**Universidad Tecnológica de Panamá**

Maestría en Analítica de Datos

**Curso:**

Modelos Predictivos

**Avance de Proyecto Final**

**Análisis de precio de la onza de oro, estudio desde 2014 al 2024**

**Profesor:**

Juan Marcos Castillo, PhD

**Elaborado por:**

Edison Reyes Cedeño, 2-708-740

2025

# Introducción

El presente informe tiene como objetivo principal analizar y comprender la evolución histórica del precio de la onza de oro desde enero de 2014 hasta diciembre de 2024. Para ello, se abordará un estudio detallado de esta serie temporal, con el fin de identificar patrones, tendencias y fluctuaciones que han caracterizado este mercado a lo largo de una década. Además de su valor intrínseco, el oro desempeña un papel fundamental como inversión refugio en períodos de incertidumbre económica, convirtiéndose en una opción clave para proteger el capital frente a la volatilidad de los mercados financieros.

Como parte del análisis, se procederá a realizar una predicción a corto plazo empleando modelos de series temporales: Winters, ARIMA, SARIMA y Prophet. Concretamente se van a comparar estos 4 modelos utilizando herramientas en Python para alcanzar este cometido.

# Justificación

El oro ha sido históricamente un refugio de valor en tiempos de incertidumbre económica. Es también por tal motivo que se quiere ver cómo impactaron los eventos geopolíticos, la pandemia y crisis financieras a lo largo de esos años y que tan bien puede adaptarse el modelo para previsiones futuras.

Un análisis del comportamiento del precio del oro nos proporciona elementos para:

* Entender patrones históricos y su relación con los mercados financieros
* Modelar tendencias y estacionalidades que ayudan a prever su comportamiento hacia futuro
* Servir de guía en la toma de decisiones relacionadas con inversiones e incluso de política monetaria.

# Antecedentes

Los modelos ARIMA y SARIMA han sido utilizados ampliamente para identificar patrones lineales de series temporales financieras, incluyendo metales preciosos.

El modelo Holt-Winters (Triple Exponential Smoothing) ha sido útil en contextos donde existen tendencias y estacionalidad, como en las investigaciones de Atsalakis y Valavanis (2009) aplicadas a materias primas.

Últimamente se aplican modelos como Prophet (Facebook) o redes neuronales (LSTM) principalmente para captar no linealidades o efectos de eventos externos.

Este trabajo busca realizar una comparación de los modelos Winters, ARIMA, SARIMA y Prophet para ver qué modelo se adapta mejor y así poder realizar predicciones a futuro.

# Definición del problema

Con los datos históricos de los precios de la onza de oro entre enero de 2014 y diciembre de 2024 se aplicará los modelos de Winters, ARIMA, SARIMA y Prophet con la ayuda de herramientas disponible en lenguaje Python.

Se pretende evaluar y comparar el desempeño de los modelos e identificar patrones de tendencia y estacionalidad, así como también evaluarlos según las métricas de error.

Se realizará una proyección a 60 días y se evaluará el rendimiento de cada uno de los modelos.

# Análisis Predictivo

## Determinación de la base de datos

Los datos utilizados en este estudio fueron obtenidos directamente de la plataforma financiera https://www.investing.com, reconocida por su amplia cobertura y precisión en información financiera. Estos datos presentan una notable consistencia a lo largo del periodo analizado, lo que garantiza su fiabilidad para los propósitos de esta investigación.

La base de datos incluye los precios diarios de la onza de oro, recopilados en días hábiles, abarcando desde enero de 2014 hasta diciembre de 2024, lo que permite realizar un análisis exhaustivo de las tendencias y fluctuaciones del mercado en dicho intervalo temporal.

## Pre-procesamiento y limpieza

Este conjunto de datos presenta datos consistentes, por ello fue necesario realizar apenas algunas adecuaciones como:

* Para trabajar con las series de tiempo en lugar de utilizar el índice numérico y consecutivo del dataframe, se realizó el cambio tomando el índice como la fecha (variable Date)
* Se transformó la variable Price a float con el fin de realizar los cálculos pertinentes, ya que venían en formato no numérico

## Análisis descriptivo

Se realiza un breve análisis exploratorio EDA con el fin de obtener una idea general de los datos a ser manipulados.

### **Inspección del dataframe, primeros registros**

**# Inspección de datos del dataframe, primeras 5 filas**

df\_gold.head()

|  | **Date** | **Price** | **Open** | **High** | **Low** | **Vol.** | **Change %** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 12/31/2024 | 2,641.00 | 2,620.00 | 2,642.00 | 2,614.30 | 86.29K | 0.87% |
| **1** | 12/30/2024 | 2,618.10 | 2,636.10 | 2,640.70 | 2,608.40 | 110.92K | -0.01% |
| **2** | 12/27/2024 | 2,618.40 | 2,639.70 | 2,639.70 | 2,612.10 | 0.64K | -0.83% |
| **3** | 12/26/2024 | 2,640.30 | 2,623.30 | 2,642.40 | 2,622.80 | 1.40K | 0.67% |
| **4** | 12/24/2024 | 2,622.80 | 2,616.50 | 2,622.70 | 2,612.50 | 0.42K | 0.28% |

### **Números de registros y descripción de variables:**

### Dataframe original

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 2816 entries, 0 to 2815

Data columns (total 7 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Date 2816 non-null object

1 Price 2816 non-null object

2 Open 2816 non-null object

3 High 2816 non-null object

4 Low 2816 non-null object

5 Vol. 2814 non-null object

6 Change % 2816 non-null object

dtypes: object(7)

memory usage: 154.1+ KB

### **Valores nulos**

df\_gold.isnull().sum()

|  | **0** |
| --- | --- |
| **Price** | 0 |

**dtype:** int64

No se cuentan con valores nulos en esta serie de tiempo.

### **Estadísticas descriptivas**

Df\_gold.describe()

|  | **Price** |
| --- | --- |
| **count** | 2816.000000 |
| **mean** | 1576.401669 |
| **std** | 387.842859 |
| **min** | 1049.600000 |
| **25%** | 1256.750000 |
| **50%** | 1417.550000 |
| **75%** | 1850.275000 |
| **max** | 2800.800000 |

Podemos distinguir la cantidad de registros, el promedio en USD $1576.40, el valor mínimo ubicándose en los USD $1049.60 y el máximo en USD $2800.80

**Distribución de frecuencias**

Se realizó un análisis de distribución de frecuencia para ver en qué intervalos se agrupan los precios del oro. Se observa claramente que en esos 11 años la mayoría de las observaciones se agrupan alrededor de los USD $1250 y los USD $1750A graph of a number of blue bars

Description automatically generated with medium confidence

## Selección de variables

En el marco de esta investigación, se decidió trabajar exclusivamente con las variables 'Date' y 'Price'. La variable 'Date' permitió estructurar el análisis en torno a una referencia temporal precisa, mientras que 'Price' sirvió como indicador principal para evaluar las variaciones y tendencias dentro del periodo estudiado.

Se optó también por reemplazar el índice numérico y consecutivo del dataframe, asignando la fecha (variable Date) como nuevo índice.

**# Selección de las columnas que se van a utilizar**

df\_gold = df\_gold[['Date', 'Price']].copy()

df\_gold['Date'] = pd.to\_datetime(df\_gold['Date'])

df\_gold.set\_index('Date', inplace=True)

**#Transformación de columna numérica a float y eliminar la coma que venía** df\_gold['Price'] = df\_gold['Price'].replace({',':''},regex=True).astype(float)

## Selección de Modelos

Con base en los antecedentes que destacan el uso de los modelos ARIMA y SARIMA como herramientas efectivas para analizar series de tiempo financieras y de materias primas, así como la aplicación del modelo Holt-Winters para capturar patrones estacionales, esta investigación evaluará el desempeño de estos tres modelos junto al modelo Facebook Prophet. Este último ofrece la ventaja de incluir factores externos como variables adicionales para enriquecer el análisis. El estudio utilizará datos históricos del precio del oro, abarcando un periodo suficientemente amplio para permitir un análisis completo y las comparaciones entre los cuatro modelos desde distintas perspectivas. En las siguientes secciones, se presentará una descripción de cada uno de los modelos utilizados, seguida de un análisis de los resultados obtenidos.

### Modelo Winters

El modelo Holt-Winters, también conocido como el método de suavizado exponencial triple, es una técnica utilizada para analizar y predecir series de tiempo que presentan componentes estacionales, tendencia y nivel base. Este modelo se basa en descomponer la serie en tres elementos principales:

- Nivel base: Representa el valor promedio de la serie en el tiempo.

- Tendencia: Captura cambios consistentes a largo plazo, ya sea ascendentes o descendentes.

- Estacionalidad: Identifica patrones repetitivos en intervalos regulares.

Hay dos versiones principales de Holt-Winters: aditiva (para series con variaciones estacionales de magnitud constante) y multiplicativa (para series con variaciones estacionales proporcionales al nivel base). Gracias a su versatilidad, es muy útil para prever valores futuros en datos que cambian con el tiempo y tienen patrones estacionales.

Para llevar a cabo el análisis del dataframe, se utilizó la librería **statsmodels.tsa.holtwinters** disponible en Python. Esta herramienta permite aplicar el modelo Holt-Winters para series de tiempo, facilitando la descomposición de los datos en sus componentes

**# Modelo Winters**

modelo\_winters = ExponentialSmoothing(

df\_gold['Price'],

trend='add',

seasonal='mul',

seasonal\_periods=260,

initialization\_method="legacy-heuristic"

).fit()

modelo\_winters.summary()

Tabla resumen – Modelo Winters

|  |  |  |  |
| --- | --- | --- | --- |
| Dep. Variable: | Price | No. Observations: | 2869 |
| Model: | ExponentialSmoothing | **SSE** | 908543.627 |
| Optimized: | True | **AIC** | 17047.356 |
| Trend: | Additive | **BIC** | 18621.250 |
| Seasonal: | Multiplicative | **AICC** | 17101.946 |
| Seasonal Periods: | 260 | **Date:** | Mon, 07 Apr 2025 |
| Box-Cox: | False | **Time:** | 03:55:51 |
| Box-Cox Coeff.: | None |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | coeff | code | optimized |
| smoothing\_level | 0.8535714 | alpha | True |
| smoothing\_trend | 0.0001 | beta | True |
| smoothing\_seasonal | 0.1464286 | gamma | True |

Gráfico:

Predicción del precio de la onza de Oro - Modelo Winters

A graph with blue and orange lines

Description automatically generated

### Modelo ARIMA

El modelo ARIMA (AutoRegressive Integrated Moving Average) es una técnica ampliamente utilizada para analizar y predecir series de tiempo. Combina tres componentes clave:

- **AR (AutoRegressive)**: Modela la relación entre una variable y sus valores pasados, basándose en observaciones anteriores.

- **I (Integrated)**: Introduce una diferenciación para convertir una serie no estacionaria en estacionaria.

- **MA (Moving Average)**: Representa la dependencia de la variable con los errores de predicción de observaciones previas.

El ARIMA es especialmente eficaz para datos que no presentan estacionalidad.

Para llevar a cabo el análisis del dataframe, se utilizó la librería **statsmodels.tsa.arima.model** disponible en Python.

**#--- Modelo ARIMA(1,1,1)**

modelo\_arima = ARIMA(df\_gold['Price'], order=(1,1,1)).fit()

modelo\_arima.summary()

Tablas resumen – Modelo ARIMA

|  |  |  |  |
| --- | --- | --- | --- |
| Dep. Variable: | Price | No. Observations: | 2869 |
| Model: | ARIMA(1, 1, 1) | **Log Likelihood** | -11859.944 |
| Date: | Mon, 07 Apr 2025 | **AIC** | 23725.889 |
| Time: | 03:55:54 | **BIC** | 23743.773 |
| Sample: | 01-02-2014 - 12-31-2024 | **HQIC** | 23732.336 |
|  |  |  |  |
| Covariance Type: | opg |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **z** | **P>|z|** | **[0.025** | **0.975]** |
| **ar.L1** | -0.0107 | 0.336 | -0.032 | 0.975 | -0.670 | 0.649 |
| **ma.L1** | -0.0306 | 0.334 | -0.091 | 0.927 | -0.686 | 0.625 |
| **sigma2** | 228.9802 | 3.232 | 70.841 | 0.000 | 222.645 | 235.315 |

|  |  |  |  |
| --- | --- | --- | --- |
| Ljung-Box (L1) (Q): | 0.00 | Jarque-Bera (JB): | 3211.84 |
| Prob(Q): | 0.95 | **Prob(JB):** | 0.00 |
| Heteroskedasticity (H): | 2.98 | **Skew:** | -0.42 |
| Prob(H) (two-sided): | 0.00 | **Kurtosis:** | 8.11 |

Gráfico:

Predicción del precio de la onza de Oro - Modelo ARIMA

A graph with lines and dots

Description automatically generated with medium confidence

### Modelo SARIMA

El modelo SARIMA (Seasonal AutoRegressive Integrated Moving Average) es una extensión del modelo ARIMA diseñada para manejar series de tiempo con patrones estacionales. Incorpora componentes adicionales para capturar las fluctuaciones repetitivas dentro de intervalos regulares. SARIMA se define por:

1. **Parte ARIMA**: Maneja la tendencia y los movimientos aleatorios de la serie, utilizando componentes autorregresivos (AR), de diferenciación integrada (I) y promedio móvil (MA).
2. **Parte estacional**: Añade parámetros para modelar la estacionalidad, que incluyen autorregresión, diferenciación y promedios móviles estacionales.

Es especialmente útil para datos con ciclos que se repiten en intervalos específicos, como ventas mensuales o temperaturas estacionales. Su capacidad para integrar estacionalidad lo convierte en una opción robusta para análisis predictivos complejos.

Para llevar a cabo el análisis del dataframe, se utilizó la librería **statsmodels.tsa.statespace.sarimax** disponible en Python.

**#--- Modelo SARIMA (1, 1, 1)x(1, 0, 1, 5)**

modelo\_sarima = SARIMAX(df\_gold['Price'],

order=(1,1,1),

seasonal\_order=(1,0,1,5),

enforce\_stationarity=False,

enforce\_invertibility=False).fit()

modelo\_sarima.summary()

Tablas resumen – Modelo SARIMA

|  |  |  |  |
| --- | --- | --- | --- |
| Dep. Variable: | Price | No. Observations: | 2869 |
| Model: | SARIMAX(1, 1, 1)x(1, 0, 1, 5) | **Log Likelihood** | -11832.389 |
| Date: | Mon, 07 Apr 2025 | **AIC** | 23674.778 |
| Time: | 03:55:55 | **BIC** | 23704.573 |
| Sample: | 01-02-2014 | **HQIC** | 23685.521 |
|  | - 12-31-2024 |  |  |
| Covariance Type: | opg |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | coef | std err | z | P>|z| | [0.025 | 0.975] |
| ar.L1 | -0.0237 | 0.324 | -0.073 | 0.942 | -0.660 | 0.612 |
| ma.L1 | -0.0191 | 0.323 | -0.059 | 0.953 | -0.652 | 0.613 |
| ar.S.L5 | 0.4242 | 0.473 | 0.896 | 0.370 | -0.503 | 1.352 |
| ma.S.L5 | -0.4441 | 0.469 | -0.947 | 0.344 | -1.363 | 0.475 |
| sigma2 | 228.8427 | 3.253 | 70.353 | 0.000 | 222.467 | 235.218 |

|  |  |  |  |
| --- | --- | --- | --- |
| Ljung-Box (L1) (Q): | 0.00 | Jarque-Bera (JB): | 3175.58 |
| Prob(Q): | 0.95 | **Prob(JB):** | 0.00 |
| Heteroskedasticity (H): | 2.99 | **Skew:** | -0.41 |
| Prob(H) (two-sided): | 0.00 | **Kurtosis:** | 8.10 |

#### Gráfico:

#### Predicción del precio de la onza de Oro - Modelo SARIMA

#### A graph with lines and numbers Description automatically generated with medium confidence

### Modelo Facebook Prophet

Facebook Prophet es un modelo de series de tiempo diseñado para manejar datos con tendencias no lineales y patrones estacionales. Es especialmente útil para trabajar con datos que tienen irregularidades, valores atípicos o intervalos faltantes. Una de sus ventajas clave es que permite incorporar fácilmente factores externos o "regresores" para enriquecer las predicciones. Además, Prophet ofrece una interfaz simple y flexible que lo hace accesible tanto para principiantes como para expertos en análisis de series de tiempo.

Para llevar a cabo el análisis del dataframe, se utilizó la librería **prophet** disponible en Python.

Gráfico:

Modelo Facebook Prophet

A graph with blue and black lines

Description automatically generated

### Métricas de error

Para cada modelo se calcularon las principales métricas de error para evaluar en rendimiento y la confiabilidad del modelo.

MSE (Mean Square Error)

MAD (Mean Absolute Desviation)

MAPE (Mean Absolute Percentage Error)

Desviación Estándar

Se presenta una tabla resumen con los hallazgos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Método** | **MSE** | **MAD** | **MAPE %** | **Desv. Est** |
| Winters | 316.68 | 12.36 | 0.77 | 15.45 |
| ARIMA | 751.90 | 10.74 | 0.68 | 13.43 |
| SARIMA | 774.65 | 10.83 | 10.83 | 13.53 |
| Prophet | 3855.11 | 45.46 | 45.46 | 56.82 |

Con un MSE de 316 el Winters es el modelo que mejor se adapta, captando de norma precisa el comportamiento de la serie. Si bien es cierto el MAD, MAPE y la desviación estándar están un poquito arriba que los modelos ARIMA y SARIMA, el MSE de USD $300 donde los valores actuales están rondando los USD $1900, sería un comportamiento aceptable.

Los valores MSE para ARIMA y SARIMA si están bastante alejados, lo cual podría representar hasta casi un 30% del valor real, no nos da mucha precisión. Su adaptabilidad para los valores ajustados, siempre y cuando se tomen en cuenta el MSE, es buena.

Los indicadores del modelo Prophet no son aceptables por su alta incertidumbre notada en la gráfica también. Es necesario realizar una mejor parametrización para evaluar si el rendimiento mejora.

# Coclusiones

Después de haber analizado una serie temporal de los datos históricos del precio de la onza de oro bajo 4 modelos predictivos, se concluye que cada modelo presenta sus bondades y desventajas.

Para este caso, el modelo Winters fue el modelo más preciso, fue excelente capturando la estacionalidad y la tendencia, aún cuando la estacionalidad era muy leve, denotando lo bueno que se ajustó.

Los modelos ARIMA y SARIMA replicaron bien los datos basados en los datos históricos, más para proyectar una predicción no se obtuvieron los resultados deseados, aún cuando el SARIMA incluye el componente estacionalidad.

El modelo Facebook Prophet, al ser evaluado por las métricas de error, resultó no adecuado. Esto se observó en la gráfica donde había variaciones considerables entre los datos ajustados y los datos reales.

# Recomendaciones

Llevar el modelo Winters a periódos de tiempo más largos, por ejemplo, para detectar ciclos largos del precio del oro. Se sabe que en estos últimos años la subida está moldeada por la Pandemia y otros aspectos geopolíticos.

Buscar ajustes para los modelos ARIMA y SARIMA para lograr captar el componente predictivo a futuro.

Tomando en cuenta las bondades del modelo Facebook Prophet para incluir variables externas, buscar un mejor ajuste y adaptarlo por ejemplo a los últimos acontecimientos de la geopolítica.

# Bibliografía

Chopra, S., & Meindl, P. (2019). *Administración de la cadena de suministro: Estrategia, planeación y operación* (5ta ed.). Pearson.

<https://www.investing.com>

<https://www.statsmodels.org/stable/index.html>

# Anexos

El código de la aplicación de los modelos se encuentra en el repositorio:

<https://github.com/edreyesc/ProyectoFinal/>