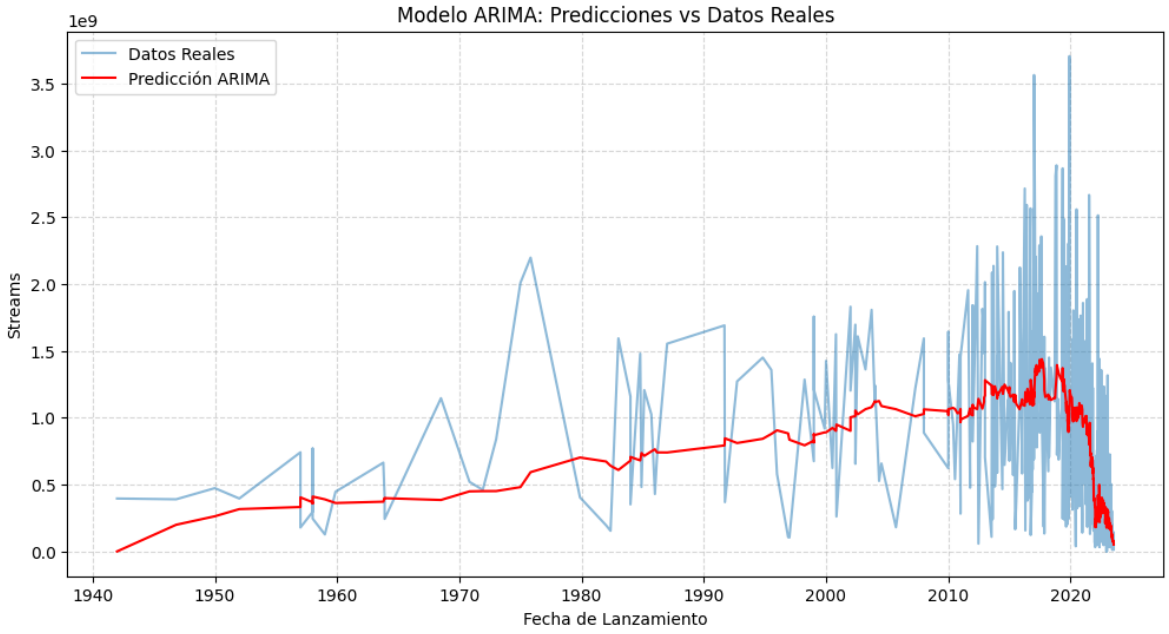

Figura 15. Demostración gráfica de la predicción en el modelo ARIMA (0,1,2).

Además, se realiza un análisis de estacionalidad utilizando la descomposición de series de tiempo y se calculó el error cuadrático medio (RMSE) el cual nos dio un valor de:

Error RMSE del modelo ARIMA: 429810488.4162128

Conclusiones por integrante

Aguilar Ramírez Carlos Francisco

El análisis realizado en esta práctica refleja la importancia de seleccionar modelos adecuados para el estudio de series de tiempo, especialmente en el contexto de la industria musical y el comportamiento de las canciones en plataformas de streaming. A lo largo del trabajo, se evidenció que la regresión lineal, aunque útil para identificar tendencias generales, tiene limitaciones cuando se enfrenta a datos temporales con autocorrelación y estacionalidad.

El modelo ARIMA, por su parte, permitió capturar patrones más complejos en la evolución de los streams de las canciones, aunque con ciertas limitaciones en su ajuste a los datos reales. Esto resalta la necesidad de seguir explorando enfoques más sofisticados, como modelos que incorporen factores exógenos o técnicas más avanzadas de aprendizaje automático para mejorar la precisión de las predicciones.

En términos generales, esta práctica no solo sirvió para aplicar herramientas estadísticas, sino también para reflexionar sobre la complejidad del análisis de series de tiempo en un contexto real. La predicción del éxito de una canción sigue siendo un desafío abierto, influenciado por múltiples factores externos, por lo que el desarrollo de modelos más robustos y adaptativos será clave en futuras investigaciones.

Arista Romero Juan Ismael

En esta práctica, analizamos una serie de tiempo basada en los streams de canciones en Spotify, aplicando técnicas de Regresión Lineal y ARIMA para modelar su comportamiento. Al final, pudimos concluir que la Regresión Lineal puede capturar tendencias generales, pero no es suficiente para modelar datos con fluctuaciones temporales complejas. Por otro lado, el modelo ARIMA(0,1,2) permitió capturar mejor la estructura temporal de la serie, aunque el error RMSE fue alto, lo que sugiere que los datos presentan un comportamiento difícil de predecir con métodos tradicionales.

Jiménez Flores Luis Arturo

En esta práctica utilizamos la regresión lineal y la comparamos con el método ARIMA, teniendo como contexto las series de tiempo. En primera instancia, noté que la aplicación de la regresión lineal en series de tiempo requiere consideraciones especiales debido a la dependencia temporal, la autocorrelación y la estacionariedad que difieren del uso estándar de la regresión lineal. Es por ello que en esta práctica desarrollamos y presentamos dos modelos de regresión: uno lineal simple para ver como era su comportamiento con respecto a los datos y otro con transformación logarítmica para conseguir una aproximación de la forma funcional de los datos si es que no se tiene una relación lineal, que podría haber sido el caso.

Ambos modelos proporcionan una representación matemática de la relación entre la fecha y los streams, aunque con diferentes escalas y ajustes que se pueden ver en la sección “propuesta de solución”.

Por otra parte, implementamos el modelo ARIMA para capturar la dinámica temporal de los datos. Donde se realizó un análisis de ACF y PACF para la identificación de los parámetros del modelo, además, este se ajustó a los datos, para generar predicciones.

Algunas cosas que identifiqué. Son que, los resultados del modelo ARIMA mostró limitaciones en su capacidad para ajustarse a los datos reales, lo que sugiere, que debería probarse otro modelo u otro enfoque que pueda captar mejor las características de la serie de tiempo, como la estacionalidad. Además, también hicimos un análisis de estacionalidad para investigar la presencia de patrones estacionales en los datos, lo que puede dar pie al uso de otro modelo.

Vazquez Martin Marlene Gabriela

En esta práctica, analizamos el comportamiento de series de tiempo en el contexto de plataformas de streaming, aplicando modelos de Regresión Lineal y ARIMA para identificar patrones y realizar predicciones. A lo largo del proceso, pude observar que la Regresión Lineal, aunque sencilla de interpretar y útil para detectar tendencias generales, presenta limitaciones significativas cuando se enfrenta a datos con autocorrelación y variaciones estacionales.

Por otro lado, la implementación del modelo ARIMA permitió capturar mejor la estructura temporal de la serie de datos, aunque no de manera óptima en todos los casos. A través del análisis de ACF y PACF, logramos ajustar los parámetros del modelo, pero los errores obtenidos evidenciaron que el comportamiento de los datos es complejo y que podrían explorarse técnicas más avanzadas, como modelos híbridos o de aprendizaje profundo, para mejorar la precisión de las predicciones.

En general, esta práctica no solo me permitió aplicar herramientas estadísticas para el análisis de series de tiempo, sino también reflexionar sobre los desafíos que implica modelar fenómenos reales con múltiples factores influyentes. La predicción del comportamiento de una canción en plataformas digitales sigue siendo un reto abierto, lo que motiva a seguir explorando enfoques más sofisticados en futuras investigaciones.