

**UNIVERSIDAD NACIONAL DE CAJAMARCA**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE MINAS**



**TESIS**

**DESARROLLO DE UN MODELO PREDICTIVO PARA LA  
FLUCTUACIÓN DEL PRECIO DEL ORO UTILIZANDO  
MECANISMOS DE MACHINE LEARNING**

**Para optar el Título Profesional de:**

Ingeniero de Minas

**Presentado por:**

Bach. Manosalva Horna Heyler Gustavo

**Asesor:**

M.Cs. Ing. Arapa Vilca Víctor Ausberto

Cajamarca - Perú

2025

## CONSTANCIA DE INFORME DE ORIGINALIDAD

### - FACULTAD DE INGENIERÍA -

- Investigador** : MANOSALVA HORNA HEYLER GUSTAVO  
**DNI** : 71492181  
**Escuela Profesional** : INGENIERÍA DE MINAS
- Asesor** : M. CS. ING. ARAPA VILCA VICTOR AUSBERTO  
**Facultad** : INGENIERÍA
- Grado académico o título profesional**  
 Bachiller  Título profesional  Segunda especialidad  
 Maestro  Doctor
- Tipo de Investigación:**  
 Tesis  Trabajo de investigación  Trabajo de suficiencia profesional
- Título de Trabajo de Investigación:**  
"DESARROLLO DE UN MODELO PREDICTIVO PARA LA FLUCTUACIÓN DEL PRECIO DEL ORO UTILIZANDO MECANISMOS DE MACHINE LEARNING"
- Fecha de evaluación:** 18 DE NOVIEMBRE DE 2024
- Software antiplagio:**  TURNITIN  URKUND (OURIGINAL) (\*)
- Porcentaje de Informe de Similitud:** 1%
- Código Documento:** oid:3117:406893152
- Resultado de la Evaluación de Similitud:**  
 APROBADO  PARA LEVANTAMIENTO DE OBSERVACIONES O DESAPROBADO

Fecha Emisión:22/11/2024

		Firmado digitalmente por: BAZAN DIAZ Laura Sofia FAU 20148258601 soft Motivo: En señal de conformidad Fecha: 22/11/2024 09:23:57-0500
<hr/> <b>FIRMA DEL ASESOR</b> <b>M. CS. ING. ARAPA VILCA VICTOR AUSBERTO</b> <b>DNI: 29552145</b>	<hr/> <b>UNIDAD DE INVESTIGACIÓN FI</b>	

\* En caso se realizó la evaluación hasta setiembre de 2023



## ACTA DE SUSTENTACIÓN PÚBLICA DE TESIS.

TITULO : "DESARROLLO DE UN MODELO PREDICTIVO PARA LA FLUCTUACIÓN DEL PRECIO DEL ORO UTILIZANDO MECANISMOS DE MACHINE LEARNING."

ASESOR : M.Cs. Ing. Víctor Ausberto Arapa Vilca.

En la ciudad de Cajamarca, dando cumplimiento a lo dispuesto por el Oficio Múltiple N° 0048-2025-PUB-SA-FI-UNC, de fecha 16 de enero de 2025, de la Secretaría Académica de la Facultad de Ingeniería, a los **siete días del mes de febrero de 2025**, siendo las nueve horas (09:00 a.m.) en el Auditorio de la Escuela Profesional de Ingeniería Geológica (Ambiente 4J - 210), se reunieron los Señores Miembros del Jurado Evaluador:

Presidente : Dr. Ing. Segundo Reinaldo Rodríguez Cruzado.  
Vocal : Dr. Ing. Crispín Zenón Quispe Mamani.  
Secretario : M.Cs. Ing. Roberto Severino Gonzales Yana.

Para proceder a escuchar y evaluar la sustentación pública de la tesis titulada "DESARROLLO DE UN MODELO PREDICTIVO PARA LA FLUCTUACIÓN DEL PRECIO DEL ORO UTILIZANDO MECANISMOS DE MACHINE LEARNING", presentado por el Bachiller en Ingeniería de Minas HEYLER GUSTAVO MANOSALVA HORNA, asesorado por el M.Cs. Ing. Víctor Ausberto Arapa Vilca, para la obtención del Título Profesional

Los Señores Miembros del Jurado replicaron al sustentante debatieron entre sí en forma libre y reservada y lo evaluaron de la siguiente manera:

EVALUACIÓN PRIVADA : 03:00 PTS.  
EVALUACIÓN PÚBLICA : 10:00 PTS.  
EVALUACIÓN FINAL : 13:00 PTS

TRECE (En letras)

En consecuencia, se lo declara APROBADO con el calificativo de TRECE acto seguido, el presidente del jurado hizo saber el resultado de la sustentación, levantándose la presente a las 10:00 horas del mismo día, con lo cual se dio por terminado el acto, para constancia se firmó por quintuplicado.

Dr. Ing. Segundo Reinaldo Rodríguez Cruzado.  
Presidente

Dr. Ing. Crispín Zenón Quispe Mamani.  
Vocal

M.Cs. Ing. Roberto Severino Gonzales Yana.  
Secretario

M.Cs. Ing. Víctor Ausberto Arapa Vilca.  
Asesor



### EVALUACIÓN DE LA SUSTENTACIÓN PÚBLICA DE TESIS.

Bachiller en Ingeniería de Minas: HEYLER GUSTAVO MANOSALVA HORNA.

RUBRO	PUNTAJE
	Máximo/Calificación
<b>2. DE LA SUSTENTACIÓN PÚBLICA</b>	
2.1. Capacidad de síntesis	4.0
2.2. Dominio del tema	4.0
2.3. Consistencia de las alternativas presentadas	1.0
2.4. Precisión y seguridad en las respuestas	1.0
<b>PUNTAJE TOTAL (MÁXIMO 12 PUNTOS)</b>	<b>10.0</b>

Cajamarca, 07 de febrero de 2025

Dr. Ing. Segundo Reinaldo Rodríguez Cruzado.  
Presidente

Dr. Ing. Crispin Zepón Quispe Mamani.  
Vocal

M.Cs. Ing. Roberto Severino Gonzales Yana.  
Secretario

M.Cs. Ing. Víctor Ausberto Arapa Vilca.  
Asesor



#### EVALUACIÓN FINAL DE LA SUSTENTACIÓN DE TESIS.

Bachiller en Ingeniería de Minas: HEYLER GUSTAVO MANOSALVA HORNA.

RUBRO	PUNTAJE
A.- EVALUACIÓN DE LA SUSTENTACIÓN PRIVADA	03.00
B.- EVALUACIÓN DE LA SUSTENTACIÓN PÚBLICA	10.00
<b>EVALUACIÓN FINAL</b>	
EN NÚMEROS ( A + B )	13.00
EN LETRAS ( A + B )	TRECE
- Excelente 20 - 19	REGULAR
- Muy Bueno 18 - 17	
- Bueno 16 - 14	
- Regular 13 a 11	
- Desaprobado 10 a menos	

Cajamarca, 07 de febrero de 2025

Dr. Ing. Segundo Reinaldo Rodríguez Cruzado.  
Presidente

Dr. Ing. Crispin Zenón Quispe Mamani.  
Vocal

M.Cs. Ing. Roberto Severino Gonzales Yana.  
Secretario

M.Cs. Ing. Víctor Ausberto Arapa Vilca.  
Asesor

## **AGRADECIMIENTO**

A DIOS, por ser mi fuerza y guía en todo momento.

A la Escuela Profesional de Ingeniería de Minas de la Universidad Nacional de Cajamarca, por brindarme un ambiente adecuado para crecer y aprender.

A mi asesor, M.Cs. Víctor Arapa Vilca por su ayuda y orientación, para la realización de este trabajo.

A cada persona que ha contribuido de manera significativa a esta investigación.

## **DEDICATORIA**

Esta tesis es un emotivo tributo a mi madre Alicia, cuyo apoyo constante y amor han sido la clave de mi éxito. Cada logro alcanzado lleva consigo un fragmento de su dedicación. Su valiente ejemplo y sacrificio han sido mi inspiración constante; en momentos de duda, sus palabras alentadoras me han impulsado a seguir adelante. Mi gratitud hacia ella es inmensa, ya que todo lo que soy como profesional y persona es gracias a su amor incondicional y apoyo incansable. Espero que esta tesis la haga sentir tan orgullosa como yo me siento de tenerla como madre.

## ÍNDICE

	Pág.
AGRADECIMIENTO .....	i
DEDICATORIA.....	ii
ÍNDICE DE TABLAS.....	v
ÍNDICE DE FIGURAS.....	vii
LISTA DE ABREVIATURAS .....	viii
RESUMEN.....	x
ASBTRACT.....	xi

### CAPÍTULO I INTRODUCCIÓN

### CAPÍTULO II MARCO TEÓRICO

2.1. ANTECEDENTES TEÓRICOS.....	3
2.1.1. Internacionales.....	3
2.1.2. Nacionales.....	4
2.1.3. Locales .....	5
2.2. BASES TEÓRICAS.....	6
2.2.1. El Oro.....	6
2.2.1.1. Evolución del Precio del Oro .....	6
2.2.1.2. Mercado del Oro .....	9
2.2.2. Aprendizaje Automático (Machine Learning).....	12
2.2.2.1. Tipos de Aprendizaje Automático.....	13
2.2.3. Aspectos Clave en Modelos Predictivos.....	18
2.2.3.1. Algoritmos Relevantes de Regresión.....	18
2.2.3.2. Métricas de Evaluación en Modelos Predictivos.....	23
2.2.3.3. Problemas Comunes en Modelos Predictivos.....	25
2.2.3.4. Técnicas de Mejora de Modelos Predictivos .....	27
2.2. DEFINICIÓN DE TÉRMINOS BÁSICOS .....	32

### CAPÍTULO III MATERIALES Y MÉTODOS

3.1. UBICACIÓN DE LA INVESTIGACIÓN.....	34
3.2. METODOLOGÍA DE LA INVESTIGACIÓN.....	34
3.2.1. Tipo, Nivel, Diseño y Enfoque de Investigación.....	34
3.2.2. Población de Estudio .....	35
3.2.3. Muestra.....	35

	Pág.
3.2.4. Unidad de Análisis .....	35
3.2.5. Definición de Variables .....	36
3.2.5.1. Independientes .....	36
3.2.5.2. Dependientes .....	36
3.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS .....	36
3.3.1. Técnicas .....	36
3.3.2. Instrumentos .....	37
3.3.3. Materiales y Equipos .....	37
3.3.4. Softwares .....	37
3.4. PROCEDIMIENTOS .....	38
3.4.1. Elección del Método de Aprendizaje Automático.....	39
3.4.2. Recopilación de Datos.....	39
3.4.3. Análisis Exploratorio de Datos .....	40
3.4.4. Preprocesamiento de Datos.....	40
3.4.5. Elección de Algoritmos.....	41
3.4.6. Construcción de Modelo Predictivo .....	41
3.4.7. Elección del mejor Modelo Predictivo .....	41
3.4.8. Predicción .....	41
3.5. TRATAMIENTO, ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS .....	42

#### CAPÍTULO IV ANÁLISIS Y DISCUSIÓN DE RESULTADOS

4.1. ANÁLISIS DE RESULTADOS.....	84
4.1.1. Análisis de Ajuste de Hiperparámetros.....	85
4.1.2. Análisis de Selección de Variables Relevantes .....	87
4.1.3. Análisis de Selección del Mejor Algoritmo.....	90
4.2. CONTRASTACIÓN DE LA HIPÓTESIS .....	91

#### CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES .....	92
5.2. RECOMENDACIONES .....	93

REFERENCIAS BIBLIOGRÁFICAS.....	94
ANEXOS.....	98

## ÍNDICE DE TABLAS

	Pág.
Tabla 1. Métodos de Aprendizaje Automático .....	39
Tabla 2. Variables Recolectadas .....	43
Tabla 3. Clasificación de Principales Hiperparámetros según el tipo de Kernel.....	59
Tabla 4. Valores Aleatorios de los Principales Hiperparámetros del Algoritmo SVR.....	60
Tabla 5. Rango de Valores para los Principales Hiperparámetros del Algoritmo SVR .....	60
Tabla 6. Valores Óptimos de los Hiperparámetros del Algoritmo SVR.....	61
Tabla 7. Entrenamiento del Algoritmo SVR.....	62
Tabla 8. Valor de Métricas de Evaluación para el Algoritmo SVR.....	64
Tabla 9. Determinación del Número de Variables Óptimas para el Algoritmo SVR.....	65
Tabla 10. Subconjunto Óptimo de Variables para el Algoritmo SVR.....	66
Tabla 11. Métricas de Evaluación usando Subconjunto Óptimo de Variables con SVR .....	66
Tabla 12. Ranking de Variables de Entrada en el Algoritmo SVR .....	67
Tabla 13. Valores Aleatorios de los Principales Hiperparámetros del Algoritmo RFR .....	68
Tabla 14. Rango de Valores para los Principales Hiperparámetros del Algoritmo RFR .....	69
Tabla 15. Valores Óptimos de los Hiperparámetros del Algoritmo RFR.....	69
Tabla 16. Entrenamiento del Algoritmo RFR.....	70
Tabla 17. Valor de Métricas de Evaluación para el Algoritmo RFR.....	72
Tabla 18. Determinación del Número de Variables Óptimas para el Algoritmo RFR .....	73
Tabla 19. Subconjunto Óptimo de Variables para el Algoritmo RFR.....	74
Tabla 20. Métricas de Evaluación usando Subconjunto Óptimo de Variables con RFR .....	74
Tabla 21. Ranking de Variables de Entrada en el Algoritmo RFR.....	74
Tabla 22. Valores Aleatorios de los Principales Hiperparámetros del Algoritmo GBR .....	75
Tabla 23. Rango de Valores para los Principales Hiperparámetros del Algoritmo GBR.....	75
Tabla 24. Valores Óptimos de los Hiperparámetros del Algoritmo GBR .....	76
Tabla 25. Entrenamiento del Algoritmo GBR .....	77
Tabla 26. Valor de Métricas de Evaluación para el Algoritmo GBR .....	79
Tabla 27. Determinación del Número de Variables Óptimas para el Algoritmo GBR .....	80
Tabla 28. Subconjunto Óptimo de Variables para el Algoritmo GBR .....	80
Tabla 29. Métricas de Evaluación usando Subconjunto Óptimo de Variables con GBR.....	81
Tabla 30. Ranking de Variables de Entrada en el Algoritmo GBR.....	81
Tabla 31. Comparación de Valor de Métricas de Evaluación de Modelos Predictivos.....	82
Tabla 32. Predicción de Precio de Oro con Nuevos Datos .....	83

	Pág.
Tabla 33. Valores de Hiperparámetros Asignados a cada Algoritmo.....	85
Tabla 34. Comparación de Rendimiento de Algoritmos según el Tipo de Selección de Hiperparámetros.....	86
Tabla 35. Variables Óptimas Según el Algoritmo Utilizado .....	87
Tabla 36. Comparación de Rendimiento de Algoritmos en base al Ajuste de Hiperparámetros y Selección de Variables .....	89
Tabla 37. Evaluación de Errores de Predicción con Nuevos Datos.....	91

## ÍNDICE DE FIGURAS

	Pág.
Figura 1. Precio Histórico del Oro .....	8
Figura 2. Oferta Mundial del Oro .....	10
Figura 3. Demanda Mundial del Oro .....	11
Figura 4. Esquema de Aprendizaje Supervisado .....	14
Figura 5. Esquema de Aprendizaje No Supervisado .....	16
Figura 6. Esquema de Aprendizaje Reforzado .....	17
Figura 7. Representación del Algoritmo SVR .....	19
Figura 8. Esquema del Algoritmo Regresor de Bosques Aleatorios .....	21
Figura 9. Esquema del Algoritmo Regresor de Incremento Gradual.....	22
Figura 10. Ejemplo de Sobreajuste y Subajuste.....	26
Figura 11. Comportamiento del Error frente al Sobreajuste.....	27
Figura 12. Método de Retención.....	28
Figura 13. Método de la Validación Cruzada .....	29
Figura 14. Esquema de Método de RFE .....	31
Figura 15. Flujograma para el Desarrollo del Modelo Predictivo .....	38
Figura 16. Visualización de Datos .....	45
Figura 17. Número de Filas y Columnas .....	45
Figura 18. Tipo de Variables .....	46
Figura 19. Número de Valores Faltantes por Variable .....	47
Figura 20. Análisis Estadístico de Variables Numéricas .....	48
Figura 21. Histogramas de Variables de Estudio.....	50
Figura 22. Correlación de Variables de Estudio .....	51
Figura 23. Diagramas de Caja y Bigotes de las Variables de Estudio.....	54
Figura 24. Estandarización de Datos .....	55
Figura 25. Base de Datos con lags .....	56
Figura 26. División de Datos .....	57
Figura 27. División de Datos para el Desarrollo del Modelo Predictivo.....	57
Figura 28. Cálculo de Métricas de Evaluación para el Algoritmo SVR.....	63
Figura 29. Código Python para hallar la Importancia de Variables de Entrada en el SVR...67	
Figura 30. Cálculo de Métricas de Evaluación para el Algoritmo RFR .....	71
Figura 31. Cálculo de Métricas de Evaluación para el Algoritmo GBR .....	78
Figura 32. Comparación del Precio del Oro Real con el Precio del oro Predicho.....	84

## LISTA DE ABREVIATURAS

SVR	:	Regresión con Máquinas de Vectores de Soporte (Support Vector Regression)
SVM	:	Máquina de Vectores de Soporte (Support Vector Machine)
RFR	:	Regresor de Bosques Aleatorios (Random Forest Regressor)
GBR	:	Regresor de Gradiente Aumentado (Gradient Boosting Regressor)
RFE	:	Eliminación Recursiva de Características (Recursive Feature Elimination)
ARIMA	:	Modelo Autorregresivo Integrado de Media Móvil
IA	:	Inteligencia Artificial
MAE	:	Error Absoluto Medio (Mean Absolute Error)
MSE	:	Error Cuadrático Medio (Mean Squared Error)
RMSE	:	Raíz Cuadrada del Error Cuadrático (Root Mean Squared Error)
MAPE	:	Error Porcentual Absoluto Medio (Mean Absolute Percentage Error)
$R^2$	:	Coefficiente de Determinación
ETF	:	Fondo Cotizado en Bolsa (Exchange Traded Fund)
PBI	:	Producto Bruto Interno
Prec Petróleo	:	Precio del Petróleo
PBI EE. UU.	:	Producto Bruto Interno de Estados Unidos
Prod Mund Oro	:	Producción Mundial de Oro
Dema Mund Oro	:	Demanda Mundial de Oro
Índ Prec Dólar	:	Índice del Precio de Dólar
Infla EE. UU.	:	Inflación de Estados Unidos
Índ S&P 500	:	Índice S&P 500
Cost Prod Oro	:	Costo de Producción de Oro
Ren Bono EE. UU.	:	Rentabilidad del Bono de Estados Unidos
Tas Par Lab EE. UU.	:	Tasa de Participación Laboral de Estados Unidos
Euro/Libra	:	Tipo de Cambio Euro/Libra

Euro/Yen	:	Tipo de Cambio Euro/Yen
SVL ETF	:	iShares Silver Trust ETF
Dow Jones IA	:	Dow Jones Industrial Average
Tasa Int R. U.	:	Tasa de Interés de Reino Unido
GDX ETF	:	VanEck Gold Miners ETF
Accio EGO	:	Precio de las Acciones de Eldorado Gold Corporation
USO ETF	:	United States Oil Fund ETF
Prec Oro	:	Precio del Oro
Prod Mund Oro_ lag1	:	Producción Mundial de Oro del día anterior
Dema Mund Oro_ lag1	:	Demanda Mundial de Oro del día anterior
Índ Prec Dólar_ lag1	:	Índice del Precio de Dólar del día anterior
Infla EE. UU. _ lag1	:	Inflación de Estados Unidos del día anterior
Ren Bono EE. UU. _ lag1	:	Rentabilidad del Bono de Estados Unidos del día anterior
SVL ETF_ lag1	:	iShares Silver Trust ETF del día anterior
Dow Jones IA_ lag1	:	Dow Jones Industrial Average del día anterior
GDX ETF_ lag1	:	VanEck Gold Miners ETF del día anterior

## RESUMEN

La sostenibilidad y rentabilidad de un proyecto minero está fuertemente influenciada por el precio metales, como el oro. Aunque el aprendizaje automático, un subcampo de la inteligencia artificial, puede facilitar el pronóstico de estos precios, la mayoría de las compañías mineras no lo utilizan, principalmente por falta de conocimiento y porque su implementación puede ser costosa en algunos casos. El principal objetivo es desarrollar un modelo predictivo para la fluctuación del precio del oro utilizando mecanismos de machine learning. Para esto se empleó una metodología aplicada, exploratoria, correlacional y descriptiva, con un diseño no experimental-transversal y un enfoque cuantitativo. Se estructuró una base de datos desde enero de 2012 a diciembre de 2022, considerando variables que influyen y se relacionan con el precio del oro. Luego, se dividieron los datos en dos subconjuntos: el 85% para el entrenamiento de los algoritmos y el 15% para evaluar su rendimiento. Las pruebas mostraron que el SVR con un kernel radial fue el algoritmo más preciso, logrando un MAE = 6.20, RMSE = 8.37, MSE = 70.01 y  $R^2 = 0.99$  en la fase de prueba. Los pasos más importantes que permitieron desarrollar un buen modelo predictivo fueron: el ajuste adecuado de los hiperparámetros y la selección de variables más relevantes.

**Palabras Claves:** Inteligencia artificial, aprendizaje automático, modelo predictivo, precio del oro, hiperparámetros.

## ASBTRACT

The sustainability and profitability of a mining project is strongly influenced by the price of metals, such as gold. Although machine learning, a subfield of artificial intelligence, can facilitate the forecasting of these prices, most mining companies do not use it, mainly because of lack of knowledge and because it can be costly to implement in some cases. The main objective is to develop a predictive model for gold price fluctuation using machine learning mechanisms. For this purpose, an applied, exploratory, correlational and descriptive methodology was used, with a non-experimental-transversal design and a quantitative approach. A database was structured from January 2012 to December 2022, considering variables that influence and are related to the price of gold. Then, the data was divided into two subsets: 85% for training the algorithms and 15% to evaluate their performance. The tests showed that SVR with a radial kernel was the most accurate algorithm, achieving MAE = 6.20, RMSE = 8.37, MSE = 70.01 and  $R^2 = 0.99$  in the testing phase. The most important steps that allowed the development of a good predictive model were: the adequate adjustment of the hyperparameters and the selection of the most relevant variables.

**Keywords:** Artificial intelligence, machine learning, predictive model, gold price, hyperparameters.

## **CAPÍTULO I**

### **INTRODUCCIÓN**

En el sector minero, el precio de los metales es crítico, ya que afecta la rentabilidad, la viabilidad de los proyectos y la decisión de inversión. Los cambios en los precios de los metales preciosos, como el oro, son volátiles, y varios factores influyen en esto, razón por la cual las empresas mineras enfrentan desafíos para operar de manera sostenible con cambios continuos. Por lo tanto, es esencial utilizar modelos para pronosticar los precios futuros de los minerales que se extraerán a fin de evaluar la viabilidad de los proyectos mineros.

Existen diversos modelos tradicionales de predicción como ARIMA, el suavizado exponencial y la media móvil que se pueden utilizar para predecir el precio del oro. Sin embargo, investigaciones recientes señalan que el uso de inteligencia artificial genera mejores resultados predictivos, lo que podría generar mayores ganancias económicas para la industria minera. Según Castillo (2022), muchas grandes empresas mineras han realizado inversiones en la investigación y adopción de tecnologías basadas en inteligencia artificial en los últimos años. No obstante, algunas operaciones no aprovechan estas herramientas debido a limitaciones técnicas y financieras, así como también, por falta de conocimientos sobre el tema.

El problema se plantea en: ¿Cómo se desarrolla un modelo predictivo para la fluctuación del precio del oro utilizando mecanismos de machine learning?, cuya hipótesis es: El modelo predictivo desarrollado mediante mecanismos de machine learning permitirá predecir la fluctuación del precio del oro con un Error Absoluto Medio (MAE) inferior a 10 USD/onza troy, al ser evaluado con nuevos datos.

La justificación ante el problema antes planteado es que, debido a la gran importancia del precio del mineral, en este caso el del oro, en la industria minera; se desarrollará un modelo predictivo para la variación del precio del oro utilizando herramientas de aprendizaje automático, ya que actualmente es uno de los métodos más confiables y precisos a comparación de los métodos convencionales. Además, debido a la falta de conocimiento y poca información de cómo se

aplica el aprendizaje automático, el presente trabajo facilitará un enfoque educativo sobre estas nuevas herramientas tecnológicas; incentivando a los profesionales del rubro minero a aplicar estas herramientas en distintos problemas que se les puede presentar, ya que esta metodología puede ser utilizada en muchos campos.

El objetivo general es: Desarrollar un modelo predictivo para la fluctuación del precio del oro utilizando mecanismos de machine learning y los objetivos específicos son: Determinar las principales variables que influyen y tienen relación con la fluctuación del precio del oro, Identificar los hiperparámetros óptimos para mejorar el ajuste de los algoritmos planteados e Identificar el mejor algoritmo planteado de regresión para predecir el precio del oro.

La presente investigación se estructuró en cinco capítulos. El primer capítulo: presenta la introducción, que incluye el planteamiento y formulación del problema, hipótesis, justificación de la relevancia del estudio y los objetivos planteados. El segundo capítulo: aborda los antecedentes a nivel internacional, nacional y local, así como las bases teóricas relevantes y la definición de términos básicos clave utilizados en este estudio. El tercer capítulo: titulado materiales y métodos, especifica la ubicación y metodología de la investigación, que incluye el tipo de estudio, diseño, métodos aplicados, así como la descripción de las técnicas, instrumentos, materiales y procedimientos utilizados para la recolección y análisis de datos. El cuarto capítulo: denominado análisis y discusión de resultados, se analizan detalladamente los resultados obtenidos, enfocándose en evaluar los datos recolectados y contrastarlos con la hipótesis inicial de la investigación. El quinto capítulo: expone las conclusiones derivadas del estudio realizado, así como las recomendaciones basadas en los hallazgos obtenidos.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1. ANTECEDENTES TEÓRICOS

##### 2.1.1. Internacionales

**Manjula y Karthikeyan (2019)**, en su artículo científico titulado “Gold Price Prediction using Ensemble based Machine Learning Techniques”. Analizan la relación entre el precio del oro y otros parámetros como: el precio del petróleo crudo, el tipo de cambio rupia-dólar, la inflación y la tasa de interés. Para esto, utilizaron los siguientes algoritmos de aprendizaje automático: Regresor de Gradiente Aumentado, Regresor de Bosques Aleatorios y Regresión Lineal. Se concluye que los algoritmos de aprendizaje automático son útiles para analizar el precio del oro y los factores que influyen en él. Indican también que la precisión de los modelos depende de las características de los datos, por lo que se recomienda seguir investigando con diferentes metodologías y datos para mejorar la funcionalidad de los modelos.

**Makala y Li (2021)**, en su artículo científico titulado “Prediction of gold price with ARIMA and SVM”. Predicen el precio del oro utilizando el modelo de aprendizaje automático SVM y el modelo ARIMA convencional. El análisis se realiza con los datos diarios del Consejo Mundial del Oro desde 1979 hasta 2019. Los resultados muestran que SVM es superior a ARIMA en términos de rendimiento, evaluado mediante herramientas de medición como RMSE y MAPE; en base a esto señalan que los modelos de inteligencia artificial como el aprendizaje automático superan a los modelos convencionales como ARIMA.

**Carrión (2023)**, en su tesis de pre-grado titulada “Análisis del precio del oro mediante Inteligencia Artificial – proyecto integrador”, publicada por la Universidad Central del Ecuador. Determinó que las eventualidades sociales y los índices macroeconómicos son factores importantes a considerar para predecir el precio del oro. Con la ayuda del lenguaje de programación Python, creó un modelo predictivo, utilizando el 80% de los datos para el

entrenamiento y el 20% para la evaluación. El resultado fue una RMSE = 115 y un error de predicción inferior a 1.17 USD\$/onza troy.

### 2.1.2. Nacionales

**Huilca y Quispe (2019)**, en su tesis de pre-grado titulada “Sistema inteligente para la predicción del precio diario de las acciones mineras en la Bolsa de New York usando un modelo híbrido de redes neuronales y máquina de soporte vectorial de regresión”, publicada por la Universidad Nacional Mayor de San Marcos. Afirman que la correcta elección de variables y métodos, así como el preprocesamiento previo de los datos, influyen en la precisión de los modelos de aprendizaje automático.

**Fosca (2020)**, en su tesis de pre-grado titulada “Desarrollo de un modelo para la predicción del precio del cobre empleando herramientas de Machine Learning”, publicada por la Pontificia Universidad Católica del Perú. Concluye que los modelos de aprendizaje automático tienen una capacidad de predicción mayor que los modelos convencionales, en este caso ARIMA. Además, demuestra que cada algoritmo emplea una estructura y un análisis distinto para las variables. Por otro lado, señala que la selección óptima de variables y el ajuste de hiperparámetros son pasos cruciales para mejorar el rendimiento del modelo. De los algoritmos que empleó, obtuvo un mejor resultado con el de Regresión con Máquinas de Vectores de Soporte (SVR).

**Castillo (2022)**, en su tesis de pre-grado titulada “Desarrollo de modelos predictivos de regresión en la industria minera mediante el uso de algoritmo de machine learning”, publicada por la Universidad Nacional Mayor de San Marcos. Utilizó varios modelos de regresión de aprendizaje automático para abordar tres casos en la industria minera, logrando buenas predicciones en:

- ✓ **El precio del oro**, con el algoritmo de Regresión con Máquinas de Vectores de Soporte (SVR), con un  $R^2=0.94$ , MAE=4.63 y RMSE=5.29.
- ✓ **El contenido de sílice en el concentrado de hierro**, con el algoritmo Regresor de Gradiente Aumentado, con un  $R^2= 0.51$ , MAE=0.81 y RMSE=0.81.

- ✓ **El consumo de combustible de camiones mineros de gran tonelaje**, con el algoritmo Regresor de Bosques Aleatorios, con un  $R^2=0.98$ , MAE= 0.87, RMSE=0.91.

Los modelos se eligieron tras comparar diversos algoritmos en base a métricas de evaluación. Además, destaca que el aprendizaje automático puede mejorar la eficiencia, seguridad y rentabilidad en la industria minera.

### **2.1.3. Locales**

**Bardales y Zamora (2019)**, en su tesis de pre-grado titulada “Evaluación de los criterios de selección de un método de explotación mediante la inteligencia artificial. Caso práctico yacimiento de hierro en Ventanillas, Yonán, Cajamarca, 2019”, publicada por la Universidad Privada del Norte. Concluyen que la aplicación de la inteligencia artificial facilita la selección de un método de explotación minera de manera eficiente, tanto en operaciones como en proyectos mineros. Además, mencionan que los algoritmos adecuadamente entrenados tienen un mejor desempeño y un margen de error mucho menor, lo que ayuda a obtener los resultados deseados. Por otro lado, sugieren que la inteligencia artificial y metodologías afines pueden utilizarse para obtener resultados más precisos y optimizar los procesos mineros, mejorando así la administración de los recursos.

**Díaz (2017)**, en su tesis de pre-grado titulada “Uso de las redes neuronales artificiales en el modelado del ensayo de resistencia a compresión de concreto de construcción según la norma ASTM C39/C39M”, publicada por la Universidad Nacional de Cajamarca. Utilizó Redes Neuronales Artificiales para predecir la resistencia a compresión del concreto. El modelo, entrenado con cinco redes neuronales, logró una tasa de error de 3.29%, demostrando que la inteligencia artificial o herramientas similares, pueden manejar eficazmente problemas de predicción.

## **2.2. BASES TEÓRICAS**

### **2.2.1. El Oro**

Es reconocido como uno de los metales preciosos más importantes, cuyo nombre deriva del latín Aurum, es por esto que su símbolo químico es Au. Se caracteriza por su elevada densidad, ductibilidad y maleabilidad; es de color amarillo brillante, inoxidable y químicamente inactivo. El oro tiene muchas aplicaciones en distintas industrias, y a comparación de los otros metales, es uno de los más cotizados por la humanidad desde hace mucho tiempo (López, 2007).

#### **2.2.1.1. Evolución del Precio del Oro**

Desde la antigüedad, el oro ha sido considerado un símbolo de riqueza, estabilidad y valor intrínseco. A lo largo de la historia, su precio ha mostrado variaciones notables debido a diversos factores económicos, políticos y sociales (Saavedra, 2014).

A continuación, se hará una exploración de cómo ha evolucionado el precio del oro desde 1975 hasta el 2024 (ver Figura 1), donde se destacarán periodos de auge y declive, así como los principales factores que impulsaron estas tendencias.

Entre 1975 y 1980 se produjo un período de gran crecimiento en el precio del oro ya que en 1971 el presidente Nixon de Estados Unidos unilateralmente eliminó la convertibilidad del dólar americano en oro, lo que llevó a la devaluación del dólar americano y aumentó el precio del oro hasta los 38 USD/onza troy. En 1975 se inició, en Nueva York y Chicago, la negociación con futuros sobre el oro en el mercado de materias primas. Adicionalmente, en 1980, el precio del oro experimentó un fuerte impulso, llegando a los 870 USD/onza troy, debido a la alta inflación en Estados Unidos desde 1976 y a la invasión de Afganistán por parte de la Unión Soviética (Saavedra, 2014).

Durante el período comprendido entre 1981 y 1999, el precio del oro experimentó una tendencia bajista en general, ya que la demanda de oro como activo seguro disminuyó y los inversores buscaron otras oportunidades de inversión. Desde 1996 hasta 1999 varios bancos

centrales comenzaron a vender su oro, lo cual contribuyó significativamente a la tendencia bajista, al aumentar la oferta de oro en el mercado (Saavedra, 2014).

Entre los años 2000 y 2011 el precio del oro comenzó a aumentar nuevamente, impulsado por varios factores. En el 2000 estalló la llamada burbuja punto com debido a una crisis en el sector de empresas tecnológicas e internet, lo que dio pie a una recesión, un período de inseguridad económica y el comienzo de un mercado en crecimiento para el oro. Sin embargo, fue en 2003 cuando se lanzó el primer ETF (fondo de inversión cotizado) de oro, que permitió que inversores institucionales y grandes fondos puedan invertir en este activo de manera más accesible, generando una importante entrada de capital y un aumento notable en el precio del oro (Castellano, 2020). A partir de 2006, se inició la crisis subprime en Estados Unidos y en 2007 se desencadenó la crisis financiera, que posteriormente llevó a la Gran Recesión (Saavedra, 2014), esto provocó que el precio del oro experimente un aumento significativo debido a su condición como activo refugio en tiempos de incertidumbre económica. Posteriormente, en 2009 los bancos centrales de países emergentes comenzaron a comprar oro, y se pusieron en marcha programas de expansión cuantitativa que implicaban la compra de activos financieros, como bonos y deuda pública, para aumentar la oferta monetaria. Esto provocó una caída en el valor de la moneda y un incremento en el precio de los activos estimados en dicha moneda. Como resultado, el precio del oro logró un récord histórico en 2011, llegando a los 1920 USD/onza troy, lo que generó una burbuja financiera que finalmente estalló (Castellano, 2020).

Luego, entre 2012 y 2017 la subida, similar a la ocurrida en los años 80, fue seguida por un descenso, y el valor del oro alcanzó los 1600 USD/onza troy. En 2013, con la economía más estable, los consumidores retomaron su consumo, lo que aprovecharon los inversores para reducir gradualmente sus inversiones en oro, ocasionando que el metal experimente otra bajada, llegando a los 1200 USD/onza troy. En 2015, la Reserva Federal de EE. UU., que había bajado los intereses previamente, decidió aumentarlos nuevamente, impactando directamente en el precio del oro, que cayó a 1180 USD/onza troy. En 2016, el BREXIT en Europa llevó a una caída inesperada en los mercados y a un aumento del oro a 1300 USD/onza troy. En agosto del mismo año, una nueva subida llevó al oro a 1400 USD/onza, debido a la postergación del aumento de intereses por parte de la Reserva Federal de Estados Unidos. En los meses siguientes, el oro cayó a 1140 USD/onza troy debido al ascenso de Donald Trump como presidente de Estados Unidos. En 2017, el oro subió un 13% (Carrión, 2023).

Finalmente, entre 2018 y 2024, el precio del oro mostró grandes fluctuaciones debido a tensiones comerciales, desaceleración económica y la pandemia de COVID-19. En 2019, subió por preocupaciones geopolíticas y económicas, y en 2020, el oro se disparó debido a la incertidumbre global generada por la pandemia. En 2021, con la distribución de vacunas, el oro se estabilizó, aunque aún estaba influenciado por eventos internacionales. En marzo de 2022, el oro alcanzó un récord de aproximadamente 2070 USD/onza debido a la invasión de Rusia en Ucrania. En 2023, el oro alcanzó un nuevo récord histórico de 2135.4 USD/onza el 4 de diciembre, impulsado por la alta demanda en mercados emergentes, el apoyo de los bancos centrales, la caída del dólar y expectativas de recortes en las tasas de interés. En agosto de 2024, el oro subió hasta 2513 USD/onza, marcando otro récord gracias a la caída del dólar, menores rendimientos de bonos y la reducción de aranceles en India, que aumentó la demanda (World Gold Council, 2024).



Figura 1. Precio Histórico del Oro  
Fuente: Adaptado de Investing, 2024.

### **2.2.1.2. Mercado del Oro**

El mercado del oro es un factor muy importante de la economía mundial. Los bancos utilizan este valioso metal para mantener su dinero seguro y las personas que invierten también lo consideran una buena manera de asegurarse de que su dinero esté bien protegido. En tiempos de inestabilidad financiera, el oro se vuelve aún más relevante, ya que actúa como un recurso de respaldo en situaciones de emergencia y contribuye a la recuperación económica (Castellano, 2020).

#### **a. Oferta de Oro**

En la Figura 2 se muestran las toneladas de oro correspondientes a la producción minera, al oro reciclado y el total de la oferta a nivel mundial desde el 2010 hasta el 2022, donde se puede observar que en 2010 hubo una oferta total de 4426 toneladas y 4790 toneladas en 2022, lo cual indica que hubo un ligero incremento. Adicionalmente, World Gold Council (2023a) indica que la oferta aumentó en un 2% en el 2022 respecto al año anterior, ya que tanto la producción minera como el reciclaje registraron un crecimiento modesto.

Como se sabe y se corrobora en la Figura 2, la producción minera es responsable de la mayor parte del suministro global de oro, mostrando una tendencia ascendente desde 2010 hasta 2022, aumentando 895 toneladas en este periodo. World Gold Council (2023d) indica que en 2022 la producción minera anual ha experimentado un aumento del 1% en comparación del año anterior, sin embargo, sigue siendo inferior al máximo histórico registrado en 2018, que fue de aproximadamente 3655 toneladas. Por otro lado, Gopaul (2023) indica que China es el mayor productor de oro del mundo, seguido de Rusia, Australia, Canadá y Estados Unidos.

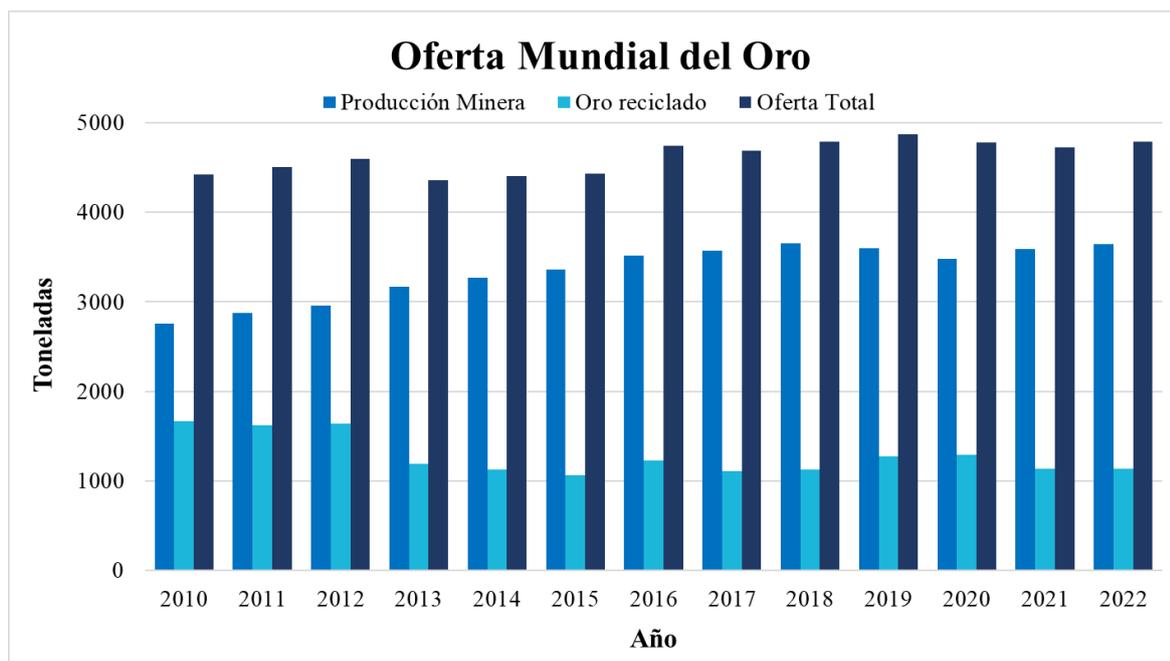


Figura 2. Oferta Mundial del Oro  
Fuente: Adaptado de World Gold Council, 2023c.

En cuanto al oro reciclado, se observa una tendencia descendente desde 2010 hasta 2022, disminuyendo 531 toneladas en este periodo. World Gold Council (2023d) indica que en 2022 el oro reciclado aumentó un 1% respecto al año anterior. Sin embargo, a pesar de que el oro alcanzó un precio promedio récord en 2022, el suministro de oro reciclado fue un 30 % inferior al máximo histórico registrado en 2009, que fue de 1728 toneladas aproximadamente.

El reciclaje es la principal fuente de suministro de oro que reacciona rápidamente a los cambios en el precio del oro y las crisis económicas. La mayor parte del oro reciclado, aproximadamente el 90%, proviene de la joyería, mientras que el oro recuperado de dispositivos tecnológicos representa el resto. Aunque la producción minera constituye la mayor parte del suministro mundial de oro, normalmente alrededor del 75% cada año, la demanda anual supera la cantidad extraída y este déficit se equilibra mediante el reciclaje (World Gold Council, 2023b).

## b. Demanda de Oro

En la Figura 3 se muestra la evolución de la demanda mundial de oro en total y por sectores, desde el 2010 hasta el 2022, donde hubo un incremento del 12% de oferta total. World Gold Council (2023a) indica que la demanda total de oro en 2022 aumento un 18% respecto al año

anterior, alcanzando las 4741 toneladas, casi igualando el nivel registrado en 2011, un período caracterizado por una demanda de inversión excepcional.

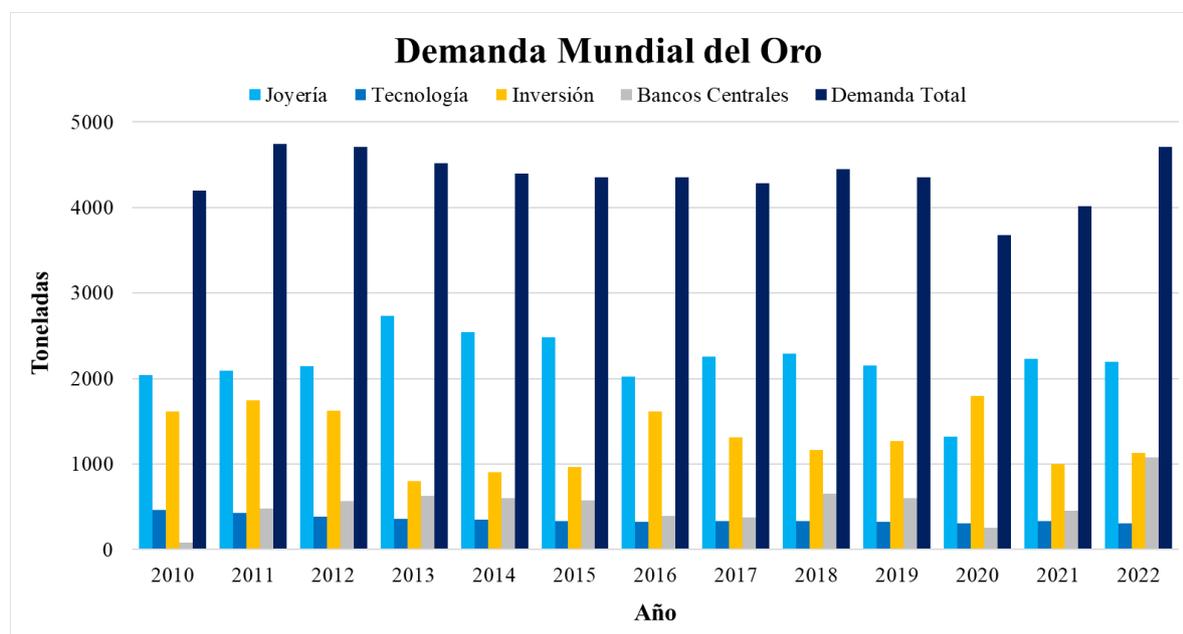


Figura 3. Demanda Mundial del Oro  
Fuente: Adaptado de World Gold Council, 2023c.

Según Castellano (2020), la joyería constituye la mayor fuente de demanda anual de oro y equivale al 50% de la demanda total. En el gráfico se observa que durante el 2010 y el 2013 la demanda de joyería estuvo en aumento, sin embargo, después de este periodo a disminuido debido a la estabilización del precio del oro y su aumento sostenido durante los últimos años. World Gold Council (2023a) señala que en 2022 la demanda de joyería experimentó una leve reducción del 3% respecto al año anterior, llegando a 2086 toneladas. Esta disminución se debe principalmente a la notable caída en la demanda de joyas en China, la cual se redujo en un 15% debido a las restricciones de COVID-19.

La demanda de tecnología, como se puede apreciar en el gráfico, es pequeña a comparación de los demás sectores y ha mostrado una tendencia ligeramente decreciente, pero estable, entre los años 2010 y 2022, con un promedio anual de 351 toneladas aproximadamente. En 2022, la demanda de tecnología disminuyó un 6% a comparación del año anterior.

En cuanto al sector de inversión, Saavedra (2022) indica que gracias a la crisis financiera de 2007, la demanda de inversión en 2008, ha ganado importancia debido a su menor sensibilidad a los precios elevados, y dos factores clave explican esta tendencia: los bancos centrales y los fondos cotizados (ETFs). Castellano (2020) señala que hasta 2011 la demanda de inversión en oro aumentó significativamente en 1700 toneladas. Posteriormente, a partir de 2015 esta demanda volvió a sus niveles anteriores sin conseguir superarlos. El aumento de los flujos de inversión en fondos cotizados y productos relacionados fue la causa principal de este incremento; en menor medida, también influyó el aumento de las reservas de oro de los bancos centrales. Según World Gold Council (2023a), en 2022 la demanda de inversión aumentó en un 10% respecto al año anterior.

Por otro lado, la demanda de bancos centrales muestra una tendencia creciente entre los años 2010 y 2022, ya que, según Castellano (2020) después de la gran recesión, se ha observado un cambio en la actitud de los bancos centrales de países emergentes y economías avanzadas hacia el oro. Los bancos centrales de países emergentes han pasado a ser compradores de oro, mientras que los de economías avanzadas han dejado de venderlo y mantienen sus reservas, esto se debe a un reconocimiento generalizado de la importancia del oro en las reservas, lo que ha llevado a una tendencia creciente de compras masivas. Como resultado se espera que su precio continúe aumentando a largo plazo debido a esta creciente demanda. Según World Gold Council (2023a), en 2022 la demanda de bancos centrales fue de 1136 toneladas, superando en más del doble la cifra del año anterior que fue de 450 toneladas, marcando un récord sin precedentes en los últimos 55 años.

### **2.2.2. Aprendizaje Automático (Machine Learning)**

La inteligencia artificial abarca muchos subcampos o ramas, pero entre los principales y más conocidos se encuentra el aprendizaje automático (machine learning), el cual es el estudio de algoritmos que realizan una tarea sin necesidad de definir explícitamente el código para llevarla a cabo, si no que utilizan datos para aprender (Agrawal, 2020). En otras palabras, el aprendizaje automático es la ciencia que aplica distintos algoritmos con el fin de que la máquina aprenda patrones dentro de los datos, aprenda de ello y ajuste su conocimiento de manera automatizada (Madhavan et al., 2019).

Por otro lado, Pan (2023) nos dice que el aprendizaje automático es la disciplina científica que permite a los humanos diseñar algoritmos y enseñar a las computadoras a aprender patrones a partir de grandes cantidades de datos y utilizar esos patrones para tomar decisiones o hacer predicciones automáticamente. En este proceso, los datos pueden ser de diversos tipos, como valores numéricos, texto, gráficos, fotos, audio y más. Está estrechamente relacionado con la estadística computacional, la ciencia de datos y la minería de datos, y se aplica con frecuencia a otros dominios de investigación, como el procesamiento del lenguaje natural, la visión artificial, la robótica, la bioinformática, y otros campos similares.

### **2.2.2.1. Tipos de Aprendizaje Automático**

Agrawal (2020) señala que de acuerdo al el tipo de problema que se busca resolver, los algoritmos de aprendizaje automático pueden clasificarse en diferentes tipos, como:

#### **a. Aprendizaje Supervisado**

Camastra y Vinciarelli (2008) mencionan que el aprendizaje supervisado consiste en utilizar datos de entrada y salida para enseñar a un modelo a predecir salidas futuras. Los elementos en el conjunto de datos ya están vinculados a los valores objetivo conocidos, es decir, clases o valores reales. Como ejemplos se tiene el reconocimiento de letras escritas a mano y la predicción de índices del mercado de valores. El objetivo es encontrar una función que relacione con precisión las entradas con las salidas y minimizar los errores en lo posible. En función de la salida, que el modelo está prediciendo, el aprendizaje supervisado puede ser de tipo: clasificación (etiquetar categorías) o regresión (predecir valores numéricos).

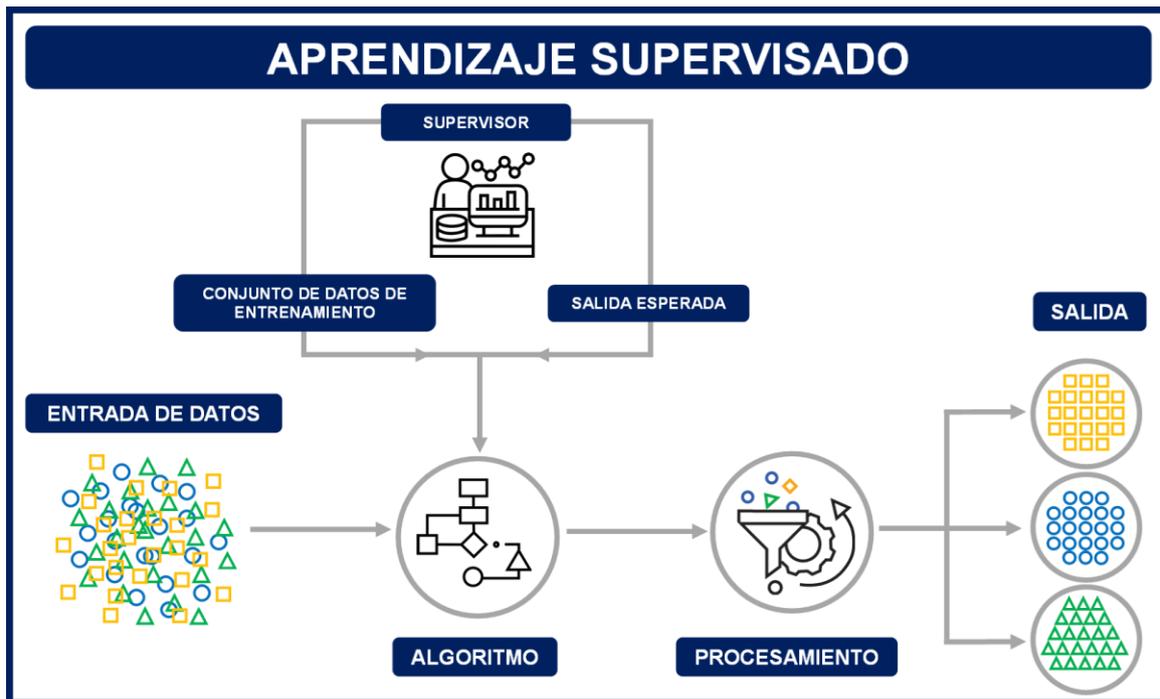


Figura 4. Esquema de Aprendizaje Supervisado  
Fuente: Adaptado de Universidad Anáhuac, 2023.

## ❖ Clasificación

En el aprendizaje de clasificación, el espacio de salida se organiza para determinar si dos elementos son iguales o diferentes. Cada elemento en el espacio de salida es una clase, y el objetivo es asignar nuevas entradas a categorías discretas o clases utilizando un algoritmo llamado clasificador. Este tipo de problema es común en actividades de identificación de patrones, como la identificación de letras del alfabeto a partir de imágenes de caracteres (Camastra y Vinciarelli, 2008). En otras palabras, los problemas de clasificación se presentan cuando la variable objetivo o target contiene dos o más clases categóricas, y el propósito del modelo es predecir a cuál clase pertenece la variable objetivo considerando las variables de entrada proporcionadas (Idrogo, 2022).

Existen diversos algoritmos desarrollados específicamente para tratar con problemas de clasificación. Idrogo (2022) indica que entre estos algoritmos se encuentran los siguientes:

- ✓ Análisis Discriminante Lineal
- ✓ K-Vecinos más Cercanos
- ✓ Clasificador de Máquinas de Vectores de Soporte

- ✓ Redes Neuronales
- ✓ Clasificador de Árboles de Decisión

## ❖ **Regresión**

Un problema de regresión es aquel en el que la variable objetivo toma valores numéricos continuos. El modelo desarrollado relaciona las variables de entrada con el valor de la variable objetivo para generar una estimación numérica del resultado deseado (Idrogo, 2022).

Al igual que en los problemas de clasificación, existen varios algoritmos especializados en el tratamiento de problemas de regresión. Idrogo (2022) menciona los siguientes algoritmos:

- ✓ Regresión Lineal
- ✓ Regresión Lineal Multivariable
- ✓ Regresión Ridge
- ✓ Red de Elasticidad
- ✓ Redes Neuronales
- ✓ Regresor de Árboles de Decisión
- ✓ Regresión con Máquinas de Vectores de Soporte

## **b. Aprendizaje No Supervisado**

El aprendizaje no supervisado se refiere a un tipo de problema en el que los datos consisten únicamente en una muestra de objetos sin valores objetivo asociados. Aquí, no hay un profesor para guiar el proceso. El objetivo principal es extraer estructura y patrones de los datos (Camastra y Vinciarelli, 2008). En otras palabras, el aprendizaje no supervisado es llamado así porque los modelos en este enfoque trabajan con un conjunto de datos que contiene parámetros, pero no tienen un objetivo o target predefinido. En lugar de eso, el modelo busca de manera autónoma encontrar estructuras y patrones dentro de los datos que se utilizan para construirlo (Idrogo, 2022).

Algunos ejemplos comunes de tareas en este ámbito incluyen la segmentación de imágenes y texto, así como la detección de novedades en el control de procesos. (Camastra y Vinciarelli, 2008).



Figura 5. Esquema de Aprendizaje No Supervisado  
Fuente: Adaptado de Universidad Anáhuac, 2023.

De acuerdo con Vega (2019), como ejemplos de modelos en el aprendizaje supervisado tenemos:

- ❖ Agrupamiento
- ✓ Algoritmo K-medias.
- ✓ Algoritmo de Particionamiento Alrededor de Medoides
- ✓ Algoritmo de Agrupamiento para Aplicaciones Grandes
- ✓ Algoritmo K-Vecinos más Cercanos
- ❖ Asociación
- ❖ Redes Neuronales Artificiales

### c. Aprendizaje Reforzado

A diferencia de una tarea de aprendizaje supervisado, en el aprendizaje reforzado, el algoritmo no recibe instrucciones explícitas sobre qué acciones tomar en una situación determinada (Camastra y Vinciarelli, 2008). En otras palabras, un algoritmo de aprendizaje por refuerzo aprende a partir de su interacción con el entorno; si tiene un buen desempeño, recibe una recompensa, y el objetivo es maximizar esa recompensa (Agrawal, 2020).

Para una mejor comprensión del aprendizaje reforzado, Agrawal (2020) toma como ejemplo el juego del "dinosaurio corriendo" en Chrome, donde el dinosaurio corre continuamente hacia obstáculos y el jugador debe presionar la barra espaciadora en el momento preciso para hacer que el dinosaurio salte sobre los obstáculos y aumente de esta manera su puntuación. En esta situación, esos puntos son la recompensa y saltar es la variable que debe decidirse en el momento adecuado. Una de las aplicaciones más destacadas de este aprendizaje es permitir a un robot aprender a caminar mediante el proceso de prueba y error.

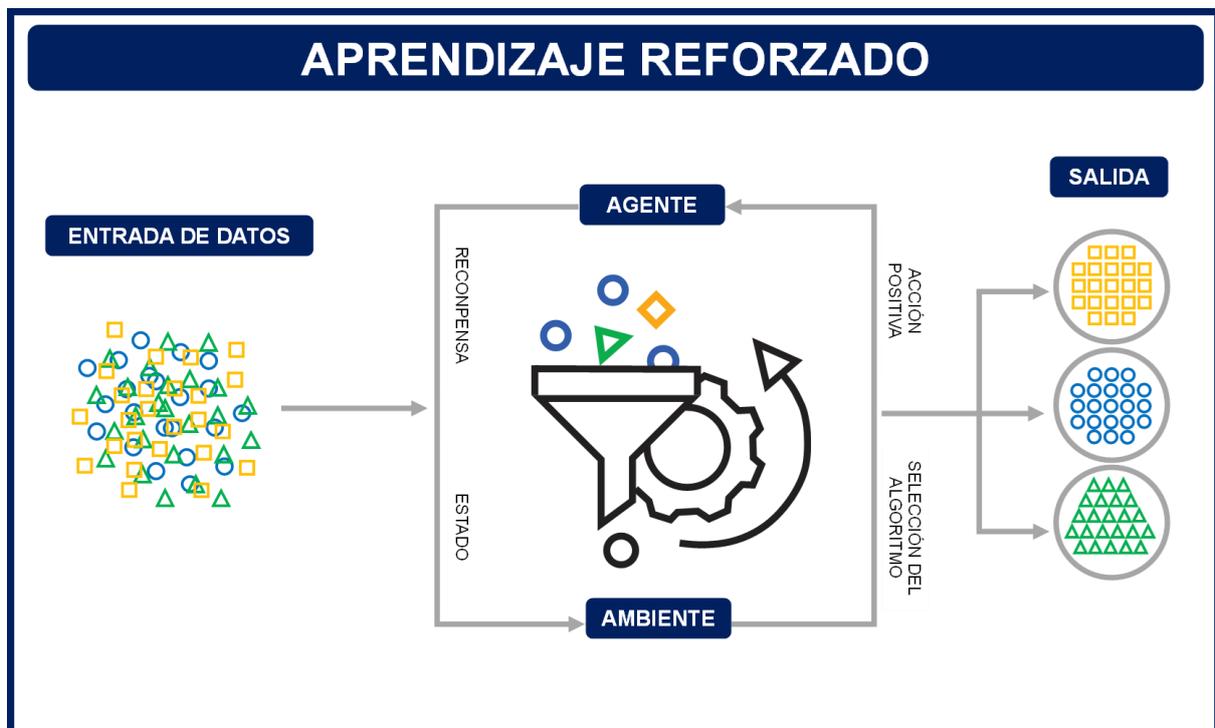


Figura 6. Esquema de Aprendizaje Reforzado  
Fuente: Adaptado de Universidad Anáhuac, 2023.

Entre los algoritmos del aprendizaje reforzado se encuentran:

- ✓ Red Neuronal Profunda (Deep Q-Network)
- ✓ Optimización de Políticas Proximales
- ✓ Actor-Crítico (Actor-Critic)

### **2.2.3. Aspectos Clave en Modelos Predictivos**

#### **2.2.3.1. Algoritmos Relevantes de Regresión**

##### **a. Regresión con Máquinas de Vectores de Soporte (Support Vector Regression - SVR)**

Se trata de un algoritmo de aprendizaje supervisado diseñado para abordar problemas de regresión, lo que significa que se usa para pronosticar valores numéricos continuos. El SVR se fundamenta en los principios de Máquinas de Vectores de Soporte (SVM) y aplica un método de optimización para determinar la función de regresión más adecuada.

A continuación, se describe el funcionamiento del algoritmo SVR para una mejor comprensión:

En la Figura 7 se visualiza gráficamente el algoritmo SVR donde la línea intermedia es el hiperplano que modela la tendencia de los datos que en este caso es lineal y está representado por la siguiente fórmula:

$$y = wx + b$$

Las líneas de los extremos representan a los vectores de soporte que tienen como finalidad cubrir la mayor cantidad de datos posibles y se expresan a través siguientes ecuaciones:

$$y = wx + b + \epsilon$$

$$y = wx + b - \epsilon$$

Sin embargo, aún con la ayuda de los vectores no se logra cubrir todos los datos, dejando fuera algunos de ellos, los cuales son considerados errores. Es por esto que se calcula la distancia entre el error y el vector más cercano, a esta distancia se la conoce como épsilon. En base a esto se utiliza el enfoque de optimización de Vapnik (1995), con el fin de encontrar la función de regresión óptima.

$$\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i + \xi_i^*)$$

$$\text{subject to } \begin{cases} y_i - wx_i - b \leq \varepsilon + \xi_i \\ wx_i + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

Donde:

- $\|w\|^2$ : Es la norma al cuadrado del vector de pesos  $w$ .
- $C$ : Es un parámetro de regularización que controla el balance entre la complejidad del modelo y el error de entrenamiento.
- $\xi_i, \xi_i^*$ : Son variables de holgura que representan los errores permitidos para cada punto de entrenamiento.
- Las restricciones establecen que las diferencias entre las predicciones del modelo ( $wx_i - b$ ) y los valores reales ( $y_i$ ) deben estar dentro de un margen  $\varepsilon$ , y las variables de holgura  $\xi_i$  y  $\xi_i^*$  se utilizan para cuantificar los errores permitidos.

En base a lo anterior, como es un caso lineal, se define que la función de predicción del SVR es la siguiente:

$$f(x) = \sum_{i=1}^m (\alpha_i^* - \alpha_i) \langle x_i, x \rangle + b$$

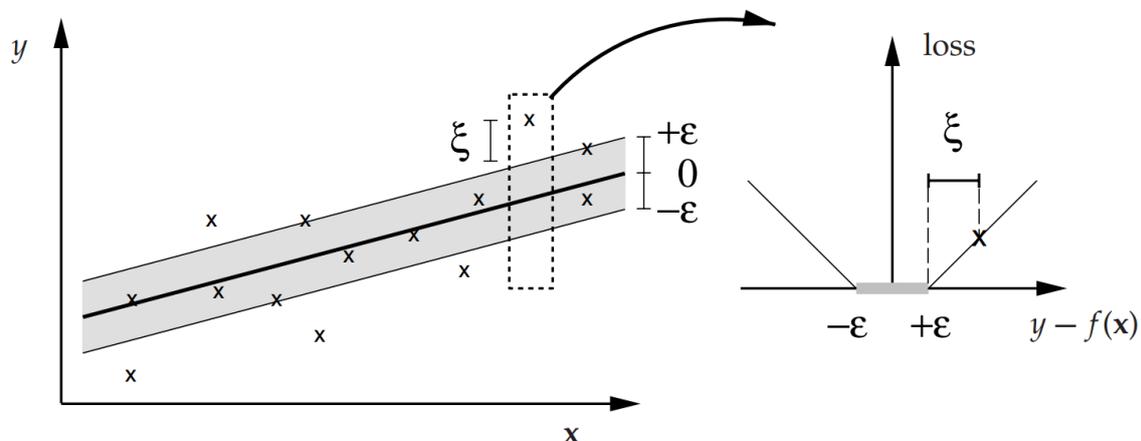


Figura 7. Representación del Algoritmo SVR  
Fuente: Schölkopf y Smola, 2002.

En la mayoría de los casos, no se trabaja con hiperplanos lineales, debido a las limitaciones y dificultades del aprendizaje lineal. Para superar estas limitaciones, se utilizan funciones de Kernel, ya que estas funciones son capaces de transformar los datos agregando dimensiones adicionales mientras se preservan las características originales de los objetos. Esto permite encontrar un hiperplano que se ajuste de manera más óptima al problema. En base a esto se planteó la siguiente función general de predicción del SVR (Jadue, 2021).

Donde:

$$f(x) = \sum_{i=1}^m (\alpha_i^* - \alpha_i) k(x_i, x) + b$$

- m: Es el número total de vectores de soporte.
- $\alpha_i^*$  y  $\alpha_i$ : Son los coeficientes de peso asociados a los vectores de soporte.
- $k(x_i, x)$ : Es el resultado de aplicar el kernel a los vectores de soporte ( $x_i$ ) y el vector de entrada ( $x$ ).
- b: Es el término de sesgo.

Es importante destacar que existen diversas funciones de Kernel disponibles, y la elección de la función adecuada depende de los datos y el problema específico. Algunas de estas funciones de Kernel incluyen:

- ✓ **Kernel lineal:**  $k(x, x') = \langle x, x' \rangle$
- ✓ **Kernel polinómico:**  $k(x, x') = \langle x, x' \rangle^d$
- ✓ **Kernel radial:**  $k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$

#### b. **Regresor de Bosques Aleatorios (Random Forest Regressor - RFR)**

Deriva del algoritmo de Bosques Aleatorios, el cual es un algoritmo de aprendizaje automático que se fundamenta en el concepto de ensamblado o modelo de conjunto (ensemble model). Un modelo de conjunto es un modelo que mejora la probabilidad predictiva al combinar modelos. (Kwak et al. 2022). Este modelo es un ensamble de tipo bagging el cual consiste en entrenar modelos individuales (árboles de decisión) de forma paralela y cada uno de estos modelos se entrena con un subconjunto aleatorio de los datos (Jadue, 2021). En otras palabras, los bosques aleatorios son un conjunto de modelos predictivos fundamentados en árboles de decisión, donde cada uno de estos árboles se construye utilizando los elementos de un vector aleatorio, que se selecciona de manera independiente y mantiene una distribución uniforme para todos

los árboles en el conjunto. Además, es preciso mencionar que, conforme se incrementa el número de árboles en el bosque, el error de predicción tiende estabilizarse en un límite con alta confiabilidad (Breiman, 2001).

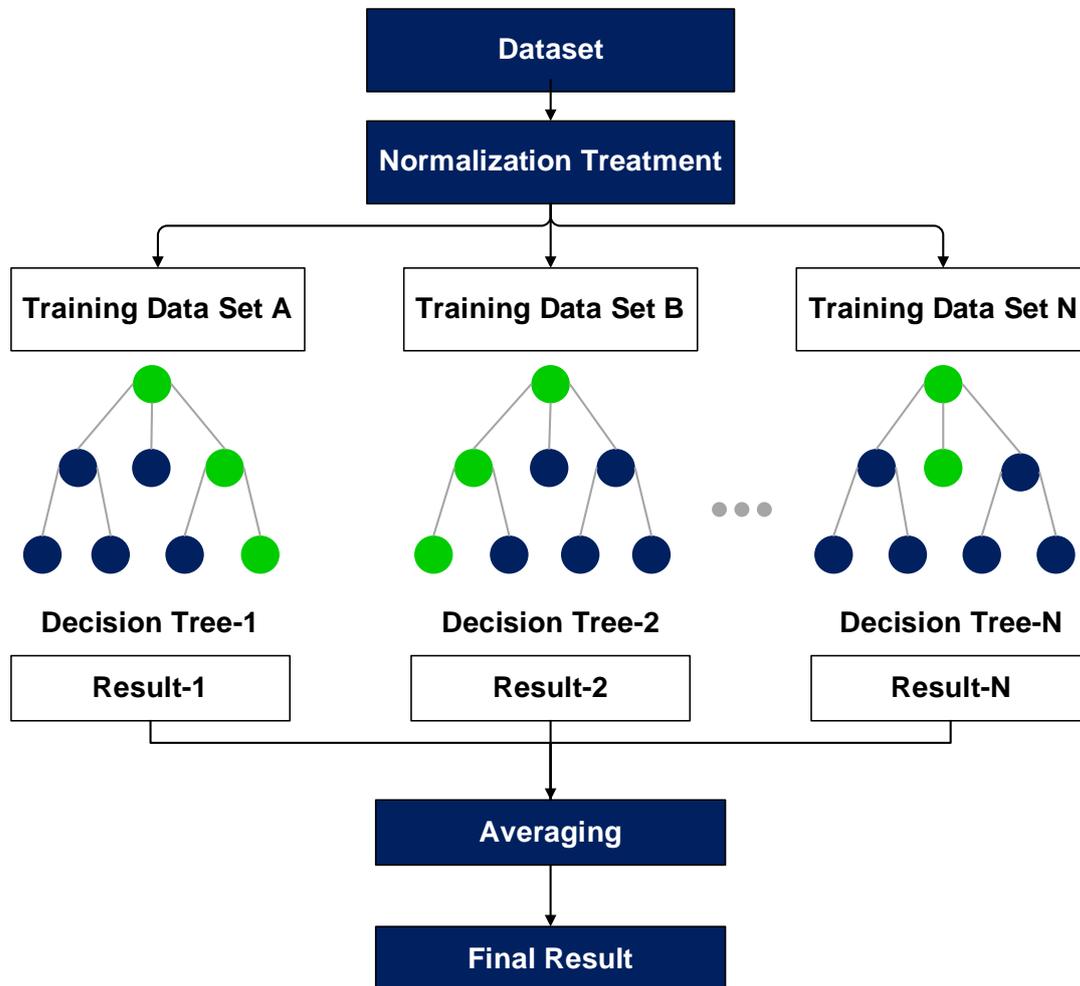


Figura 8. Esquema del Algoritmo Regresor de Bosques Aleatorios  
Fuente: Adaptado de Zhang et al., 2023.

En la Figura 8 se ilustra cómo funciona el algoritmo Regresor de Bosques Aleatorios, donde se tiene un grupo de árboles de decisión en el cual primero se genera un árbol de decisión (Decision Tree 1) utilizando un subgrupo aleatorio de los datos de entrenamiento; el árbol de decisión divide el conjunto de datos en ramas utilizando diferentes características. Este proceso se repite varias veces, creando así diversos árboles de decisión cada uno con diferentes subgrupos de datos seleccionados aleatoriamente. Una vez que se han creado todos los árboles, se obtiene la predicción final del modelo, a través de un promedio de las predicciones de todos los árboles del conjunto.

### c. Regresor de Incremento Gradual (Gradient Boosting Regressor - GBR)

Es otro algoritmo de aprendizaje automático utilizado principalmente para tareas de regresión. Este algoritmo también utiliza un enfoque de ensamble que combina varios modelos más sencillos, como: árboles de decisión. Construye múltiples árboles de manera secuencial, donde cada nuevo árbol aprende corrigiendo los errores de los anteriores. De esta manera, el algoritmo realiza predicciones aprovechando el "impulso" (boosting) de un conjunto de modelos débiles, en este caso, árboles de decisión, hasta generar un modelo más robusto (Rao et al., citado por Asante et al., 2022).

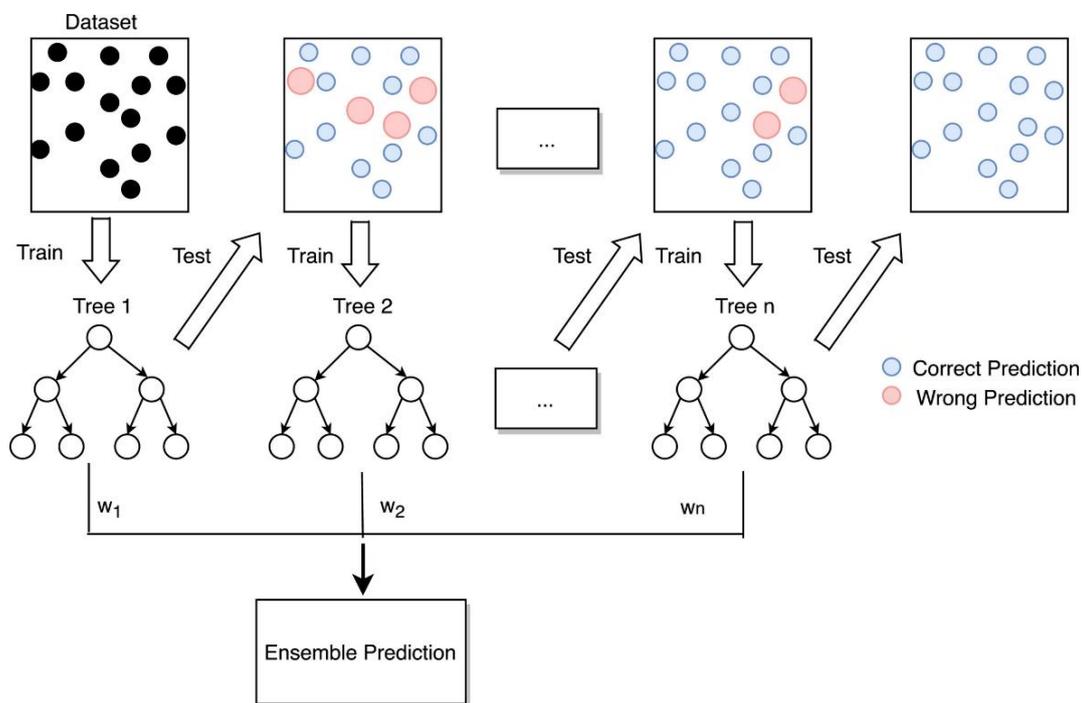


Figura 9. Esquema del Algoritmo Regresor de Incremento Gradual  
Fuente: Zhang et al., 2021.

En la Figura 9 se ilustra cómo funciona el algoritmo Regresor de Incremento Gradual, donde se tiene un grupo de árboles de decisión en el cual se comienza con un solo árbol de decisión simple (Tree 1), también conocido como "árbol débil", que se usa como estimador base. Luego, se ajusta el árbol débil al conjunto de entrenamiento y se calculan las predicciones iniciales, con el fin de calcular los residuos (errores). Como siguiente paso se crean varios árboles adicionales y cada uno se ajusta para predecir los residuos del paso anterior. Cada árbol nuevo se enfoca en corregir los errores cometidos por el modelo en el paso anterior. Finalmente, para obtener la predicción final del modelo se realiza una combinación de todas las predicciones de los árboles.

### 2.2.3.2. Métricas de Evaluación en Modelos Predictivos

- a. **Error Absoluto Medio (MAE):** Mide la dimensión del error absoluto entre los valores predichos y reales. Un MAE más bajo señala que el error es de menor magnitud, lo cual implica una mayor exactitud en los pronósticos del modelo; por otro lado, un MAE más elevado hace referencia a un modelo menos preciso. Es importante mencionar que el MAE mide la magnitud del error sin considerar la dirección del mismo (Carles, 2023).

La ecuación del MAE es la siguiente:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

**Donde:**

N = número de observaciones

$y_i$  = valor real de la variable respuesta

$\hat{y}_i$  = valor predicho por el modelo

- b. **Error Cuadrático Medio (MSE):** Mide la dimensión del error cuadrático entre los valores predichos y reales. Un MSE más bajo señala que el error cuadrático es de menor magnitud, lo cual implica una mayor exactitud en los pronósticos del modelo; por otro lado, un MSE más elevado, hace referencia a un modelo menos preciso. Es importante señalar que el MSE otorga mayor importancia a los errores más significativos y es más perceptible a los valores atípicos, es por esto que un modelo con un MSE más elevado puede ser influenciado por un punto de datos cuyo error es significativo (Carles, 2023).

La ecuación del MSE es la siguiente:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

**Donde:**

N = número de observaciones

$y_i$  = valor real de la variable respuesta

$\hat{y}_i$  = valor predicho por el modelo

- c. **Raíz Cuadrada del Error Cuadrático Medio (RMSE):** Se obtiene al calcular la raíz cuadrada del MSE. Un RMSE más bajo señala que el error cuadrático es de menor magnitud, lo cual implica una mayor exactitud en los pronósticos del modelo; por otro lado, un RMSE más elevado, hace referencia a un modelo menos preciso. Así como el MSE, el RMSE también otorga mayor importancia a los errores más significativos y es más perceptible a los valores atípicos (Carles, 2023).

La ecuación del RMSE es la siguiente:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

**Donde:**

$N$  = número de observaciones

$y_i$  = valor real de la variable respuesta

$\hat{y}_i$  = valor predicho por el modelo

- d. **Coefficiente de Determinación ( $R^2$ ):** Mide qué tan bien el modelo puede explicar la variación en los datos. Un  $R^2$  cercano a uno indica que el modelo captura la mayoría de las alteraciones en los datos, lo que se traduce en predicciones más precisas, sin embargo, un  $R^2$  igual a uno no garantiza que el modelo sea perfecto, ya que también es importante considerar su capacidad de generalización y su desempeño con datos nuevos (Carles, 2023).

La ecuación del  $R^2$  es la siguiente:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2}$$

**Donde:**

$y_i$  = valor real de la variable respuesta

$\hat{y}_i$  = valor predicho por el modelo

$\bar{y}_i$  = media de  $y_i$

### 2.2.3.3. Problemas Comunes en Modelos Predictivos

El sobreajuste es un problema frecuente en el aprendizaje automático, donde un modelo funciona correctamente con los datos que se usaron para entrenarlo, pero no logra adaptarse de manera efectiva a datos nuevos o no vistos (como los datos de prueba); este comportamiento está relacionado con una alta varianza y suele ocurrir cuando el modelo tiene demasiados parámetros, haciéndolo excesivamente complejo para los datos originales. Por otro lado, también el modelo puede experimentar subajuste o alto sesgo, que sucede cuando el modelo es demasiado simple para identificar los patrones en los datos de entrenamiento, lo que provoca un desempeño deficiente tanto en los datos de entrenamiento como en los de prueba (Raschka y Mirjalili, 2019).

Scikit-learn (2020) explica a través de un ejemplo sencillo el problema de subajuste y sobreajuste. En la Figura 10 se presenta la función que deseamos estimar, que forma parte de la función coseno; adicionalmente se representan las muestras de la función real junto con las aproximaciones de distintos modelos que tienen características polinómicas de diferentes grados. Podemos observar que una función lineal (un polinomio de grado 1) resulta insuficiente para ajustar adecuadamente las muestras de entrenamiento, lo que se denomina subajuste. Por otro lado, un polinomio de grado 4 se acerca de manera casi precisa a la función real. No obstante, al utilizar grados polinómicos más altos, los modelos tienden a sobreajustar los datos de entrenamiento al capturar incluso el ruido presente en dichos datos.

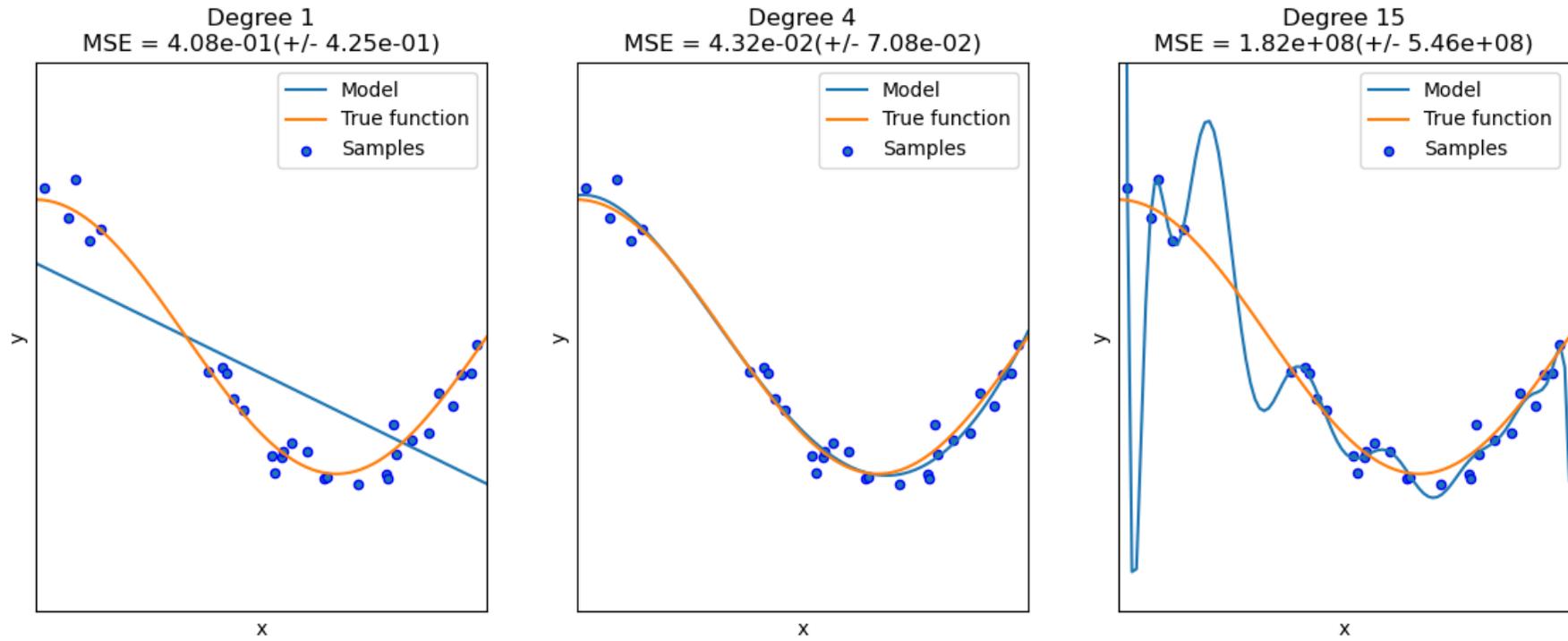


Figura 10. Ejemplo de Sobreajuste y Subajuste  
 Fuente: Scikit-learn, 2020.

El objetivo es obtener un modelo de alta calidad que presente un bajo error tanto en los datos de entrenamiento como en los datos de prueba, como se observa en la Figura 11 (Castillo, 2022).

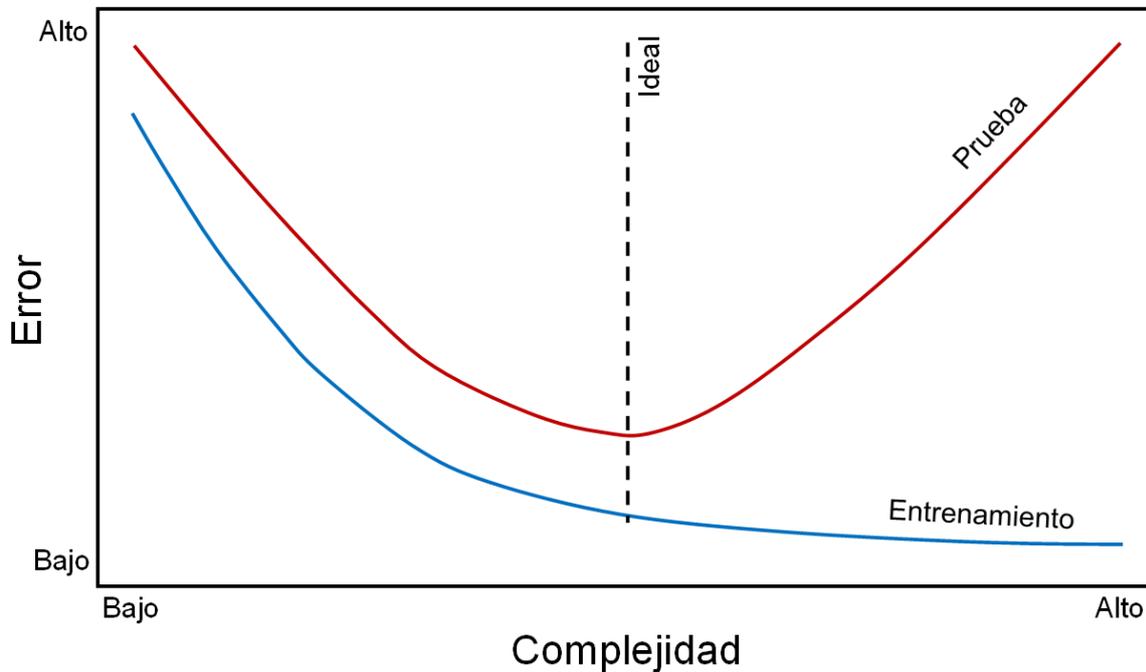


Figura 11. Comportamiento del Error frente al Sobreajuste  
Fuente: Adaptado de Scikit-learn por Castillo, 2022.

#### 2.2.3.4. Técnicas de Mejora de Modelos Predictivos

##### a. División de Datos

Uno de los métodos más clásicos y comunes para evaluar cómo un nuevo modelo de aprendizaje automático se comportará con datos no vistos es el llamado método de retención (holdout method). Este consiste en subdividir el conjunto de datos original en dos partes: un conjunto de entrenamiento, utilizado para ajustar el modelo, y un conjunto de prueba, que se usa para analizar su desempeño frente a datos nuevos. Además, en la práctica, también nos interesa ajustar y comparar las diferentes combinaciones de parámetros (hiperparámetros) para mejorar la precisión del modelo en la predicción de nuevos datos (Raschka y Mirjalili, 2019).

El problema surge cuando usamos repetidamente el mismo conjunto de prueba durante este proceso de selección de modelos, lo que puede hacer que el modelo se adapte mucho a los datos de prueba, llevándolo al sobreajuste. Para evitar esto, una mejor alternativa es dividir los datos en tres partes: un conjunto de entrenamiento, un conjunto de validación, y un conjunto de prueba. El conjunto de validación se utiliza para analizar distintas configuraciones del modelo, y el conjunto de prueba se reserva para la evaluación final, garantizando que el modelo no lo haya visto previamente. Esto proporciona una estimación más confiable de la capacidad del modelo para generalizar (Raschka y Mirjalili, 2019).

La figura siguiente muestra cómo funciona el método de retención, donde se utiliza un conjunto de datos de validación para examinar repetidamente el desempeño del modelo después de haberlo entrenado, probando diferentes configuraciones de hiperparámetros. Una vez que estamos contentos con cómo hemos ajustado estos valores, pasamos a estimar qué tan bien puede generalizar el modelo utilizando el conjunto de datos de prueba (Raschka y Mirjalili, 2019).

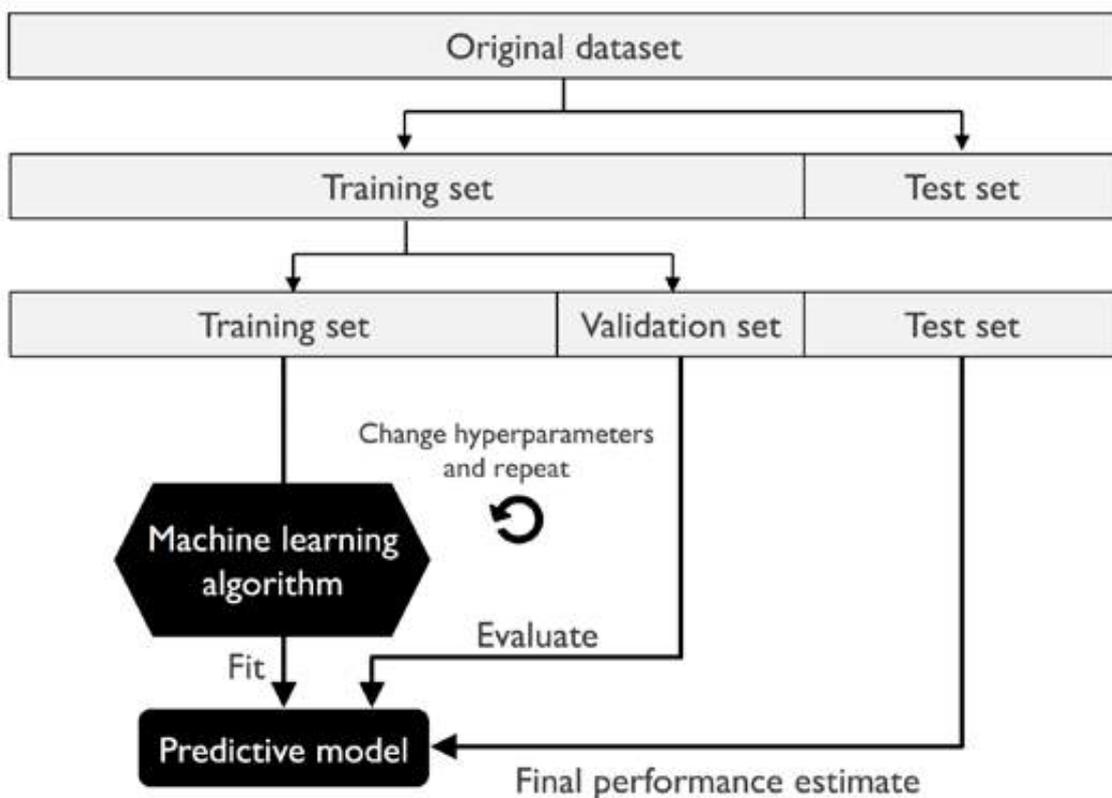


Figura 12. Método de Retención  
Fuente: Raschka y Mirjalili, 2019.

Scikit-learn (2020) indica que al dividir los datos en tres conjuntos (entrenamiento, validación y prueba), se reduce significativamente la cantidad de muestras que se pueden utilizar para entrenar el modelo. Esto puede ser problemático, especialmente si el conjunto de datos es pequeño, ya que la cantidad de datos de entrenamiento disponible puede ser insuficiente para que el modelo aprenda de manera efectiva. Por otro lado, los resultados del modelo pueden depender de una elección aleatoria específica para el par de conjuntos de entrenamiento y validación, lo cual puede inducir a una división no representativa de los datos y por ende se hace una mala interpretación del rendimiento del modelo. Una solución para este problema es la validación cruzada k-fold, ya que cuando se emplea esta técnica ya no es necesario un conjunto de validación.

La validación cruzada k-fold se basa en fraccionar el conjunto de datos de entrenamiento en k pliegues (conjuntos), donde cada pliegue se utiliza una vez como conjunto de prueba y los k-1 pliegues restantes se usan como conjunto de entrenamiento. Se realizan k iteraciones, entrenando y evaluando el modelo en distintas conjugaciones de pliegues, esto permite obtener una medida de rendimiento promedio y una estimación más precisa de cómo se comportará el modelo en datos que no visto antes (Scikit-learn, 2020). La validación cruzada ayuda a ajustar los hiperparámetros de manera efectiva, mejorando así, la precisión y robustez del modelo (Raschka y Mirjalili, 2019).

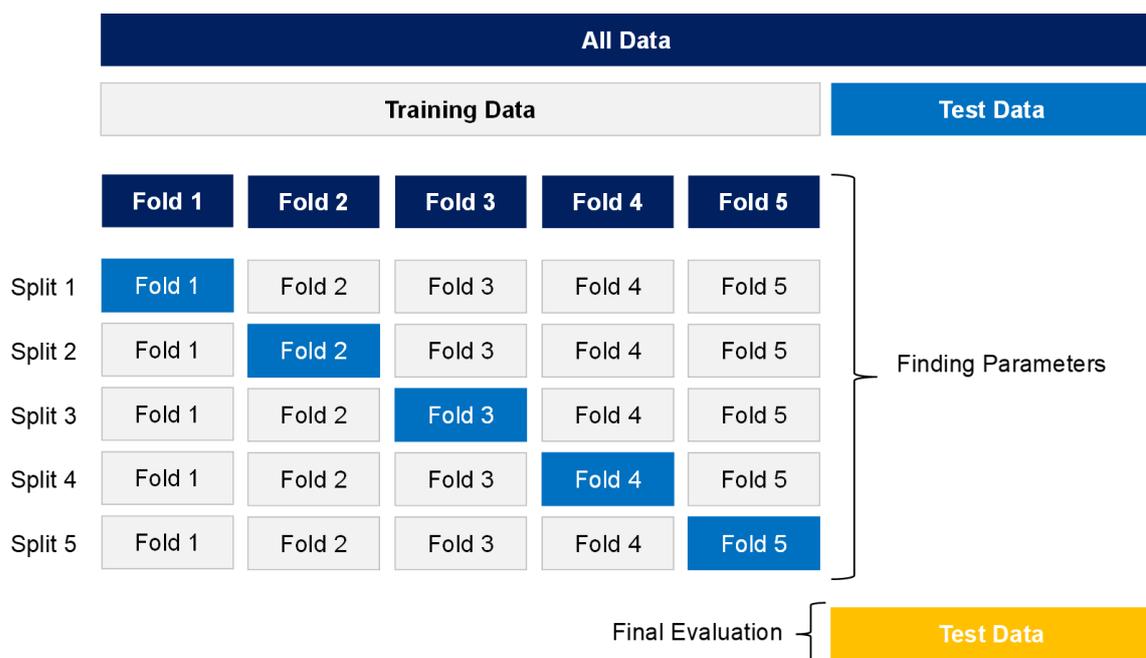


Figura 13. Método de la Validación Cruzada  
Fuente: Adaptado de Scikit-learn, 2020.

La configuración de un valor estándar para “k” en la validación cruzada k-fold de 10 es recomendable, ya que estudios empíricos han demostrado que proporciona un buen equilibrio entre sesgo y varianza. No obstante, si se cuenta con conjuntos de entrenamiento pequeños, puede ser útil aumentar el número de pliegues, a fin de usar más datos en cada iteración y, por lo tanto, reducir el sesgo en la estimación del rendimiento del modelo. Por otro lado, tener un “k” mayor también aumenta el tiempo de ejecución y puede aumentar la varianza de las estimaciones, ya que los pliegues de entrenamiento serán más similares entre sí. Con conjuntos de datos grandes, se puede optar por un “k” más pequeño, por ejemplo:  $k = 5$ , permitiendo obtener estimaciones precisas del rendimiento promedio del modelo a un costo computacional menor (Raschka y Mirjalili, 2019).

#### **b. Estandarización de Datos**

Es un proceso esencial en el preprocesamiento de datos para varios algoritmos de aprendizaje automático. Este proceso ajusta la escala de cada variable para que se encuentre en un rango de 0 a 1. Aunque algunos algoritmos no requieren este escalado, otros lo necesitan para acelerar el proceso de aprendizaje y lograr un ajuste del modelo diferente; además, trabajar con datos no estandarizados puede afectar significativamente el rendimiento y la exactitud de los resultados en algunos casos (Scikit-learn, 2020).

#### **c. Determinación de Hiperparámetros**

Los hiperparámetros son aquellos parámetros que no se ajustan de forma directa en los modelos y se introducen como argumentos al definirlos. Es crucial determinar estos hiperparámetros, ya que optimizarlos correctamente puede mejorar notablemente el rendimiento del modelo. La búsqueda de los mejores valores para estos hiperparámetros se realiza a través de técnicas de validación cruzada, siendo GridSearchCV (búsqueda de cuadrícula) uno de los métodos más destacados, ya que evalúa todas las conjugaciones posibles de parámetros (Scikit-learn, 2020).

#### d. Selección de Variables Relevantes

Para Guyon y Elisseeff, citados por Fosca (2020), la selección adecuada de variables permite optimizar la capacidad predictiva del modelo, ofrecer soluciones más rápidas y proporcionar una comprensión más profunda del proceso que generó los datos. Al seleccionar las variables con cuidado, se pueden mejorar significativamente los hallazgos y, al mismo tiempo, simplificar el modelo, haciéndolo más fácil de entender y explicar.

El hallazgo del grupo óptimo de variables requiere la verificación de todas las conjugaciones posibles de acuerdo con estas variables, lo cual sería computacionalmente muy difícil de realizar. En la mayoría de los modelos, la función que representa cómo cada variable se vincula con el resultado es extremadamente compleja y difícil de comprender para determinar cómo se conecta con el objetivo final (Fosca, 2020).

Entre los métodos más populares para seleccionar variables relevantes se encuentra la Eliminación Recursiva de Características (RFE). Este es un método de tipo envoltura, lo que significa que se basa en excluir las variables cuyo rendimiento es deficiente en el modelo en construcción. Como su nombre lo indica, es recursivo porque el proceso se realiza de manera iterativa hasta obtener el resultado deseado, en esencia, es una selección hacia atrás de las variables. El RFE comienza construyendo un modelo que utiliza todas las variables para calcular la puntuación de importancia de cada una. Luego, se eliminan las variables poco relevantes, se reconstruye el modelo y se recalculan las puntuaciones de importancia. Durante este proceso, es necesario determinar el número y tamaño de los subgrupos de variables a evaluar, siendo este un parámetro ajustable del método. Finalmente, se selecciona el subgrupo óptimo, basado en las puntuaciones de importancia, para entrenar el modelo definitivo (Fosca, 2020).

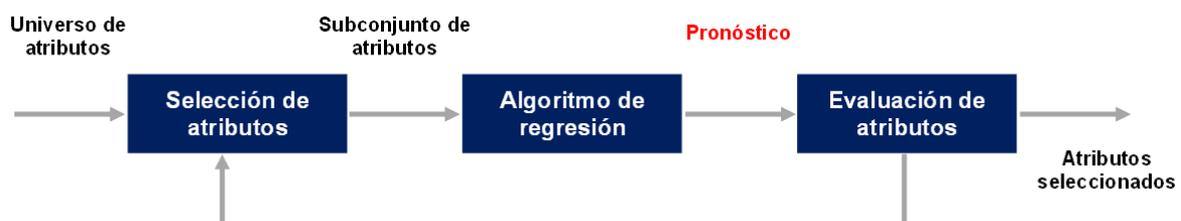


Figura 14. Esquema de Método de RFE  
Fuente: Adaptado de Fosca, 2020.

## 2.2. DEFINICIÓN DE TÉRMINOS BÁSICOS

**Oferta:** Es el conjunto de bienes o productos disponibles en el mercado, presentados con un precio específico en un momento determinado (Real Academia Española, 2023).

**Demanda:** Se refiere al total de bienes y servicios que una comunidad o grupo de personas tiene la intención de comprar o que ya han comprado (Real Academia Española, 2023).

**Fluctuación:** Se entiende como la acción y efecto de fluctuar. En otras palabras, la fluctuación hace referencia al proceso y resultado de experimentar variaciones o cambios continuos (Real Academia Española, 2023).

**Base de datos:** Se trata de un conjunto de datos estructurados que permite atender a múltiples aplicaciones simultáneamente mediante almacenamiento y gestión individual; permite acceder a los datos necesarios sin necesidad de generar documentos o especificaciones de datos distintos en los sistemas informáticos (Laudon y Laudon, 2012).

**Algoritmo:** Es una fórmula o conjunto de reglas, procedimientos o instrucciones destinadas a solucionar una situación o ejecutar una tarea determinada. En el ámbito de la Inteligencia Artificial, el algoritmo guía a la máquina para encontrar respuestas a preguntas o soluciones a problemas planteados (Guo et al., 2019).

**Inteligencia Artificial:** También llamada inteligencia de máquina, se refiere a sistemas que demuestran capacidades inteligentes al evaluar su entorno y realizar acciones, con un nivel de autonomía, para alcanzar determinadas metas (Guo et al., 2019).

**Aprendizaje Automático:** Es un área de la informática que desarrolla modelos computacionales capaces de "aprender" de los datos y realizar predicciones. Estos modelos examinan patrones en la información para adaptarse a ellos y así proporcionar resultados a partir de nuevos datos que se les introduzcan (Guo et al., 2019).

**Hiperparámetros:** Son parámetros que no se ajustan automáticamente al entrenar un modelo, sino que se establecen antes de empezar y son muy importantes porque afectan el rendimiento del modelo (Scikit-learn, 2020).

**Clasificación:** Es un método utilizado para agrupar muestras mediante la asignación de una etiqueta o categoría (Guo et al., 2019).

**Rregresión:** Es un método que se utiliza para predecir un valor dentro de un rango continuo. Es parte del aprendizaje supervisado y ayuda a estimar resultados basados en datos de entrada (Guo et al., 2019).

**Agrupamiento:** También conocido como clustering, es una tarea que consiste en organizar datos en grupos según ciertas características. Este análisis se usa mucho para reconocer patrones, análisis de imágenes y gráficos por computadora, entre otras aplicaciones (Guo et al., 2019).

**Redes Neuronales:** También llamadas redes neuronales artificiales o redes profundas, son sistemas computacionales basados la estructura del cerebro humano. Básicamente, son modelos de machine learning que procesan información de forma similar a como lo hacen las neuronas en el cerebro, ayudando a reconocer patrones y tomar decisiones (Guo et al., 2019).

**Sobreajuste:** Más conocido como overfitting, ocurre cuando un modelo es demasiado complejo y se ajusta demasiado a los errores del conjunto de datos de entrenamiento, lo que lo hace impreciso para predecir nuevos datos (Guo et al., 2019).

**Subajuste:** Más conocido como underfitting, ocurre cuando un modelo no consigue procesar bien la información que tiene, por lo que no representa correctamente lo que los datos realmente muestran (Guo et al., 2019).

## **CAPÍTULO III**

### **MATERIALES Y MÉTODOS**

#### **3.1. UBICACIÓN DE LA INVESTIGACIÓN**

La presente investigación se desarrolla en la ciudad de Cajamarca, Perú, y aborda un análisis global del mercado del oro utilizando una perspectiva integral. Desde esta ubicación, se examinan las siguientes variables que tienen una relación con el precio del oro: PBI de Estados Unidos, Inflación de Estados Unidos, Tasa de Participación Laboral de Estados Unidos, Tasa de Interés de Reino Unido, Índice del Precio del Dólar, Índice S&P 500, Dow Jones Industrial Average, iShares Silver Trust ETF, VanEck Gold Miners ETF, Precio de las acciones de Eldorado Gold Corporation, United States Oil ETF, Producción Mundial de Oro, Demanda Mundial de Oro, Precio del Petróleo, Costo de Producción de Oro, Tipo de Cambio Euro/Libra y Tipo de Cambio Euro/Yen.

#### **3.2. METODOLOGÍA DE LA INVESTIGACIÓN**

##### **3.2.1. Tipo, Nivel, Diseño y Enfoque de Investigación**

Siguiendo las ideas expuestas por Hernández (2014), encontramos que:

- ✓ **Tipo:** El tipo de la investigación es aplicada, porque se va abordar un problema práctico y real, que es el desarrollo de un modelo predictivo.
- ✓ **Nivel:** El nivel de la investigación es exploratorio porque se emplearán diferentes algoritmos y técnicas de aprendizaje automático para identificar el más adecuado, además se dejará un campo abierto para nuevos estudios. También es correlacional porque se asocian variables mediante un patrón predecible. Asimismo, es descriptivo, ya que busca especificar y medir las propiedades de las variables relacionadas con la fluctuación del precio del oro, permitiendo entender mejor su contexto y dimensiones.

- ✓ **Diseño:** El diseño de la investigación es no experimental – transversal, debido a que no se va a manipular deliberadamente las variables de investigación y los datos se recolectarán en un momento único.
- ✓ **Enfoque:** El enfoque de la investigación es cuantitativo, porque implica recopilar datos con el propósito de comprobar la hipótesis mediante mediciones numéricas y análisis estadísticos.

### **3.2.2. Población de Estudio**

Datos históricos del precio del oro comprendidos entre el 01 de enero del 2012 y el 31 de diciembre del 2022.

### **3.2.3. Muestra**

Datos históricos del precio del oro comprendidos entre el 01 de enero del 2012 y el 31 de diciembre del 2022.

### **3.2.4. Unidad de Análisis**

Factores que influyen y tienen relación con la fluctuación del precio del oro, como: PBI de Estados Unidos, Inflación de Estados Unidos, Tasa de Participación Laboral de Estados Unidos, Tasa de Interés de Reino Unido, Índice del Precio del Dólar, Índice S&P 500, Dow Jones Industrial Average, iShares Silver Trust ETF, VanEck Gold Miners ETF, Precio de las acciones de Eldorado Gold Corporation, United States Oil ETF, Producción Mundial de Oro, Demanda Mundial de Oro, Precio del Petróleo, Costo de Producción de Oro, Tipo de Cambio Euro/Libra y Tipo de Cambio Euro/Yen.

### 3.2.5. Definición de Variables

#### 3.2.5.1. Independientes

- ✓ **Variables Macroeconómicas:** PBI de Estados Unidos, Inflación de Estados Unidos, Tasa de Participación Laboral de Estados Unidos, Tasa de Interés de Reino Unido e Índice del Precio de Dólar.
- ✓ **Variables Financieras - Acciones y ETFs:** Índice S&P 500, Dow Jones Industrial Average, iShares Silver Trust ETF, VanEck Gold Miners ETF, Precio de las acciones de Eldorado Gold Corporation y United States Oil ETF.
- ✓ **Variables de Mercado de Materias Primas:** Producción Mundial de Oro, Demanda Mundial de Oro, Precio del Petróleo, Costo de Producción de Oro
- ✓ **Variables de Mercado de Divisas:** Tipo de Cambio Euro/Libra y Tipo de Cambio Euro/Yen.

#### 3.2.5.2. Dependientes

Precio futuro del oro

### 3.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

#### 3.3.1. Técnicas

**Datos Secundarios:** Implica la revisión de documentos, registros públicos y archivos físicos o electrónicos que han sido previamente creados con fines distintos a la investigación actual (Hernández et al., 2014). Esta técnica ahorra tiempo y recursos al utilizar información ya disponible, pero es esencial evaluar la calidad y confiabilidad estos datos, para asegurar su validez en el nuevo contexto de investigación.

### 3.3.2. Instrumentos

- ✓ **Plataformas Financieras:** Acceso a plataformas en línea que proporcionan información histórica y actual sobre el precio del oro y otros indicadores financieros relevantes.
- ✓ **Bases de Datos Económicos y Financieros:** Utilización de fuentes de datos económicos y financieros que almacenan indicadores macroeconómicos, tasas de interés, inflación, políticas monetarias y otros datos que puedan afectar el precio del oro.
- ✓ **Informes de Analistas y Expertos del Mercado:** Utilización de informes y análisis elaborados por expertos de la industria financiera y del mercado del oro, que a menudo incluyen interpretaciones y perspectivas sobre los factores que afectan el precio.

### 3.3.3. Materiales y Equipos

- ✓ **Conexión a Internet:** Para descargar información y acceder a recursos en línea para el aprendizaje y la investigación.
- ✓ **Laptop:** Con suficiente potencia de procesamiento y memoria para ejecutar métodos y modelos de aprendizaje automático, y para llevar a cabo actividades como procesar información, análisis exploratorio, preparación de datos y generar informes a través de programas especializados.

### 3.3.4. Softwares

- ✓ **Microsoft Excel:** Se utilizó para almacenar, estructurar y manipular datos importantes para el desarrollo de la base de datos para el modelo predictivo.
- ✓ **Microsoft Word:** Para analizar y evaluar los resultados obtenidos en este estudio.
- ✓ **Google Colaboratory:** Conocido también como Google Colab, es un entorno gratuito en línea creado por Google. Se utilizó para emplear el lenguaje Python, permitiendo el acceso a recursos computacionales para la exploración, análisis y preprocesamiento de la información, y para el desarrollo del modelo predictivo en general.

### 3.4. PROCEDIMIENTOS

El modelo predictivo para la fluctuación del precio del oro se desarrollará siguiendo el flujograma presentado a continuación, el cual ha sido estructurado a partir de una revisión bibliográfica detallada:

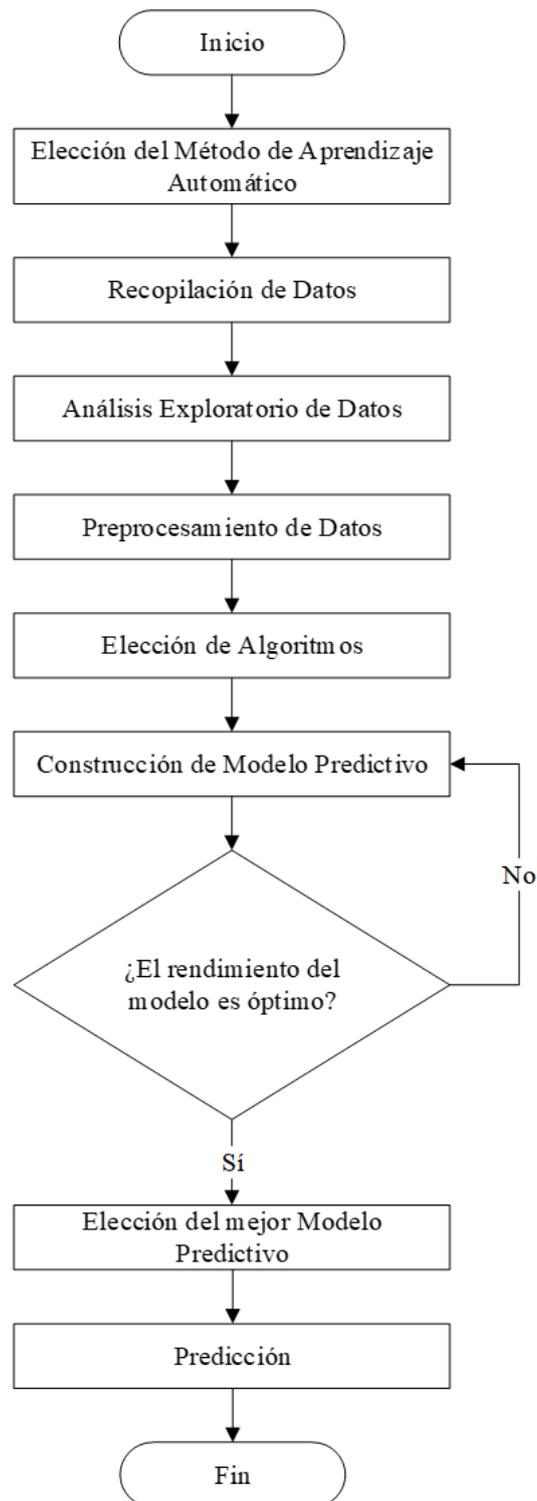


Figura 15. Flujograma para el Desarrollo del Modelo Predictivo

### 3.4.1. Elección del Método de Aprendizaje Automático

En esta etapa se seleccionará el método de aprendizaje automático (machine learning) más adecuado, tomando en cuenta las características de los datos y el propósito del estudio. Los métodos se dividen en tres grupos principales, como se presenta en la tabla a continuación:

Tabla 1. Métodos de Aprendizaje Automático

Método	Tipo de Data	Análisis	Objetivo	Enfoque
Aprendizaje Supervisado	Datos de entrada y salida son conocidos y pre-categorizados	Offline	El resultado se predice utilizando la data de entrada categorizada	- Regresión - Clasificación
Aprendizaje no Supervisado	Solo los datos de entrada son conocidos	En tiempo real	El resultado se predice en base a los patrones en los datos de entrada	- Clustering - Representation Learning
Aprendizaje por reforzamiento	Data no predefinida	En tiempo real y con interacción de un agente	El resultado se predice en base a patrones y al enfoque de prueba y error	- Optimización de estrategias - Behavior learning

Fuente: Adaptado de Fosca, (2020).

### 3.4.2. Recopilación de Datos

En esta fase, se definirá el período y la frecuencia (diariamente, mensualmente o trimestralmente) de los datos que se van a usar, luego se identificarán las fuentes disponibles para obtener estos datos, y como último paso se organizarán los datos recopilados según el período y la frecuencia previamente establecidos, para crear una base de datos ordenada que se usará para construir el modelo predictivo.

### 3.4.3. Análisis Exploratorio de Datos

En esta parte se revisarán y examinarán los datos recopilados para comprender su estructura y características, este paso es muy importante para reconocer tendencias, patrones y cualquier anomalía en los datos.

### 3.4.4. Preprocesamiento de Datos

El preprocesamiento de datos es una de las etapas más importantes en cualquier aplicación de machine learning o aprendizaje automático, ya que con frecuencia los datos sin procesar no tienen la estructura ni organización adecuada para lograr un rendimiento óptimo del algoritmo (Raschka y Mirjalili, 2019). Este proceso asegura la calidad de los datos, contribuyendo a obtener un modelo robusto.

En esta etapa se hará lo siguiente:

- ✓ **Limpieza de Datos:** Se identificarán y tratarán valores atípicos, datos faltantes o inconsistentes en el conjunto de datos.
- ✓ **Estandarización de Datos:** Aquí todos los datos de las variables se ajustarán a una escala común, para facilitar su comparación y evitar que alguna variable tenga más peso que otra.
- ✓ **Agregación de Retraso Temporal (lag):** Se añadirán al modelo valores pasados en función de las variables de estudio, lo cual es fundamental en el análisis de series de tiempo, ya que permite comprender cómo los datos históricos influyen en el comportamiento futuro.
- ✓ **División de Datos:** Se dividirá el conjunto de datos en dos grupos: uno para entrenar el modelo y otro para evaluarlo. Este proceso es clave para asegurar que el modelo aprenda de los datos y se pruebe con información nueva, ayudando a evitar el sobreajuste y garantizando que el modelo pueda generalizar correctamente a nuevos datos.

### 3.4.5. Elección de Algoritmos

En este paso se elegirán tres algoritmos para poder entrenarlos, ya que según Raschka y Mirjalili (2019) es imprescindible comparar al menos un puñado de algoritmos diferentes para entrenar y seleccionar el modelo con mejor rendimiento. La elección de los algoritmos se hará en base al método de aprendizaje automático que se eligió previamente.

### 3.4.6. Construcción de Modelo Predictivo

La construcción del modelo predictivo se hará en base a los siguientes puntos clave:

- ✓ **Determinación de Hiperparámetros:** Se elegirán los valores más adecuados para los hiperparámetros que controlan el funcionamiento de los algoritmos, con el fin de asegurar que el modelo funcione de la mejor manera posible.
- ✓ **Entrenamiento de los Algoritmos:** Se entrenarán los algoritmos usando el conjunto de datos de entrenamiento, para que puedan identificar vínculos entre las variables y de esta manera hagan buenas predicciones.
- ✓ **Evaluación de Algoritmos:** Aquí se evaluará el desempeño de los algoritmos utilizando métricas como: MAE, MSE, RMSE y  $R^2$ , para identificar el algoritmo más efectivo y medir su capacidad de predecir con nuevos datos.

### 3.4.7. Elección del mejor Modelo Predictivo

En esta fase, los algoritmos empleados se compararán en función de las métricas de evaluación, con la finalidad de elegir el modelo con mejor rendimiento en la fase de prueba.

### 3.4.8. Predicción

Después de seleccionar el mejor modelo predictivo, se realizarán estimaciones con datos nuevos que no forman parte del conjunto general. De esta manera, el modelo aplicará lo aprendido durante el entrenamiento para generar resultados a partir de patrones que ya conoce. El objetivo será evaluar cómo se desempeña el modelo en situaciones reales y asegurar que las predicciones sean útiles y precisas.

### **3.5. TRATAMIENTO, ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS**

#### **3.5.1. Elección del Método de Aprendizaje Automático**

Según las características de cada método de aprendizaje automático, se determinó que el método a utilizar en este estudio será el aprendizaje supervisado, porque:

- ✓ Los datos son continuos y han sido etiquetados o clasificados, es decir, los datos de entrada (input) y salida (output) son conocidos.
- ✓ El dato de salida (output) es el precio futuro del oro, y es de tipo numérico; por lo tanto, el enfoque será de tipo regresión.

#### **3.5.2. Recopilación de Datos**

La recopilación de datos se hizo desde el 01 de enero de 2012 hasta el 31 de diciembre de 2022, teniendo como objetivo obtener los datos con una frecuencia diaria, sin embargo, en algunas variables sólo se pudo encontrar datos históricos con una frecuencia mensual y en otros de manera trimestral. Es por esto que al momento de estructurar la base de datos se tuvo que mantener los valores constantes de los registros que tenían frecuencia mensual y trimestral, en relación de las fechas de los registros que tenían frecuencia diaria. En la siguiente tabla se muestran las variables con su respectiva frecuencia y fuente de donde fueron descargados los datos.

Tabla 2. Variables Recolectadas

Variable	Abreviatura	Unidad	Frecuencia	Fuente
Precio del Petróleo	Prec Petróleo	USD/Barril	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
PBI de Estados Unidos	PBI EE. UU.	Miles de millones de dólares	Trimestral	<a href="https://fred.stlouisfed.org/">https://fred.stlouisfed.org/</a>
Producción Mundial de Oro	Prod Mund Oro	Toneladas	Trimestral	<a href="https://www.gold.org/">https://www.gold.org/</a>
Demanda Mundial de Oro	Dema Mund Oro	Toneladas	Trimestral	<a href="https://www.gold.org/">https://www.gold.org/</a>
Índice del Precio de Dólar	Índ Prec Dólar	Unidad	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Inflación de Estados Unidos	Infla EE. UU.	Unidad	Mensual	<a href="https://fred.stlouisfed.org/">https://fred.stlouisfed.org/</a>
Índice S&P 500	Índ S&P 500	Unidad	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Costo de Producción de Oro	Cost Prod Oro	USD/onza troy	Trimestral	<a href="https://www.gold.org/">https://www.gold.org/</a>
Rentabilidad del Bono de Estados Unidos	Ren Bono EE. UU.	Unidad	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Tasa de Participación Laboral de Estados Unidos	Tas Par Lab EE. UU.	Porcentaje (%)	Mensual	<a href="https://fred.stlouisfed.org/">https://fred.stlouisfed.org/</a>
Tipo de Cambio Euro/Libra	Euro/Libra	Unidad	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Tipo de Cambio Euro/Yen	Euro/Yen	Unidad	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
iShares Silver Trust ETF	SVL ETF	USD	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Dow Jones Industrial Average	Dow Jones IA	Unidad	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Tasa de Interés de Reino Unido	Tasa Int R. U.	Porcentaje (%)	Mensual	<a href="https://fred.stlouisfed.org/">https://fred.stlouisfed.org/</a>
VanEck Gold Miners ETF	GDX ETF	USD	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Precio de las Acciones de Eldorado Gold Corporation	Accio EGO	USD	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
United States Oil Fund ETF	USO ETF	USD	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>
Precio del Oro	Prec Oro	USD/onza troy	Diaria	<a href="https://es.investing.com/">https://es.investing.com/</a>

La elaboración de la base de datos se hizo con el apoyo de la herramienta Excel, donde primero se organizaron los datos de manera individual a través de tablas, para luego estructurar la base de datos aplicando un análisis relacional entre las tablas en función a la fecha de cada registro.

La base de datos construida se presentará en un CD por razones de espacio. Para dar una idea de su contenido, se ha elaborado un resumen mensual, que se puede revisar en el Anexo N°6.

### **3.5.3. Análisis Exploratorio de Datos**

Para la ejecución de esta etapa y la siguientes se va a emplear la herramienta de Google Colaboratory para poder analizar los datos y desarrollar el modelo predictivo a través del lenguaje Python. Antes de realizar el análisis exploratorio se cargó la base de datos estructurada (que por razones de espacio se presentó en un CD) en la fase de recolección de datos, asignándole el nombre de “df”. A continuación, se detalla los pasos realizados para esta etapa:

#### **3.5.3.1. Descripción General de los Datos**

Primero se revisaron los nombres de las columnas para prevenir posibles inconvenientes más adelante al trabajar con los datos. Por suerte, no hubo complicaciones en este sentido, ya que todos los campos estaban bien nombrados. Para llevar a cabo esta revisión, se utilizó la función `.head()`, tal como se presenta en la Figura 16.

df.head()



	Fecha	Prec Petróleo	PBI EE.UU.	Prod Mund Oro	Dema Mund Oro	Índ Prec Dólar	Infla EE.UU.	Índ S&P 500	Cost Prod Oro	Ren Bono EE. UU.	Tas Par Lab EE. UU.	Euro / Libra	Euro / Yen	SVL ETF	Dow Jones IA	Tasa Inte R. U.	GDX ETF	Accio EGO	USO ETF	Prec Oro
0	2012-01-03	102.9	16068.8	1086.8	1187.9	79.6	227.8	1277.1	940	2.0	63.7	0.8	100.1	28.8	12397.4	2.1	53.8	73.7	317.5	1604.1
1	2012-01-04	103.3	16068.8	1086.8	1187.9	80.1	227.8	1277.3	940	2.0	63.7	0.8	99.3	28.4	12418.4	2.1	53.7	73.1	318.2	1612.4
2	2012-01-05	101.3	16068.8	1086.8	1187.9	80.9	227.8	1281.1	940	2.0	63.7	0.8	98.7	28.5	12415.7	2.1	53.9	73.2	313.4	1623.0
3	2012-01-06	101.9	16068.8	1086.8	1187.9	81.2	227.8	1277.8	940	2.0	63.7	0.8	97.9	27.9	12359.9	2.1	53.4	71.9	313.8	1616.8

Figura 16. Visualización de Datos

Luego se hizo una identificación del número de filas y columnas con la ayuda de la función .shape, donde se puede se puede observar (ver Figura 17) que se cuenta con 2764 filas y 20 columnas.

df.shape



(2764, 20)

Figura 17. Número de Filas y Columnas

Posteriormente se identificó el tipo de variables que se tiene en la base de datos, para esto se utilizó la función `.dtypes`, logrando identificar que hay tres tipos de variables (ver Figura 18), las cuales son:

- ✓ **Fecha y hora (datetime):** Indica que los datos de esa columna están en formato de fecha y hora.
- ✓ **Decimal (float):** Indica que los datos de esa columna son números decimales, es decir, números reales.
- ✓ **Entero (int):** Indica que los datos de esa columna son números enteros.

```
df.dtypes
```

Fecha	datetime64[ns]
Prec Petróleo	float64
PBI EE.UU.	float64
Prod Mund Oro	float64
Dema Mund Oro	float64
Índ Prec Dólar	float64
Infla EE.UU.	float64
Índ S&P 500	float64
Cost Prod Oro	int64
Ren Bono EE. UU.	float64
Tas Par Lab EE. UU.	float64
Euro / Libra	float64
Euro / Yen	float64
SVL ETF	float64
Dow Jones IA	float64
Tasa Inte R. U.	float64
GDX ETF	float64
Accio EGO	float64
USO ETF	float64
Prec Oro	float64
dtype:	object

Figura 18. Tipo de Variables

También se verificó, con la ayuda de la función `.isnull().sum()`, si existen valores faltantes por cada variable; en la Figura 19 se puede visualizar que el número "0" aparece frente a cada variable, lo cual indica que no hay valores ausentes en ninguna de ellas. Esto confirma que los datos están completos y no necesitan ningún tratamiento adicional respecto a valores nulos, lo cual asegura la calidad de la información antes de continuar con el análisis.

```
df.isnull().sum()
Fecha 0
Prec Petróleo 0
PBI EE.UU. 0
Prod Mund Oro 0
Dema Mund Oro 0
Índ Prec Dólar 0
Infla EE.UU. 0
Índ S&P 500 0
Cost Prod Oro 0
Ren Bono EE. UU. 0
Tas Par Lab EE. UU. 0
Euro / Libra 0
Euro / Yen 0
SVL ETF 0
Dow Jones IA 0
Tasa Inte R. U. 0
GDX ETF 0
Accio EGO 0
USO ETF 0
Prec Oro 0
dtype: int64
```

Figura 19. Número de Valores Faltantes por Variable

### 3.5.3.2. Resumen Estadístico de Variables

Se hizo un análisis estadístico de las variables numéricas, esto quiere decir que no se tomó en cuenta la variable fecha ya que es de tipo datetime, para esto se usó la función `.describe().round()` de la librería pandas. En este análisis (ver Figura 20) se puede visualizar la cantidad de datos (count), la media (mean), la desviación estándar (std), el valor mínimo (min), valor máximo (max) y los cuartiles (Q1=25%, Q2=50% y Q3=50%).

```
df.describe().round(1)
```



	Prec Petróleo	PBI EE.UU.	Prod Mund Oro	Dema Mund Oro	Índ Prec Dólar	Infla EE.UU.	Índ S&P 500	Cost Prod Oro	Ren Bono EE. UU.	Tas Par Lab EE. UU.	Euro / Libra	Euro / Yen	SVL ETF	Dow Jones IA	Tasa Inte R. U.	GDX ETF	Accio EGO	USO ETF	Prec Oro
<b>count</b>	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0	2764.0
<b>mean</b>	67.7	19894.3	1163.4	1086.4	92.5	250.1	2620.1	1005.9	2.1	62.7	0.8	127.0	19.2	22489.3	1.6	28.1	20.9	136.3	1458.6
<b>std</b>	24.4	2713.5	65.7	115.5	7.8	18.1	919.4	98.3	0.7	0.7	0.1	10.7	5.0	6907.4	0.8	9.2	18.5	92.5	253.8
<b>min</b>	7.8	16068.8	1041.5	773.2	78.3	227.8	1277.1	918.0	0.5	60.1	0.7	94.3	11.2	12101.5	0.2	12.5	2.6	17.0	1051.7
<b>25%</b>	49.4	17852.5	1116.5	1032.0	85.4	236.9	1951.3	935.0	1.7	62.4	0.8	121.6	15.5	16718.8	1.0	22.0	8.0	75.7	1251.7
<b>50%</b>	62.7	19561.9	1156.8	1097.1	94.5	244.2	2432.8	960.0	2.1	62.8	0.9	128.8	17.3	21354.8	1.5	25.5	13.3	94.2	1334.1
<b>75%</b>	91.8	21538.0	1224.9	1144.1	97.4	258.2	3130.0	1040.0	2.6	63.0	0.9	133.8	22.1	27172.4	2.0	32.9	29.5	248.9	1712.0
<b>max</b>	126.5	26138.0	1301.6	1302.8	114.1	299.0	4796.6	1262.0	4.2	63.8	0.9	149.2	35.8	36799.6	4.1	57.5	79.8	336.1	2063.8

Figura 20. Análisis Estadístico de Variables Numéricas

A partir del análisis descriptivo antes realizado, se puede observar que en varios de los casos la media es cercana al cuartil 50%, lo cual sugiere que la distribución de los datos está cercana a la simetría. Por otro lado, se observa que los datos de cada variable tienen una escala diferente, lo cual podría afectar el rendimiento del modelo predictivo. Para abordar este problema de escala, podemos aplicar la técnica de estandarización, que se detallará más adelante.

### 3.5.3.3. Visualización de Relaciones y Patrones

En este estudio se utilizaron gráficos de histogramas y la matriz de Pearson para poder visualizar las relaciones y patrones de los datos. A continuación, se describe el análisis de cada herramienta en base a las variables de estudio:

Primero se empleó los gráficos de histogramas para obtener una mejor visualización de la distribución de los datos, como se muestra en la Figura 21, donde se puede observar varios tipos de distribuciones en base al histograma de cada variable; por ejemplo, la distribución del Precio del Petróleo es de tipo bimodal, la del iShares Silver Trust ETF es de tipo sesgada a la izquierda y la del Dow Jones Industrial Average es de tipo multimodal. Según Fosca (2020), cuando el la distribución de las variables es gaussiana o normal, ayuda a que los resultados de predeción de una regresión lineal sean más confiables. En este caso se puede ver a simple vista que la mayoría de las variables no tiene una distribución normal, a excepción del tipo de cambio Euro/Yen, Tasa de Interés del Reino Unido, Demanada y Producción Mundial de Oro, que mas o menos se asemejan a dicha distribución.

También se puede observar algunos valores atípicos en los histogramas, como por ejemplo, en la Demanda Mundial del Oro y más aún en la Tasa de Participación Laboral de Estados Unidos; éstos valores atípicos están representados por las barras aisladas en la parte izquierda de la distribución. El tratamiento de estos valores atípicos se detallara en la fase de limpieza de datos del preprocesamiento.

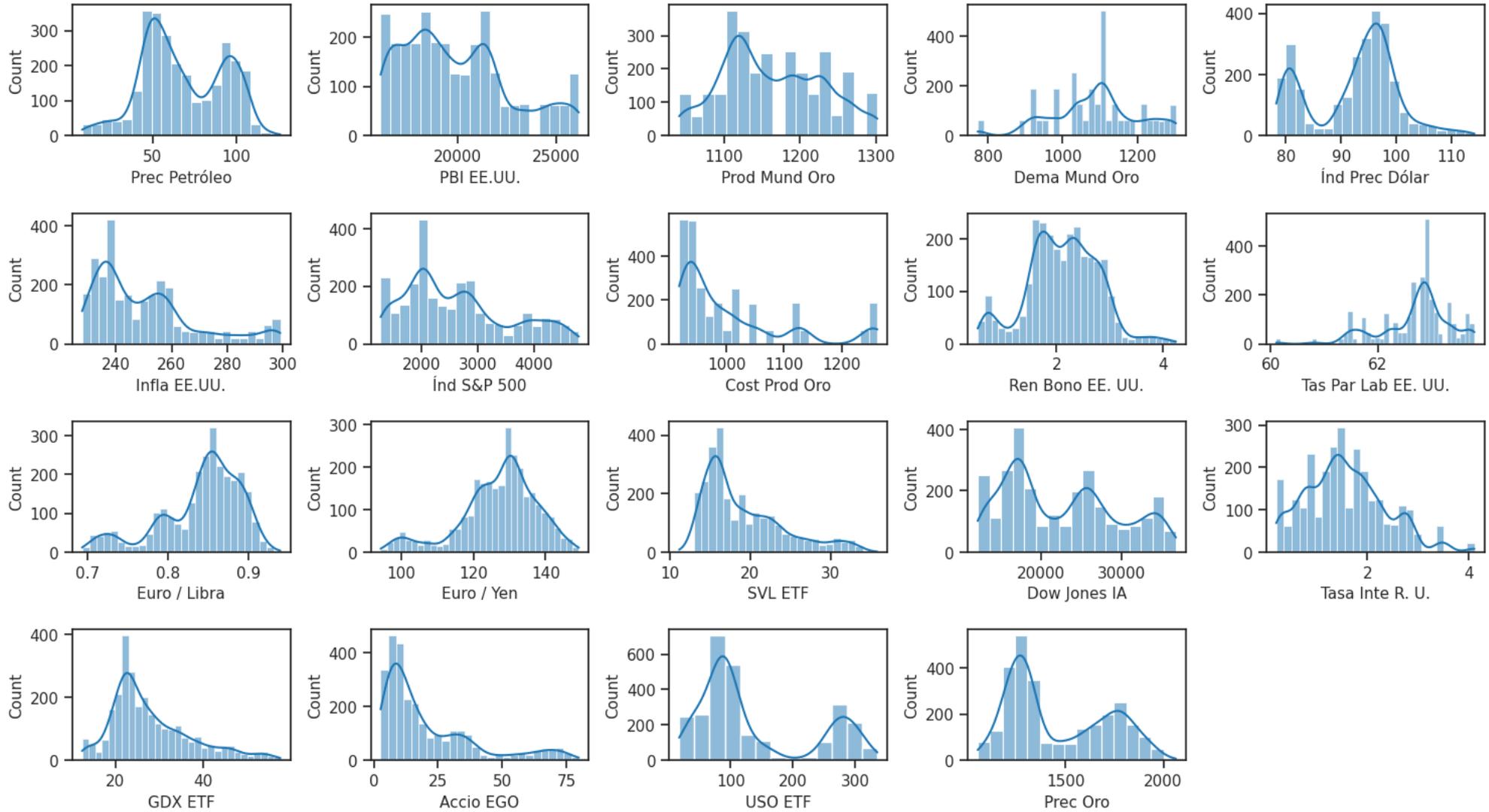


Figura 21. Histogramas de Variables de Estudio

Como siguiente paso se analizó la correlación de las variables de entrada con la variable objetivo (precio futuro del oro) a través de la matriz de Pearson como se ilustra en la Figura 22, donde se puede identificar que las 5 variables de entrada con mayor correlación con la variable objetivo son: VanEck Gold Miners ETF, Costo de Producción de Oro, iShares Silver Trust ETF, Inflación de Estados Unidos y el Índice S&P 500; lo cual indica que son los posibles factores principales que tienen un impacto significativo en la variación del precio del oro.

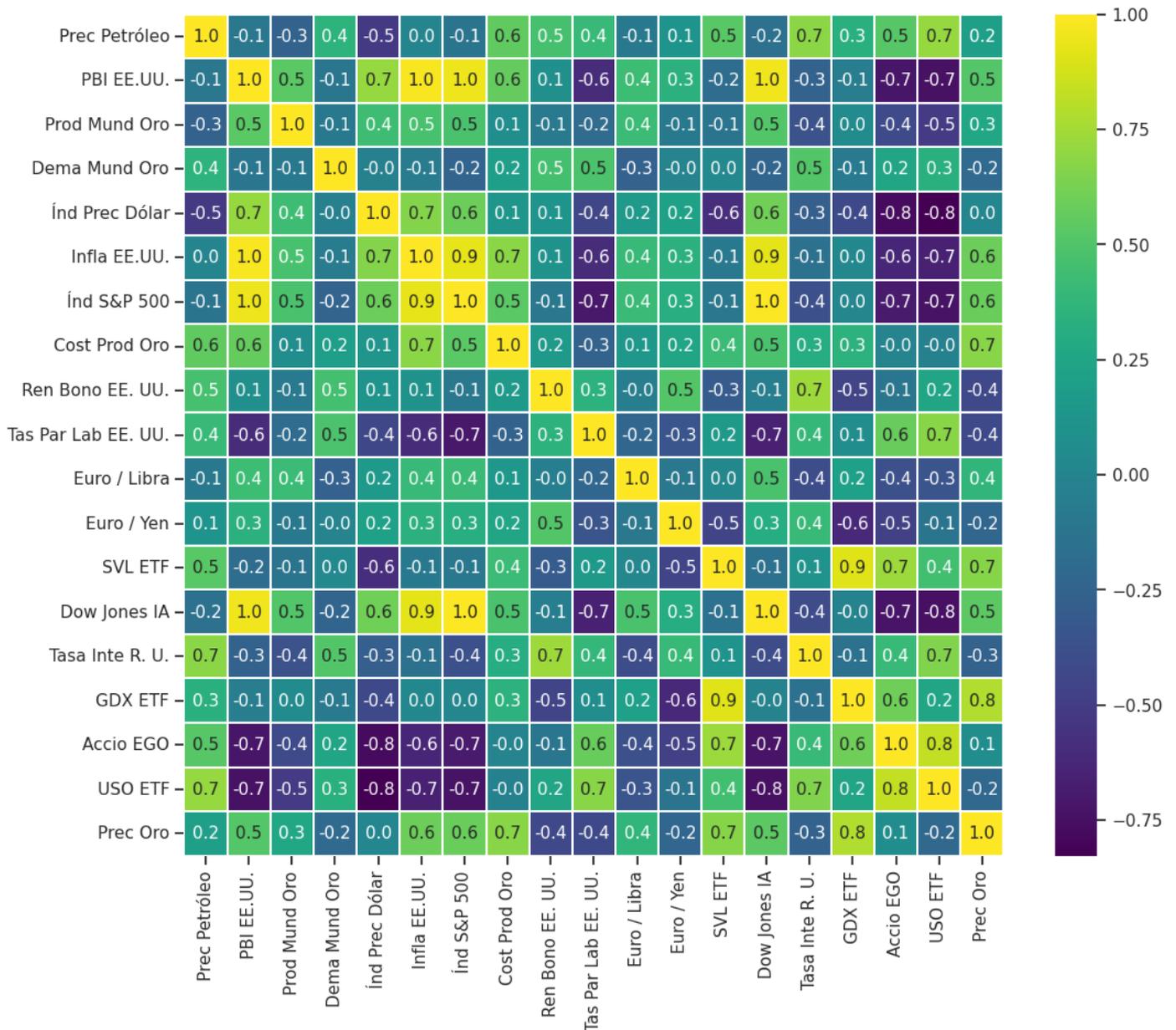


Figura 22. Correlación de Variables de Estudio

Por otro lado, es importante mencionar que las variables que tienen una correlación baja con la variable objetivo pueden no proporcionar una contribución significativa al modelo predictivo ya que tienen poca o ninguna relación lineal entre ellas. Sin embargo, se debe tener en cuenta que la correlación no es la única medida que describe la relación entre variables, ya que puede haber relaciones no lineales o patrones más complejos que no se reflejan en una correlación lineal. Además, Fosca (2020) indica que aunque no se observe una correlación significativa entre las variables de entrada y la variable objetivo, es importante considerar que al analizar cada variable por separado, se puede pasar por alto posibles combinaciones entre ellas, que podrían influir notablemente en el rendimiento del modelo.

### **3.5.4. Preprocesamiento de Datos**

Para llevar a cabo esta fase, se hizo lo siguiente:

#### **3.5.4.1. Limpieza de Datos**

Como se observó en la etapa de exploración de datos, no hay valores ausentes en el conjunto de datos, sin embargo, durante la visualización de relaciones y patrones se identificaron, mediante histogramas, algunos valores atípicos en algunas variables. Con el fin de obtener una representación más clara de estas anomalías, se crearán diagramas de caja y bigotes para cada variable.

En la Figura 23 podemos observar, a través de los diagramas de caja y bigotes, algunos valores atípicos presentes en la Demanda Mundial de Oro, Inflación de Estados Unidos, Costo de Producción de Oro, Rentabilidad del Bono de Estados Unidos, Tasa de Participación Laboral de Estados Unidos, Tipo de Cambio Euro/Libra, Tipo de Cambio Euro/Yen, iShares Silver Trust ETF, Tasa de Interés de Reino Unido, VanEck Gold Miners ETF y Precio de las Acciones de Eldorado Gold Corporation. Antes de tomar alguna decisión de qué hacer con estos valores atípicos se hizo un análisis minucioso para entender la causa; para esto se hizo lo siguiente:

En primera instancia se identificó en qué rango de fechas están los valores atípicos, con el fin de investigar posibles causas relacionadas a crisis económicas, políticas o pandemias. Como era de esperarse, la mayoría de estos valores atípicos tenían una justificación; a continuación, se detallan algunas de ellas:

- ✓ Los datos atípicos de la Inflación de Estados Unidos, el Costo de Producción de Oro, la Rentabilidad del Bono de Estados Unidos y la Tasa de Interés del Reino Unido se observaron entre 2020 y 2022, que es donde mostraron aumentos significativos debido a la reactivación económica posterior a la pandemia de COVID-19, las políticas monetarias expansivas y la escasez de suministros.
- ✓ En el caso de la Demanda Mundial de Oro los datos atípicos se centraron en el año 2020 donde hubo una baja demanda, lo cual se justifica debido a que en ese año la crisis ocasionada por el COVID-19 afectó de manera significativa a la economía mundial, reduciendo la compra de joyería y provocando ventas masivas de activos.
- ✓ Por otro lado, en el iShares Silver Trust ETF, el VanEck Gold Miners ETF y el precio de las acciones de Eldorado Gold Corporation, los datos atípicos están comprendidos entre 2012 y 2013, donde experimentaron un aumento en sus valores debido a la creciente demanda de metales preciosos como refugio seguro ante la incertidumbre económica global, que se debió a factores como: la crisis financiera en Europa, la desaceleración del crecimiento económico global y las tensiones políticas en diversas partes del mundo.
- ✓ Finalmente, los valores atípicos del tipo de cambio Euro/Yen y Euro/Libra, están comprendidos entre 2012 y 2015, donde experimentaron una baja debido a la crisis de deuda en la Eurozona producto de la crisis económica mundial en 2008, que debilitó la demanda del euro, junto con políticas monetarias expansivas que afectaron su atractivo como inversión.

Además, la incorporación de cierto nivel de variabilidad o ruido en los datos es beneficioso para mejorar el entrenamiento de los algoritmos, ya que ayuda a mitigar el sobreajuste del modelo. Esta estrategia permite que el modelo se adapte de manera más efectiva a conjuntos de datos desconocidos y no vistos previamente, lo que a su vez promueve un rendimiento más sólido y generalizable del modelo (Guozhong et al., citado por Castillo, 2022).

Por otro lado, al incorporar datos atípicos vinculados a eventos como crisis económicas, pandemias y otros eventos relacionados; estamos teniendo en cuenta esos factores en el modelo, lo cual ayuda a reflejar mejor la realidad y a anticipar cómo esos eventos pueden influir en los precios del oro.

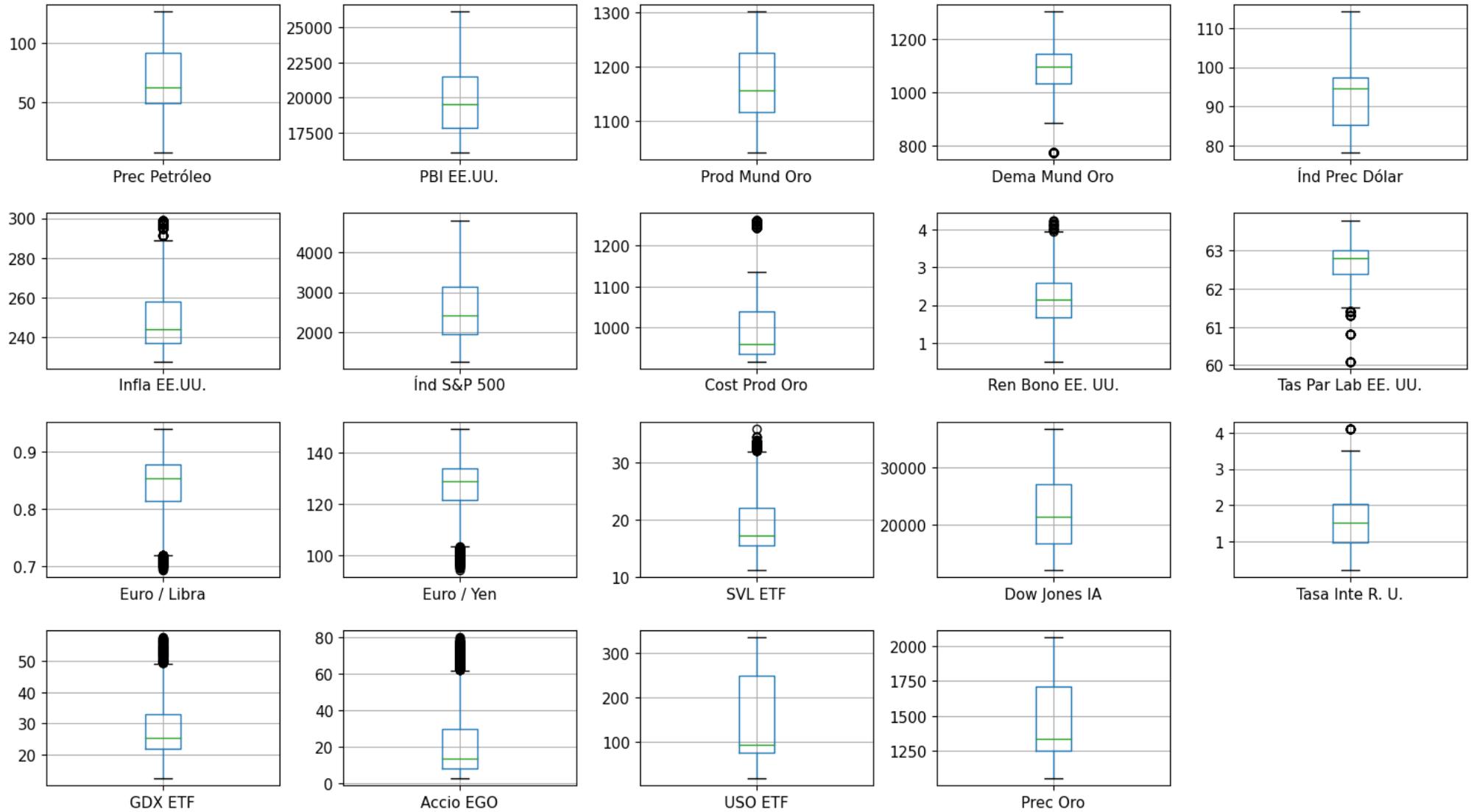


Figura 23. Diagramas de Caja y Bigotes de las Variables de Estudio

### 3.5.4.2. Estandarización de Datos

Como se observó en la fase de resumen estadístico de variables, los datos tienen una escala diferente, lo cual podría afectar en el desempeño del modelo predictivo, es por esto que se van a estandarizar los datos de entrenamiento y prueba, con la ayuda de la función `MinMaxScaler` del módulo `preprocessing` de la librería `scikit-learn` (`sklearn`), la cual escalará los datos en un rango de 0 y 1, como se muestra en la Figura 24. Antes de estandarizar los datos se definió en Google Colaboratory, la variable “y” que será la variable objetivo y la variable “x” que representará a todas las variables de entrada.

```

▶ from sklearn.preprocessing import MinMaxScaler
  scaler_x = MinMaxScaler()
  scaler_y = MinMaxScaler()

  X = scaler_x.fit_transform(X)
  y = scaler_y.fit_transform(y)

```

X														y				
Prec Petróleo	PBI EE.UU.	Prod Mund Oro	Dema Mund Oro	Índ Prec Dólar	Infla EE.UU.	Índ S&P 500	Cost Prod Oro	Ren Bono EE. UU.	Tas Par Lab EE. UU.	Euro / Libra	Euro / Yen	SVL ETF	Dow Jones IA	Tasa Inte R. U.	GDX ETF	Accio EGO	USO ETF	Prec Oro
0.80	0.0	0.17	0.78	0.04	0.0	0.0	0.06	0.39	0.97	0.57	0.10	0.72	0.01	0.49	0.92	0.92	0.94	0.55
0.80	0.0	0.17	0.78	0.05	0.0	0.0	0.06	0.39	0.97	0.55	0.09	0.70	0.01	0.49	0.92	0.91	0.94	0.55
0.79	0.0	0.17	0.78	0.07	0.0	0.0	0.06	0.40	0.97	0.54	0.08	0.70	0.01	0.49	0.92	0.91	0.93	0.56
0.79	0.0	0.17	0.78	0.08	0.0	0.0	0.06	0.39	0.97	0.53	0.07	0.68	0.01	0.49	0.91	0.90	0.93	0.56

Figura 24. Estandarización de Datos

### 3.5.4.3. Agregación de Retraso Temporal (lag)

Basándonos en las características de las variables y la naturaleza del problema en cuestión, nos encontramos ante un desafío de pronóstico de series de tiempo, es por esto que en el presente estudio se agregará un lag a todas las variables (de entrada y salida), con la finalidad de capturar patrones y tendencias fundamentales en los datos para poder realizar pronósticos precisos.

Para agregar el retraso temporal se desarrolló un código Python en Google Colaboratory, que genera lags para todos los datos y a la vez los combina con la base de datos original para crear una nueva base de datos donde se encuentran, tanto las variables originales como los lags correspondientes sumando un total de 37 variables de entrada. Es preciso mencionar que las variables con lag se diferencian por la notación “\_lag1” al final del nombre de cada variable como se muestra en la Figura 25.

Prec Petróleo	PBI EE.UU.	Prod Mund Oro	Dema Mund Oro	Índ Prec Dólar	Infla EE.UU.	Índ S&P 500	Cost Prod Oro	Ren Bono EE. UU.	Tas Par Lab EE. UU.	...	Tas Par Lab EE. UU._lag1	Euro / Libra_lag1	Euro / Yen_lag1	SVL ETF_lag1	Dow Jones IA_lag1	Tasa Inte R. U._lag1	GDX ETF_lag1	Accio EGO_lag1	USO ETF_lag1
0.80	0.0	0.17	0.78	0.05	0.0	0.0	0.06	0.39	0.97	...	0.97	0.57	0.10	0.72	0.01	0.49	0.92	0.92	0.94
0.79	0.0	0.17	0.78	0.07	0.0	0.0	0.06	0.40	0.97	...	0.97	0.55	0.09	0.70	0.01	0.49	0.92	0.91	0.94
0.79	0.0	0.17	0.78	0.08	0.0	0.0	0.06	0.39	0.97	...	0.97	0.54	0.08	0.70	0.01	0.49	0.92	0.91	0.93
0.79	0.0	0.17	0.78	0.08	0.0	0.0	0.06	0.39	0.97	...	0.97	0.53	0.07	0.68	0.01	0.49	0.91	0.90	0.93
0.80	0.0	0.17	0.78	0.07	0.0	0.0	0.06	0.39	0.97	...	0.97	0.54	0.07	0.69	0.01	0.49	0.92	0.90	0.93

Figura 25. Base de Datos con lags

### 3.5.4.4. División de Datos

Con el fin de evaluar y validar el desempeño del modelo predictivo, se dividió el conjunto de datos en conjunto de entrenamiento y conjunto de prueba, dejando de lado el conjunto de validación ya que más adelante se va a utilizar una técnica basada en validación cruzada, para determinar los hiperparámetros de cada algoritmo. Como estamos frente a un problema de pronóstico de series de tiempo es muy importante asegurarse de mantener el orden secuencial de los datos. Es por esto que la división de los datos se hizo con la ayuda de la función `train_test_split` del módulo `model_selection` de la librería de `scikit-learn` (`sklearn`), donde se estableció el parámetro "shuffle" en `False` (ver Figura 26) con el fin de garantizar que los datos se mantengan en su orden original.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15, shuffle=False)
```

Figura 26. División de Datos

Como resultado de la división se obtuvo que el conjunto de entrenamiento estará representado por 2348 (85%) registros para cada variable, comprendidos entre el 01/01/2012 y el 07/05/2021 y el conjunto de prueba por 415 (15%) registros comprendidos entre el 08/05/2021 y el 30/12/2022 como se ilustra en la siguiente figura.

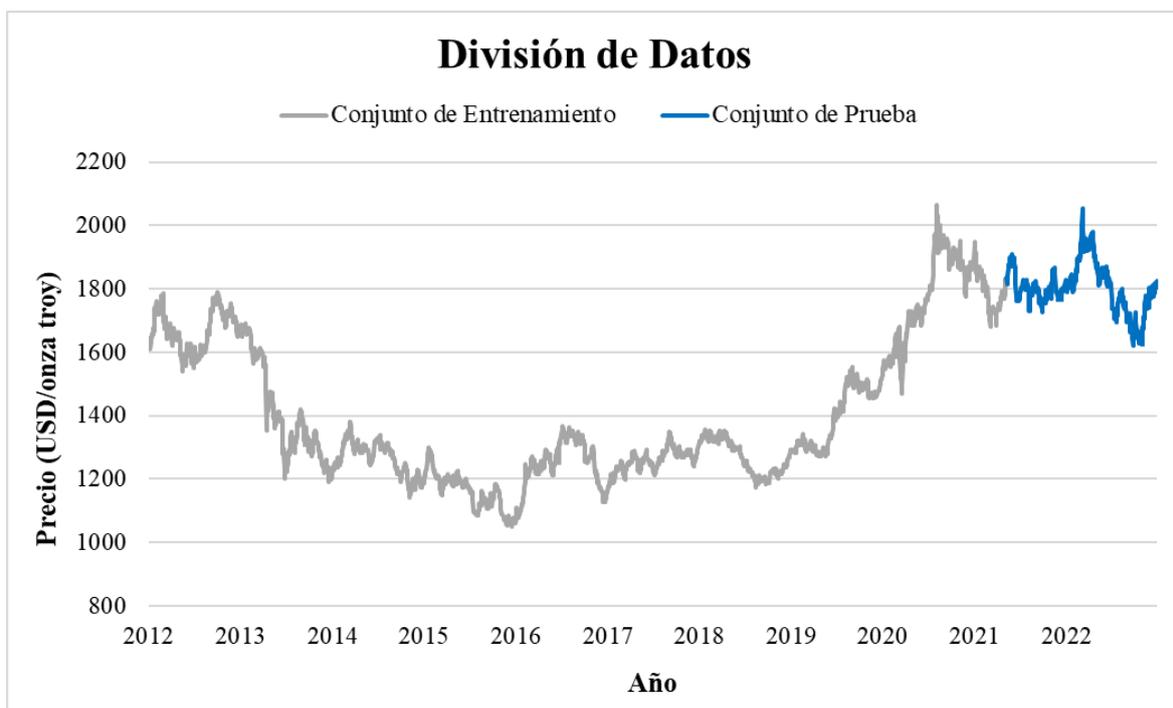


Figura 27. División de Datos para el Desarrollo del Modelo Predictivo

### **3.5.5. Elección de Algoritmos**

Dado que el método utilizado es el Aprendizaje Supervisado en un contexto de regresión, se emplearán algoritmos específicos para este tipo de problema. Para identificar el mejor modelo predictivo, se realizará una comparación entre los siguientes algoritmos: Regresión con Máquinas de Vectores de Soporte, Regresor de Bosques Aleatorios y Regresor de Gradiente Aumentado.

### **3.5.6. Construcción de Modelo Predictivo**

Para entender mejor la importancia de los hiperparámetros en un modelo predictivo, primero se determinará de manera aleatoria los valores de los hiperparámetros de cada algoritmo, luego se usará el método de GridSearchCV, una herramienta de validación cruzada disponible en la librería de scikit-learn, para ajustar los hiperparámetros y encontrar la mejor combinación de valores que optimice el rendimiento de los algoritmos.

Antes de desarrollar el modelo, es importante señalar que los algoritmos tienen varios hiperparámetros que necesitan ser ajustados para lograr un buen rendimiento, pero todos no tienen la misma relevancia, y tratar de ajustar cada uno de ellos puede consumir mucho tiempo y recursos computacionales, además, pueden ocasionar sobreajuste. Es por esto que solo se utilizarán los hiperparámetros que influyen más en el desempeño de cada algoritmo.

La construcción del modelo predictivo se hizo en base a la comparación de los tres algoritmos antes definidos. A continuación, se describe del desarrollo de cada uno de ellos:

#### **3.5.6.1. Regresión con Máquinas de Vectores de Soporte (SVR)**

El SVR tiene un hiperparámetro principal llamado kernel, el cual cuenta con varias funciones para poder abordar el problema definido; cada función tiene sus propias características y propiedades. Evaluar de manera individual cada tipo de kernel es una práctica importante, especialmente cuando se trabaja en un modelo predictivo, ya que la elección del kernel puede influir considerablemente en los resultados. Es por esto que se hará un análisis individual de los siguientes kernel: lineal (linear), polinómico (poly) y radial (rbf).

Los principales hiperparámetros del algoritmo SVR son los siguientes:

- ✓ **degree:** Ajusta la complejidad de la transformación polinómica.
- ✓ **gamma:** Controla la flexibilidad y la capacidad de ajuste del modelo.
- ✓ **C:** Controla el término de regularización y la tolerancia de error en el modelo.
- ✓ **epsilon:** Define el tamaño de la banda de margen en el modelo.

Es importante tener en cuenta que la utilización de los hiperparámetros antes descritos depende del tipo de kernel que se va a emplear. En la Tabla 3 se clasifican los hiperparámetros según el tipo de kernel.

Tabla 3. Clasificación de Principales Hiperparámetros según el tipo de Kernel

Tipo de kernel	Hiperparámetros
Lineal (linear)	- C - epsilon
Polinómico (poly)	- degree - gamma - C - epsilon
Radial (rbf)	- gamma - C - epsilon

#### a. Determinación de Hiperparámetros

Como primer análisis se determinarán los valores de los principales hiperparámetros de manera aleatoria (ver Tabla 4), según los tipos de kernel antes mencionados:

Tabla 4. Valores Aleatorios de los Principales Hiperparámetros del Algoritmo SVR

Tipo de Kernel	Hiperparámetros
Lineal (linear)	- C = 10 - epsilon = 0.1
Polinómico (poly)	- degree = 2 - gamma = 0.1 - C = 10 - epsilon = 0.1
Radial (rbf)	- gamma = 0.01 - C = 1 - epsilon = 0.01

Como segundo paso se determinarán los hiperparámetros por medio la técnica de búsqueda en cuadrícula con validación cruzada, para esto primero definiremos un conjunto de valores para cada hiperparámetro según el tipo de kernel, como se detalla en la tabla siguiente:

Tabla 5. Rango de Valores para los Principales Hiperparámetros del Algoritmo SVR

Tipo de Kernel	Hiperparámetros
Lineal (linear)	- C = [1, 10, 100, 1000] - epsilon = [0.1, 0.01]
Polinómico (poly)	- degree = [2, 3] - gamma = [1, 0.1, 0.01] - C = [1, 10, 100, 1000] - epsilon = [0.1, 0.01]
Radial (rbf)	- gamma = [1, 0.1, 0.01] - C = [1, 10, 100, 1000] - epsilon = [0.1, 0.01]

Luego se definió que el número de pliegues (CV) será 5, es decir que los datos de entrenamiento se dividirán en 5 conjuntos para su posterior evaluación; seguidamente nos ayudaremos de la función GridSearchCV del módulo model\_selection de la librería de scikit-learn para poder encontrar mediante la aplicación del atributo .best\_params\_ el conjunto de hiperparámetros óptimos para cada kernel (ver Tabla 6).

Tabla 6. Valores Óptimos de los Hiperparámetros del Algoritmo SVR

Tipo de Kernel	Código Python	Valores Óptimos de Hiperparámetros
Lineal (linear)	<pre> ▶ from sklearn.model_selection import GridSearchCV  param_grid = {'kernel':['linear'],               'C':[1, 10, 100, 1000],               'epsilon':[0.1, 0.01]}  svr_grid = GridSearchCV(SVR(), param_grid, cv=5) svr_grid.fit(X_train, y_train)                     </pre>	<pre> ▶ svr_grid.best_params_  ↳ C: 1000    epsilon: 0.01    kernel: linear                     </pre>
Polinómico (poly)	<pre> ▶ from sklearn.model_selection import GridSearchCV  param_grid = {'kernel':['poly'],               'degree':[2, 3],               'gamma':[1, 0.1, 0.01],               'C':[1, 10, 100, 1000],               'epsilon':[0.1, 0.01]}  svr_grid = GridSearchCV(SVR(), param_grid, cv=5) svr_grid.fit(X_train, y_train)                     </pre>	<pre> ▶ svr_grid.best_params_  ↳ C: 100    degree: 2    epsilon: 0.01    gamma: 0.01    kernel: poly                     </pre>
Radial (rbf)	<pre> ▶ from sklearn.model_selection import GridSearchCV  param_grid = {'kernel':['rbf'],               'gamma':[1, 0.1, 0.01],               'C':[1, 10, 100, 1000],               'epsilon':[0.1, 0.01]}  svr_grid = GridSearchCV(SVR(), param_grid, cv=5) svr_grid.fit(X_train, y_train)                     </pre>	<pre> ▶ svr_grid.best_params_  ↳ C: 100    epsilon: 0.01    gamma: 0.01    kernel: rbf                     </pre>

## b. Entrenamiento de Algoritmo

El entrenamiento de los algoritmos se hizo con el conjunto de entrenamiento, antes definido, que corresponde al 85% del total de datos, para esto nos ayudamos de la función SVR del módulo svm de la librería de scikit-learn, a continuación, se muestra el entrenamiento en base a cada kernel:

Tabla 7. Entrenamiento del Algoritmo SVR

Tipo de Kernel	Entrenamiento con Hiperparámetros Determinados Aleatoriamente	Entrenamiento con Hiperparámetros Determinados con GridSearchCV
Lineal (linear)	<pre> ▶ from sklearn.svm import SVR  svr = SVR(kernel='linear',            C=10,            epsilon=0.1)  svr.fit(X_train, y_train) </pre>	<pre> ▶ from sklearn.svm import SVR  svr = SVR(kernel='linear',            C=1000,            epsilon=0.01)  svr.fit(X_train, y_train) </pre>
Polinómico (poly)	<pre> ▶ from sklearn.svm import SVR  svr = SVR(kernel='poly',            degree=2,            gamma=0.1,            C=10,            epsilon=0.1)  svr.fit(X_train, y_train) </pre>	<pre> ▶ from sklearn.svm import SVR  svr = SVR(kernel='poly',            degree=2,            gamma=0.01,            C=100,            epsilon=0.01)  svr.fit(X_train, y_train) </pre>
Radial (rbf)	<pre> ▶ from sklearn.svm import SVR  svr = SVR(kernel='rbf',            gamma=0.01,            C=1,            epsilon=0.01)  svr.fit(X_train, y_train) </pre>	<pre> ▶ from sklearn.svm import SVR  svr = SVR(kernel='rbf',            gamma=0.01,            C=100,            epsilon=0.01)  svr.fit(X_train, y_train) </pre>

### c. Evaluación de Algoritmo

Para la evaluación del algoritmo SVR primero se hizo una comparación de las métricas de evaluación (MAE, MSE, RMSE y R2) en base a los hiperparámetros determinados tanto aleatoriamente como por la técnica de búsqueda en cuadrícula con validación cruzada (GridSearchCV). Adicionalmente el análisis se hará utilizando el conjunto de entrenamiento y prueba con la finalidad de ver si existe sobreajuste (overfitting) o subajuste (underfitting).

Como primer paso se utilizó el algoritmo entrenado para predecir el precio del oro en base al conjunto de entrenamiento y luego al conjunto de prueba, para luego hacer una comparación de las métricas de evaluación en función al valor real y el predicho. En la Figura 28 se detalla el código general de Python que se utilizó para hacer este análisis para cada kernel:

```

▶ #Predicción con valores de entrenamiento
y_train_pred = svr.predict(X_train)

#Predicción con valores de prueba
y_test_pred = svr.predict(X_test)

#Cálculo de métricas de evaluación
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error

print("MSE Test: ", mean_squared_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("MSE Train: ", mean_squared_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("RMSE Test: ", np.sqrt(mean_squared_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("RMSE Train: ", np.sqrt(mean_squared_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("MAE Test: ", mean_absolute_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("MAE Train: ", mean_absolute_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("R2 Test: ", r2_score scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("R2 Train: ", r2_score scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))

```

Figura 28. Cálculo de Métricas de Evaluación para el Algoritmo SVR

En la Tabla 8 se observa el valor de cada métrica de evaluación correspondiente a cada subconjunto de datos (entrenamiento y prueba) según el tipo de kernel y la técnica de determinación de hiperparámetros. A través de esto, podemos deducir que el modelo tiene mucho más error de predicción con los hiperparámetros que se determinaron de forma aleatoria que con los que se determinaron con la técnica de búsqueda en cuadrícula con validación cruzada. Por otro lado, también se aprecia que existe sobreajuste, sobre todo en los hiperparámetros que se determinaron de manera aleatoria, pero en el caso de los hiperparámetros determinados con búsqueda en cuadrícula con validación cruzada, se ve que este sobreajuste disminuyó notablemente, ya que el valor de las métricas de evaluación en el entrenamiento y la prueba no tienen mucha diferencia a comparación de la técnica aleatoria. En base a esto, podemos decir que la selección adecuada de los hiperparámetros puede influir considerablemente en la exactitud y el desempeño global del modelo.

Tabla 8. Valor de Métricas de Evaluación para el Algoritmo SVR

Tipo de Kernel	Tipo de Conjunto de Datos	Hiperparámetros Determinados Aleatoriamente				Hiperparámetros Determinados con GridSearchCV			
		MAE	RMSE	MSE	R <sup>2</sup>	MAE	RMSE	MSE	R <sup>2</sup>
Lineal (linear)	Entramiento	36.64	44.09	1943.82	0.96	4.86	6.71	45.01	1.00
	Prueba	74.57	91.00	8281.72	0.0	10.45	13.36	178.42	0.97
Polinómico (poly)	Entramiento	40.86	49.04	2405.38	0.95	6.59	8.84	78.21	1.00
	Prueba	94.54	106.34	11308.99	0.0	42.78	55.54	3084.55	0.47
Radial (rbf)	Entramiento	9.66	12.99	168.72	1.0	4.88	6.51	42.44	1.00
	Prueba	47.01	56.70	3215.27	0.44	9.03	11.44	130.95	0.98

De acuerdo a las métricas de evaluación, el rendimiento del algoritmo con los hiperparámetros determinados con la técnica de búsqueda en cuadrícula con validación cruzada es mejor, teniendo mejores resultados con un kernel radial, sin embargo aún se puede observar un ligero sobreajuste, pero es posible que se pueda mejorar haciendo una selección de variables relevantes, con la técnica de Eliminación Recursiva de Características (RFE), para esto nos apoyaremos de la función RFE del módulo `feature_selection` de la librería de `scikit-learn`, como estimador podemos tomar varios algoritmos, pero en este caso se tomará el `LinearRegression()` para el SVR con un kernel radial, debido a que el RFE no es compatible con este tipo de kernel y también porque se obtuvo mejores resultados a comparación de otros estimadores. Adicionalmente, nos ayudamos de la función `GridSearchCV` con un número de pliegues igual a cinco ( $CV=5$ ), con el objetivo de conseguir el número más adecuado de variables que se encuentra entre un rango de 1 a 37, ya que se tiene un total de 37 variables, sumando los lags que se le aplicó a cada variable. En la tabla siguiente se muestra el código utilizado y el resultado obtenido, según el tipo de kernel:

Tabla 9. Determinación del Número de Variables Óptimas para el Algoritmo SVR

Tipo de Kernel	Código Python	Número de Variables Óptimas
Radial (poly)	<pre data-bbox="439 379 1429 754"> ▶ from sklearn.linear_model import LinearRegression from sklearn.feature_selection import RFE from sklearn.model_selection import GridSearchCV  param_grid = {'n_features_to_select':list(range(1, 37))}  estimator = LinearRegression()  rfe_cv = GridSearchCV(RFE(estimator, step=1), param_grid, cv=5) rfe_cv.fit(X_train, y_train.ravel()) </pre>	<pre data-bbox="1518 491 2007 616"> ▶ rfe_cv.best_params_ ↳ {'n_features_to_select': 13} </pre>

En la Tabla 10 se detallan las trece variables óptimas seleccionadas anteriormente, con el método de RFE. Cabe mencionar que para obtener las variables nos apoyamos del atributo `.support_` de la función RFE.

Tabla 10. Subconjunto Óptimo de Variables para el Algoritmo SVR

Tipo de Kernel	Subconjunto Óptimo de Variables
Radial (rbf)	<ul style="list-style-type: none"> <li>- Índ Prec Dólar</li> <li>- Infla EE.UU.</li> <li>- Ren Bono EE. UU.</li> <li>- SVL ETF</li> <li>- Dow Jones IA</li> <li>- GDX ETF</li> <li>- Índ Prec Dólar_lag1</li> <li>- Infla EE.UU._lag1</li> <li>- Ren Bono EE. UU._lag1</li> <li>- SVL ETF_lag1</li> <li>- Dow Jones IA_lag1</li> <li>- GDX ETF_lag1</li> <li>- Prec Oro_lag1</li> </ul>

Posteriormente se vuelve a entrenar y evaluar el algoritmo de SVR con las nuevas variables obtenidas con el método de RFE y con los hiperparámetros obtenidos anteriormente con la técnica de búsqueda en cuadrícula con validación cruzada, ya que es con lo que se obtuvo los mejores resultados. A continuación, se presentan los resultados obtenidos en este nuevo análisis:

Tabla 11. Métricas de Evaluación usando Subconjunto Óptimo de Variables con SVR

Tipo de Kernel	Tipo de Conjunto de Datos	Métricas de Evaluación			
		MAE	RMSE	MSE	R <sup>2</sup>
Radial (rbf)	Entrenamiento	4.83	6.69	44.76	1.00
	Prueba	6.20	8.37	70.01	0.99

En la Tabla 11 podemos observar que el desempeño del modelo se optimizó, ya que se disminuyó el sobreajuste mostrando un error menor entre el conjunto de entrenamiento y prueba. Adicionalmente, es importante destacar que se disminuyó el número de variables considerablemente, pasando de 37 variables a tan solo 13, lo cual es relevante ya que se simplifica el análisis y construcción del modelo.

Con la finalidad de obtener un ranking de las variables en base a la importancia que les da el algoritmo SVR para poder predecir el precio del oro, nos ayudaremos de la función PermutationImportance del módulo eli5 de la librería de scikit-learn, como se detalla en la

Figura 29. Es preciso señalar que se utilizó esta función ya que el algoritmo SVR no cuenta con una función propia para poder analizar la importancia de cada variable para el algoritmo.

```

!pip install eli5
from sklearn.svm import SVR
import eli5
from eli5.sklearn import PermutationImportance

svr_model = SVR(kernel='rbf', gamma=0.01, C=100, epsilon=0.01)
svr_model.fit(X_train_reduced, y_train)

perm = PermutationImportance(svr_model, random_state=42)
perm.fit(X_train_reduced, y_train)

eli5.show_weights(perm, feature_names=X_train_reduced.columns.tolist())

```

Figura 29. Código Python para hallar la Importancia de Variables de Entrada en el SVR

En la Tabla 12 podemos observar el ranking de las variables en base a la importancia que le otorga el algoritmo.

Tabla 12. Ranking de Variables de Entrada en el Algoritmo SVR

Variable de Entrada		Ranking
Abreviatura	Nombre	
Prec Oro_lag1	Precio del oro del día anterior	1
SVL ETF	iShares Silver Trust ETF	2
SVL ETF_lag1	iShares Silver Trust ETF del día anterior	3
GDX ETF	VanEck Gold Miners ETF	4
GDX ETF_lag1	VanEck Gold Miners ETF del día anterior	5
Ren Bono EE. UU.	Rentabilidad del Bono de Estados Unidos	6
Índ Prec Dólar	Índice del Precio del Dólar	7
Ren Bono EE. UU._lag1	Rentabilidad del Bono de Estados Unidos del día anterior	8
Índ Prec Dólar_lag1	Índice del Precio del Dólar del día anterior	9
Dow Jones IA	Dow Jones Industrial Average	10
Dow Jones IA_lag1	Dow Jones Industrial Average del día anterior	11
Infla EE.UU._lag1	Inflación de Estados Unidos del día anterior	12
Infla EE.UU.	Inflación de Estados Unidos	13

### 3.5.6.2. Regresor de Bosques Aleatorios (RFR)

Para el análisis del algoritmo Regresor de Bosques Aleatorios, se utilizarán los siguientes hiperparámetros principales:

- ✓ **n\_estimators:** Cantidad de árboles de decisión que se utilizarán en el bosque.
- ✓ **max\_depth:** Límite máximo de profundidad que puede alcanzar cada árbol.
- ✓ **min\_samples\_split:** Cantidad mínima de registros necesarios para dividir un nodo interno.
- ✓ **min\_samples\_leaf:** Cantidad mínima de registros necesarios en una hoja.
- ✓ **max\_features:** Número máximo de variables consideradas en cada división dentro de los árboles.
- ✓ **bootstrap:** Indica si se emplea muestreo con reemplazo durante la construcción de cada árbol del bosque.

#### a. Determinación de Hiperparámetros

Para el Regresor de Bosques Aleatorios se determinaron de manera aleatoria los siguientes valores para los principales hiperparámetros antes mencionados:

Tabla 13. Valores Aleatorios de los Principales Hiperparámetros del Algoritmo RFR

Algoritmo	Hiperparámetros
Regresor de Bosques Aleatorios	<ul style="list-style-type: none"><li>- n_estimators = 100</li><li>- max_depth = 10</li><li>- min_samples_split = 2</li><li>- min_samples_leaf = 1</li><li>- max_features = None</li><li>- bootstrap = False</li></ul>

Luego, para determinar los hiperparámetros por medio de la técnica de búsqueda en cuadrícula con validación cruzada, se definieron los rangos de valores para cada hiperparámetro, como se detalla en la Tabla 14.

Tabla 14. Rango de Valores para los Principales Hiperparámetros del Algoritmo RFR

Algoritmo	Hiperparámetros
Regresor de Bosques Aleatorios	<ul style="list-style-type: none"> <li>- n_estimators = [50, 100, 200]</li> <li>- max_depth = [1, 5, 10]</li> <li>- min_samples_split = [2, 5, 10]</li> <li>- min_samples_leaf = [1, 2, 4]</li> <li>- max_features = [None, 'sqrt', 'log2']</li> <li>- bootstrap = [True, False]</li> </ul>

Posteriormente se definió que el número de pliegues (CV) será 5, seguidamente nos ayudaremos de la función GridSearchCV del módulo model\_selection de la librería de scikit-learn para poder encontrar mediante la aplicación del atributo svr\_grid.best\_params\_ el conjunto de hiperparámetros óptimos (ver Tabla 15).

Tabla 15. Valores Óptimos de los Hiperparámetros del Algoritmo RFR

Algoritmo	Código Python	Valores Óptimos de Hiperparámetros
Regresor de Bosques Aleatorios	<pre> from sklearn.model_selection import GridSearchCV  param_grid = {'n_estimators':[50, 100, 200],               'max_depth':[1, 5, 10],               'min_samples_split': [2, 5, 10],               'min_samples_leaf': [1, 2, 4],               'max_features':[None, 'sqrt', 'log2'],               'bootstrap':[True, False]}  rfr_grid = GridSearchCV(RandomForestRegressor(random_state=1), param_grid, cv=5) rfr_grid.fit(X_train, y_train) </pre>	<pre> rfr_grid.best_params_  bootstrap: True max_depth: 5 max_features: None min_samples_leaf: 4 min_samples_split: 10 n_estimators: 50 </pre>

## b. Entrenamiento de Algoritmo

Al igual que en el SVR, el entrenamiento del algoritmo se hizo con el conjunto de entrenamiento, antes definido, que corresponde al 85% del total de datos, para esto nos ayudamos de la función RandomForestRegressor del módulo ensemble de la librería de scikit-learn. En la Tabla 16 se muestra como se hizo el entrenamiento del algoritmo en base a la técnica de determinación de hiperparámetros.

Tabla 16. Entrenamiento del Algoritmo RFR

Algoritmo	Entrenamiento con Hiperparámetros Determinados Aleatoriamente	Entrenamiento con Hiperparámetros Determinados con GridSearchCV
Regresor de Bosques Aleatorios	<pre>from sklearn.ensemble import RandomForestRegressor  rfr = RandomForestRegressor(n_estimators=100,                            max_depth=10,                            min_samples_split=2,                            min_samples_leaf=1,                            max_features=None,                            bootstrap=False,                            random_state=1)  rfr.fit(X_train, y_train)</pre>	<pre>from sklearn.ensemble import RandomForestRegressor  rfr = RandomForestRegressor(n_estimators=50,                            max_depth=5,                            min_samples_split=10,                            min_samples_leaf=4,                            max_features=None,                            bootstrap=True,                            random_state=1)  rfr.fit(X_train, y_train)</pre>

### c. Evaluación de Algoritmo

Para la evaluación del algoritmo Regresor de Bosques Aleatorios, al igual que el SVR, se utilizó el algoritmo previamente entrenado para predecir el precio del oro en base al conjunto de datos de entrenamiento y prueba, para luego hacer una comparación de las métricas de evaluación en función al valor real y el predicho, y adicionalmente identificar problemas de sobreajuste o subajuste. En la Figura 30 se detalla el código general de Python que se utilizó para hacer este análisis.

```
#Predicción con valores de entrenamiento
y_train_pred = rfr.predict(X_train)

#Predicción con valores de prueba
y_test_pred = rfr.predict(X_test)

#Cálculo de métricas de evaluación
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error

print("MSE Test: ", mean_squared_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1)))
print("MSE Train: ", mean_squared_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1)))
print("RMSE Test: ", np.sqrt(mean_squared_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("RMSE Train: ", np.sqrt(mean_squared_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("MAE Test: ", mean_absolute_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1)))
print("MAE Train: ", mean_absolute_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1)))
print("R2 Test: ", r2_score scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1)))
print("R2 Train: ", r2_score scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1)))
```

Figura 30. Cálculo de Métricas de Evaluación para el Algoritmo RFR

En la Tabla 17 se observa el valor de cada métrica de evaluación asociada a cada conjunto de datos (entrenamiento y prueba) según la técnica de determinación de hiperparámetros. Como era de esperarse, se puede ver que el modelo tiene más error de predicción con los hiperparámetros que se determinaron aleatoriamente que con lo que se determinaron con la técnica de búsqueda en cuadrícula con validación cruzada. Adicionalmente, se observa la existencia de sobreajuste, especialmente en los hiperparámetros que se determinaron de manera aleatoria y en menor medida en los hiperparámetros determinados con búsqueda en cuadrícula con validación cruzada.

Tabla 17. Valor de Métricas de Evaluación para el Algoritmo RFR

Algoritmo	Tipo de Conjunto de Datos	Hiperparámetros Determinados Aleatoriamente				Hiperparámetros Determinados con GridSearchCV			
		MAE	RMSE	MSE	R <sup>2</sup>	MAE	RMSE	MSE	R <sup>2</sup>
Regresor de Bosques Aleatorios	Entrenamiento	4.21	6.19	38.38	1.00	8.97	12.62	159.25	1.00
	Prueba	34.53	43.18	1864.72	0.68	15.47	21.04	442.66	0.92

Al igual que el algoritmo SVR, se utilizará el método de RFE con la finalidad de hacer una selección de variables relevantes y disminuir un poco el sobreajuste que aún existe. Para esto también nos ayudaremos de la función RFE del módulo `feature_selection` de la librería de `scikit-learn` y como estimador se tomará el mismo algoritmo Regresor de Bosques Aleatorios, ya que a diferencia del SVR con kernel radial, este si es compatible con el RFE. Adicionalmente, nos apoyaremos de la función `GridSearchCV` con un número de pliegues igual a cinco ( $CV=5$ ), para obtener el número óptimo de variables. En la tabla siguiente se detalla el código empleado y el resultado obtenido.

Tabla 18. Determinación del Número de Variables Óptimas para el Algoritmo RFR

Algoritmo	Código Python	Número de Variables Óptimas
<p>Regresor de Bosques Aleatorios</p>	<pre data-bbox="488 352 1464 941"> ▶ from sklearn.ensemble import RandomForestRegressor from sklearn.feature_selection import RFE from sklearn.model_selection import GridSearchCV  param_grid = {'n_features_to_select':list(range(1, 37))}  estimator = RandomForestRegressor(n_estimators=50,                                   max_depth=5,                                   min_samples_split=10,                                   min_samples_leaf=4,                                   max_features=None,                                   bootstrap=True,                                   random_state=1)  rfe_cv = GridSearchCV(RFE(estimator, step=1), param_grid, cv=5) rfe_cv.fit(X_train, y_train.ravel())                     </pre>	<pre data-bbox="1563 576 1998 687"> ▶ rfe_cv.best_params_ ↳ {'n_features_to_select': 5}                     </pre>

Posteriormente, en la Tabla 19 se presentan las cinco variables óptimas seleccionadas anteriormente, con el método de RFE. Al igual que en el algoritmo de SVR, nos apoyaremos en el atributo `.support_` de la función RFE, para obtener dichas variables.

Tabla 19. Subconjunto Óptimo de Variables para el Algoritmo RFR

Algoritmo	Subconjunto Óptimo de Variables
Regresor de Bosques Aleatorios	<ul style="list-style-type: none"> <li>- SVL ETF</li> <li>- GDX ETF</li> <li>- Prod Mund Oro</li> <li>- Ren Bono EE. UU._lag1</li> <li>- Prec Oro_lag1</li> </ul>

En la Tabla 20, podemos observar que el desempeño del modelo mejoró, ya que se disminuyó un poco el sobreajuste mostrando un error menor entre el conjunto de entrenamiento y prueba. Muy aparte de que se disminuyó el sobreajuste, también se redujo significativamente el número de variables de entrada, ya que de las 37 que teníamos al principio ahora se tiene sólo 5.

Tabla 20. Métricas de Evaluación usando Subconjunto Óptimo de Variables con RFR

Algoritmo	Tipo de Conjunto de Datos	Métricas de Evaluación			
		MAE	RMSE	MSE	R <sup>2</sup>
Regresor de Bosques Aleatorios	Entrenamiento	9.40	13.36	178.64	1.00
	Prueba	12.89	17.01	289.39	0.95

Por otro lado, el algoritmo Regresor de Bosques Aleatorios cuenta con el atributo `feature_importances_`, el cual es muy útil para analizar la importancia de las variables de entrada para el algoritmo. En la Tabla 21 se presenta un ranking de variables de entrada en base al grado de importancia que le otorga el algoritmo a cada variable.

Tabla 21. Ranking de Variables de Entrada en el Algoritmo RFR

Variable de Entrada		Ranking
Abreviatura	Nombre	
Prec Oro_lag1	Precio del oro del día anterior	1
GDX ETF	VanEck Gold Miners ETF	2
Prod Mund Oro	Producción Mundial de Oro	3
Ren Bono EE. UU._lag1	Rentabilidad del Bono de Estados Unidos del día anterior	4
SVL ETF	iShares Silver Trust ETF	5

### 3.5.6.3. Regresor de Gradiente Aumentado (GBR)

Para el análisis del algoritmo Regresor de Gradiente Aumentado, se utilizarán los siguientes hiperparámetros principales:

- ✓ **learning\_rate:** Tasa de aprendizaje para controlar la contribución de cada árbol.
- ✓ **n\_estimators:** Número de árboles en la secuencia de boosting.
- ✓ **subsample:** Proporción de datos empleada para entrenar cada árbol.
- ✓ **max\_depth:** Límite máximo de profundidad que puede alcanzar cada árbol.

#### a. Determinación de Hiperparámetros

Para el Regresor de Gradiente Aumentado, al igual que los anteriores algoritmos, se determinaron de manera aleatoria los siguientes valores para los principales hiperparámetros antes mencionados:

Tabla 22. Valores Aleatorios de los Principales Hiperparámetros del Algoritmo GBR

Algoritmo	Hiperparámetros
Regresor de Gradiente Aumentado	- learning_rate = 0.1
	- n_estimators = 50
	- subsample = 0.8
	- max_depth = 1

Luego, para determinar los hiperparámetros por medio de la técnica de búsqueda en cuadrícula con validación cruzada, se definieron los rangos de valores para cada hiperparámetro, como se detalla en la Tabla 23.

Tabla 23. Rango de Valores para los Principales Hiperparámetros del Algoritmo GBR

Algoritmo	Hiperparámetros
Regresor de Gradiente Aumentado	- learning_rate = [0.1, 0.01, 0.001]
	- n_estimators = [20, 50, 80]
	- subsample = [0.8, 0.9, 1.0]
	- max_depth = [1, 3, 5]

Posteriormente se definió que el número de pliegues (CV) será 5, seguidamente nos ayudaremos de la función GridSearchCV del módulo model\_selection de la librería de scikit-learn para poder encontrar mediante la aplicación del atributo svr\_grid.best\_params\_ el conjunto de hiperparámetros óptimos (ver Tabla 24).

Tabla 24. Valores Óptimos de los Hiperparámetros del Algoritmo GBR

Algoritmo	Código Python	Valores Óptimos de Hiperparámetros
Regresor de Gradiente Aumentado	<pre> from sklearn.model_selection import GridSearchCV  param_grid = {'learning_rate': [0.1, 0.01, 0.001],               'n_estimators': [20, 50, 80],               'subsample': [0.8, 0.9, 1.0],               'max_depth': [1, 3, 5]}  gbr_grid = GridSearchCV(GradientBoostingRegressor(random_state=1), param_grid, cv=5) gbr_grid.fit(X_train, y_train) </pre>	<pre> gbr_grid.best_params_  learning_rate: 0.1 max_depth: 3 n_estimators: 80 subsample: 0.8 </pre>

## b. Entrenamiento de Algoritmo

Al igual que en los anteriores algoritmos, el entrenamiento del algoritmo se hizo con el conjunto de entrenamiento, antes definido, que corresponde al 85% del total de datos, para esto nos ayudamos de la función GradientBoostingRegressor del módulo ensemble de la librería de scikit-learn. En la Tabla 25 se muestra como se hizo el entrenamiento del algoritmo en base a la técnica de determinación de hiperparámetros.

Tabla 25. Entrenamiento del Algoritmo GBR

Algoritmo	Entrenamiento con hiperparámetros determinados aleatoriamente	Entrenamiento con hiperparámetros determinados con GridSearchCV
Regresor de Gradiente Aumentado	<pre data-bbox="443 391 1220 694"> ▶ from sklearn.ensemble import GradientBoostingRegressor  gbr = GradientBoostingRegressor(learning_rate=0.1,                                 n_estimators=50,                                 subsample=0.8,                                 max_depth=1,                                 random_state=1)  gbr.fit(X_train, y_train.ravel()) </pre>	<pre data-bbox="1249 391 2042 694"> ▶ from sklearn.ensemble import GradientBoostingRegressor  gbr = GradientBoostingRegressor(learning_rate=0.1,                                 n_estimators=80,                                 subsample=0.8,                                 max_depth=3,                                 random_state=1)  gbr.fit(X_train, y_train.ravel()) </pre>

### c. Evaluación de Algoritmo

Para la evaluación del algoritmo Regresor de Gradiente Aumentado, al igual que los anteriores, se utilizó el algoritmo previamente entrenado para predecir el precio del oro en base al conjunto de datos de entrenamiento y prueba, para luego hacer una comparación de las métricas de evaluación en función al valor real y el predicho. En la Figura 31 se detalla el código general de Python que se utilizó para hacer este análisis.

```

▶ #Predicción con valores de entrenamiento
y_train_pred = gbr.predict(X_train)

#Predicción con valores de prueba
y_test_pred = gbr.predict(X_test)

#Cálculo de métricas de evaluación
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error

print("MSE Test: ", mean_squared_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1)))
print("MSE Train: ", mean_squared_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1)))
print("RMSE Test: ", np.sqrt(mean_squared_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("RMSE Train: ", np.sqrt(mean_squared_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("MAE Test: ", mean_absolute_error scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1)))
print("MAE Train: ", mean_absolute_error scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1)))
print("R2 Test: ", r2_score scaler_y.inverse_transform(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1)))
print("R2 Train: ", r2_score scaler_y.inverse_transform(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1)))

```

Figura 31. Cálculo de Métricas de Evaluación para el Algoritmo GBR

En la Tabla 26 se presenta el valor de las métricas de evaluación según el conjunto de datos (entrenamiento y prueba) y la técnica de determinación de hiperparámetros. Al igual que en los otros algoritmos, se corrobora que el modelo tiene mucho más error de predicción con los hiperparámetros que se determinaron aleatoriamente y también se puede ver que existe un poco de sobreajuste.

Tabla 26. Valor de Métricas de Evaluación para el Algoritmo GBR

Algoritmo	Tipo de Conjunto de Datos	Hiperparámetros Determinados Aleatoriamente				Hiperparámetros Determinados con GridSearchCV			
		MAE	RMSE	MSE	R <sup>2</sup>	MAE	RMSE	MSE	R <sup>2</sup>
Regresor de Gradiente Aumentado	Entrenamiento	16.11	24.42	596.12	0.99	7.44	10.11	102.12	1.00
	Prueba	27.89	35.76	1278.94	0.78	15.48	21.00	440.98	0.92

Al igual que en los anteriores algoritmos, se hará una selección de las variables más relevantes para el algoritmo, utilizando el método RFE. Para llevar a cabo este procedimiento, también nos ayudaremos de la función RFE del módulo `feature_selection` de la librería de `scikit-learn` y como estimador se tomará el mismo algoritmo Regresor de Gradiente Aumentado, ya que al igual que el Regresor de Bosques Aleatorios, es compatible con la función RFE. Al mismo tiempo nos apoyaremos de la función `GridSearchCV` con un número de pliegues igual a cinco, con el objetivo de conseguir el número óptimo de variables relevantes. En la siguiente tabla se muestra el código empleado y el resultado obtenido.

Tabla 27. Determinación del Número de Variables Óptimas para el Algoritmo GBR

Algoritmo	Código Python	Número de Variables Óptimas
Regresor de Gradiente Aumentado	<pre> ▶ from sklearn.ensemble import RandomForestRegressor from sklearn.feature_selection import RFE from sklearn.model_selection import GridSearchCV  param_grid = {'n_features_to_select':list(range(1, 37))}  estimator = GradientBoostingRegressor(learning_rate=0.1,  n_estimators=80,  subsample=0.8,  max_depth=3,  random_state=1)  rfe_cv = GridSearchCV(RFE(estimator, step=1), param_grid, cv=5) rfe_cv.fit(X_train, y_train.ravel()) </pre>	<pre> ▶ rfe_cv.best_params_ 📄 {'n_features_to_select': 4} </pre>

En la Tabla 28 se presentan las cuatro variables óptimas seleccionadas anteriormente, con el método de RFE. Al igual que en los anteriores algoritmos, nos apoyaremos en el atributo `.support_` de la función RFE, para obtener dichas variables.

Tabla 28. Subconjunto Óptimo de Variables para el Algoritmo GBR

Algoritmo	Subconjunto Óptimo de Variables
Regresor de Gradiente Aumentado	<ul style="list-style-type: none"> <li>- GDX ETF</li> <li>- Dema Mund Oro</li> <li>- GDX ETF_lag1</li> <li>- Prec Oro_lag1</li> </ul>

En la Tabla 29, podemos observar que el desempeño del modelo mejoró, ya que se disminuyó un poco el sobreajuste, mostrando un error menor entre el conjunto de entrenamiento y prueba. Adicionalmente, se redujo significativamente el número de variables de entrada, ya que de las 37 que teníamos al principio ahora se tiene sólo 4.

Tabla 29. Métricas de Evaluación usando Subconjunto Óptimo de Variables con GBR

Algoritmo	Tipo de Conjunto de Datos	Métricas de Evaluación			
		MAE	RMSE	MSE	R <sup>2</sup>
Regresor de Gradiente Aumentado	Entrenamiento	8.73	12.30	151.32	1.00
	Prueba	13.38	17.87	319.33	0.95

Por otro lado, al igual que el algoritmo Regresor de Bosques Aleatorios, el Regresor de Gradiente Aumentado también cuenta con el atributo `feature_importances_`, para identificar qué variables de entrada son las más relevantes para el modelo. En la Tabla 30 se presenta un ranking de variables de entrada en base al grado de importancia que le otorga el algoritmo a cada variable.

Tabla 30. Ranking de Variables de Entrada en el Algoritmo GBR

Variable de Entrada		Ranking
Abreviatura	Nombre	
Prec Oro_lag1	Precio del oro del día anterior	1
GDX ETF	VanEck Gold Miners ETF	2
GDX ETF_lag1	VanEck Gold Miners ETF del día anterior	3
Dema Mund Oro	Demanda Mundial de Oro	4

### 3.5.7. Elección del Mejor Modelo Predictivo

La elección del mejor modelo predictivo se hará en base a la comparación de las métricas de evaluación respecto al conjunto de prueba, donde los algoritmos se entrenaron con los hiperparámetros determinados con la técnica de búsqueda en cuadrícula con validación cruzada y con el subconjunto óptimo de variables de entrada, ya que es con lo cual se obtuvo los mejores resultados. En la Tabla 31, se puede ver que el mejor modelo para predecir el precio del oro es el SVR con un kernel radial (rbf), ya que es el que tiene los valores más óptimos en base a las métricas de evaluación, lo cual indica un mejor rendimiento.

Tabla 31. Comparación de Valor de Métricas de Evaluación de Modelos Predictivos

Modelo	Métricas de Evaluación			
	MAE	RMSE	MSE	R2
Regresión con Máquinas de Vectores de Soporte (kernel = rbf)	6.20	8.37	70.01	0.99
Regresor de Bosques Aleatorios	12.89	17.01	289.39	0.95
Regresor de Gradiente Aumentado	13.38	17.87	319.33	0.95

### 3.5.8. Predicción

Luego de seleccionar el mejor modelo predictivo, se evaluará su rendimiento con nuevos datos que estarán comprendidos entre el 01/01/2023 y el 31/01/2023 con el fin de evaluar su capacidad predictiva. En la Tabla 32 se observa que el modelo presenta un desempeño satisfactorio, ya que las predicciones son cercanas a los valores reales.

Tabla 32. Predicción de Precio de Oro con Nuevos Datos

<b>Fecha</b>	<b>Precio de Oro Real</b>	<b>Precio de Oro Predicho</b>
3/01/2023	1839.49	1833.74
4/01/2023	1854.09	1855.67
5/01/2023	1833.19	1847.38
6/01/2023	1865.71	1867.69
9/01/2023	1871.59	1871.58
10/01/2023	1877.27	1876.96
11/01/2023	1876.38	1879.81
12/01/2023	1896.86	1898.92
13/01/2023	1920.21	1909.80
17/01/2023	1908.39	1903.68
18/01/2023	1903.76	1906.64
19/01/2023	1931.39	1917.82
20/01/2023	1926.57	1929.38
23/01/2023	1931.30	1915.54
24/01/2023	1937.45	1939.28
25/01/2023	1945.93	1945.37
26/01/2023	1928.99	1937.19
27/01/2023	1927.34	1918.90
30/01/2023	1922.52	1919.92
31/01/2023	1927.88	1925.22

## CAPÍTULO IV

### ANÁLISIS Y DISCUSIÓN DE RESULTADOS

#### 4.1. ANÁLISIS DE RESULTADOS

El desarrollo del modelo predictivo para la fluctuación del precio del oro dio como resultado un rendimiento satisfactorio, logrando una precisión destacada en la predicción de los datos históricos, en particular el algoritmo de Regresión con Máquinas de Vectores de Soporte (kernel = rbf) que fue el que mejor se ajustó a la naturaleza de los datos, proporcionando buenos resultados.

A continuación, se presenta un gráfico de líneas que compara los valores reales del precio del oro con las predicciones generadas por el modelo, donde se pueden distinguir tres líneas: el precio del oro real (en azul), el precio predicho con los datos de entrenamiento (en plomo) y el precio predicho con los datos de prueba (en anaranjado). Además, se visualiza que la línea azul que representa el precio real, es apenas visible en casi todos los tramos, ya que las predicciones son tan cercanas a los valores reales que prácticamente se superponen, lo cual indica que el modelo ha captado con precisión las fluctuaciones del precio del oro logrando predicciones muy ajustadas tanto en los datos de entrenamiento como en los de prueba.

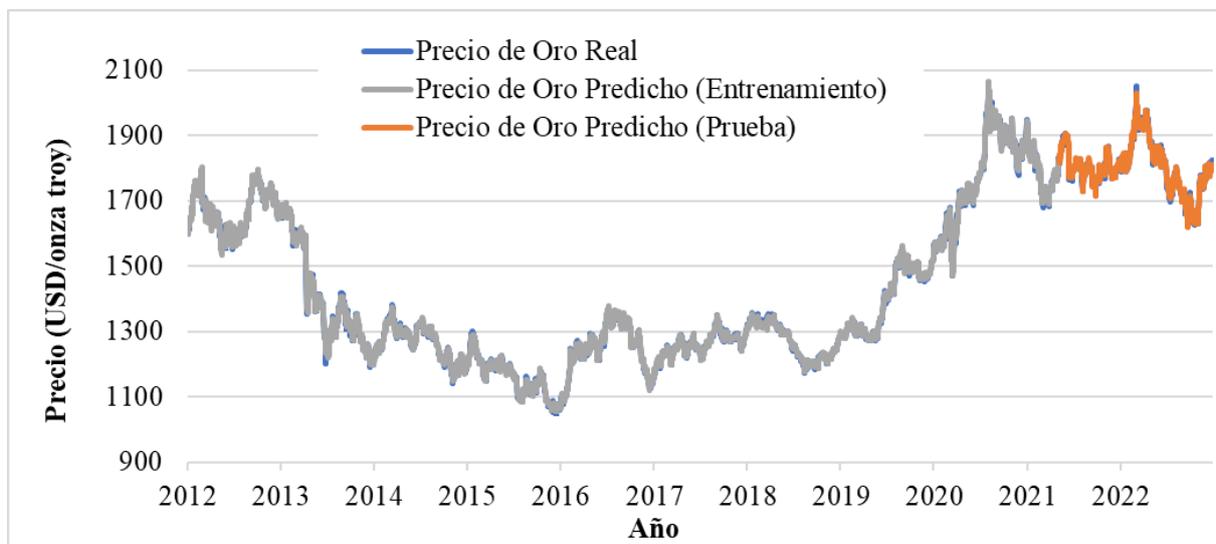


Figura 32. Comparación del Precio del Oro Real con el Precio del oro Predicho

Para lograr que el modelo haga buenas predicciones se realizaron varios análisis clave como: el ajuste de hiperparámetros, selección de variables relevantes y la selección del mejor algoritmo. Estos pasos fueron muy importantes para mejorar la eficiencia del modelo, y se analizan a continuación:

#### 4.1.1. Análisis de Ajuste de Hiperparámetros

Primero se determinó la configuración de los hiperparámetros empleando dos métodos distintos: la asignación aleatoria de valores y la búsqueda exhaustiva mediante validación cruzada (GridSearchCV). La asignación aleatoria consistió en seleccionar valores de hiperparámetros de manera no sistemática, mientras que GridSearchCV se aplicó para realizar una búsqueda exhaustiva dentro de un rango predefinido de valores para cada hiperparámetro. Los valores asignados a cada hiperparámetro según el algoritmo utilizado se detallan en la Tabla 33.

Tabla 33. Valores de Hiperparámetros Asignados a cada Algoritmo

Algoritmo	Hiperparámetros Determinados Aleatoriamente	Hiperparámetros Determinados con GridSearchCV
Regresión con Máquinas de Vectores de Soporte (kernel = rbf)	- gamma = 0.01 - C = 1 - epsilon = 0.01	- gamma = 0.01 - C = 100 - epsilon = 0.01
Regresor de Bosques Aleatorios	- n_estimators = 100 - max_depth = 10 - min_samples_split = 2 - min_samples_leaf = 1 - max_features = None - bootstrap = False	- n_estimators = 50 - max_depth = 5 - min_samples_split = 10 - min_samples_leaf = 4 - max_features = None - bootstrap = True
Regresor de Gradiente Aumentado	- learning_rate = 0.1 - n_estimators = 50 - subsample = 0.8 - max_depth = 1	- learning_rate = 0.1 - n_estimators = 80 - subsample = 0.8 - max_depth = 3

Luego se compararon las métricas de rendimiento de cada algoritmo, teniendo en cuenta el método de selección de hiperparámetros y el conjunto de datos utilizado. Los resultados de esta comparación se pueden ver en la Tabla 34:

Tabla 34. Comparación de Rendimiento de Algoritmos según el Tipo de Selección de Hiperparámetros

Algoritmo	Tipo de Conjunto de Datos	Hiperparámetros Determinados Aleatoriamente				Hiperparámetros Determinados con GridSearchCV			
		MAE	RMSE	MSE	R <sup>2</sup>	MAE	RMSE	MSE	R <sup>2</sup>
Regresión con Máquinas de Vectores de Soporte (kernel = rbf)	Entrenamiento	9.66	12.99	168.72	1.00	4.88	6.51	42.44	1.00
	Prueba	47.01	56.70	3215.27	0.44	9.03	11.44	130.95	0.98
Regresor de Bosques Aleatorios	Entrenamiento	4.21	6.19	38.38	1.00	8.97	12.62	159.25	1.00
	Prueba	34.53	43.18	1864.72	0.68	15.47	21.04	442.66	0.92
Regresor de Gradiente Aumentado	Entrenamiento	16.11	24.42	596.12	0.99	7.44	10.11	102.12	1.00
	Prueba	27.89	35.76	1278.94	0.78	15.48	21.00	440.98	0.92

En base a los valores de las métricas de evaluación (MAE, RMSE, MSE, R<sup>2</sup>) de los conjuntos de entrenamiento y prueba de cada algoritmo, podemos ver en la Tabla 34 la existencia de sobreajuste en los modelos de Regresión con Máquinas de Vectores de Soporte y Regresor de Bosques Aleatorios cuando se utilizan hiperparámetros aleatorios, ya que presentan buen desempeño con el conjunto de entrenamiento, pero un bajo desempeño con el conjunto de prueba. No obstante, al aplicar GridSearchCV, el desempeño mejora significativamente en el conjunto de prueba, lo que reduce el sobreajuste y permite que los modelos sean más generalizables. En el caso del Regresor de Gradiente Aumentado, el ajuste de hiperparámetros utilizando el método aleatorio muestra signos de subajuste, lo que significa que el modelo no está aprovechando todo su potencial para hacer buenas predicciones en el conjunto de prueba, esto señala que el modelo aún no ha aprendido lo suficiente sobre los datos. Sin embargo, cuando se utilizan técnicas de optimización más exhaustivas como GridSearchCV para ajustar los hiperparámetros, se logra mejorar la capacidad predictiva del modelo.

De acuerdo con el análisis anterior, se confirma lo señalado por Fosca (2020), quien destaca que la optimización de los hiperparámetros es fundamental para el desarrollo de modelos predictivos. Esto refuerza la idea de que elegir bien estos parámetros no solo mejora el rendimiento del modelo, sino que también ayuda a evitar problemas como el sobreajuste y el subajuste.

Cabe mencionar que, además de GridSearchCV, existen varios métodos adicionales para la optimización de hiperparámetros, como la Optimización Bayesiana, la Búsqueda Aleatoria y la Optimización por Gradiente, que también pueden ser implementados dependiendo de las necesidades del modelo y los recursos disponibles.

#### 4.1.2. Análisis de Selección de Variables Relevantes

A través del análisis de selección de variables relevantes con la ayuda de la función de Eliminación Recursiva de Características (RFE), se observó que el conjunto óptimo de variables y su grado de importancia para cada algoritmo empleado es diferente, lo cual confirma lo dicho por David Wolpert, citado por Fosca (2020), que señala que cada algoritmo procesa y utiliza los datos de forma diferente. En la Tabla 35 podemos observar el conjunto de variables óptimas y el grado de importancia de cada una de ellas para cada algoritmo a través de un raking.

Tabla 35. Variables Óptimas Según el Algoritmo Utilizado

<b>Regresión con Máquinas de Vectores de Soporte (kernel = rbf)</b>	<b>Regresor de Bosques Aleatorios</b>	<b>Regresor de Gradiente Aumentado</b>	<b>Ranking</b>
Precio del oro del día anterior	Precio del oro del día anterior	Precio del oro del día anterior	1
iShares Silver Trust ETF	VanEck Gold Miners ETF	VanEck Gold Miners ETF	2
iShares Silver Trust ETF del día anterior	Producción Mundial de Oro	VanEck Gold Miners ETF del día anterior	3
VanEck Gold Miners ETF	Rentabilidad del Bono de Estados Unidos del día anterior	Demanda Mundial de Oro	4
VanEck Gold Miners ETF del día anterior	iShares Silver Trust ETF		5

Rentabilidad del Bono de Estados Unidos	6
Índice del Precio del Dólar	7
Rentabilidad del Bono de Estados Unidos del día anterior	8
Índice del Precio del Dólar del día anterior	9
Dow Jones Industrial Average	10
Dow Jones Industrial Average del día anterior	11
Inflación de Estados Unidos del día anterior	12
Inflación de Estados Unidos	13

Adicionalmente, a través de este análisis, en base a los tres algoritmos empleados, se puede destacar la importancia de las siguientes variables para predecir el precio del oro:

- ✓ **Precio del oro del día anterior:** Las tendencias en los mercados de los metales suelen mantenerse a corto plazo, por lo que el precio del día previo es un buen indicador de lo que podría pasar al día siguiente.
- ✓ **iShares Silver Trust ETF:** La plata y el oro están estrechamente relacionados, ya que ambos son vistos como refugios seguros cuando la situación económica se vuelve inestable; por lo tanto, cuando el precio de la plata cambia, puede afectar lo que los inversores esperan del oro, ya que suelen considerar ambos metales como buenas opciones para proteger su dinero en tiempos difíciles.
- ✓ **VanEck Gold Miners ETF:** La cotización de las acciones de las empresas mineras de oro reflejan cómo los inversionistas perciben la industria del oro, ya que si estas acciones suben o bajan, es probable que el precio del oro siga la misma dirección.
- ✓ **Rentabilidad del Bono de Estados Unidos:** Los bonos del gobierno de EE. UU. compiten con el oro como activos seguros, es por esto que cuando la rentabilidad de estos bonos aumenta, la demanda de oro puede disminuir, y viceversa.

- ✓ **Índice del Precio del Dólar:** El oro se cotiza en dólares, por lo que cuando el valor del dólar baja, el oro se vuelve más atractivo para los inversionistas, haciendo que suba su precio.
- ✓ **Producción y Demanda Mundial de Oro:** Estas variables son relevantes en los modelos Regresor de Bosques Aleatorios y Regresor de Gradiente Aumentado ya que la cantidad de oro que se produce y su demanda global, influyen en gran medida en cómo se establece el precio del oro en el mercado.

Algunas de estas variables son similares a las mencionadas por Castillo (2022), quien también destacó el índice del precio del dólar, la producción y demanda mundial de oro como factores influyentes en la fluctuación del precio del oro.

Por otro lado, con objetivo de examinar la importancia de la selección de variables relevantes, se hizo una comparación del rendimiento de los algoritmos cuando solo se ajustaron los hiperparámetros y se trabajó con todas las variables de entrada, y cuando se seleccionaron las variables más relevantes para cada algoritmo conservando el ajuste de hiperparámetros. Los resultados obtenidos en base a las métricas de evaluación se presentan en la tabla siguiente:

Tabla 36. Comparación de Rendimiento de Algoritmos en base al Ajuste de Hiperparámetros y Selección de Variables

Algoritmo	Tipo de Conjunto de Datos	Con Valores Óptimos de Hiperparámetros				Con Valores Óptimos de Hiperparámetros y Variables más Importantes			
		MAE	RMSE	MSE	R <sup>2</sup>	MAE	RMSE	MSE	R <sup>2</sup>
Regresión con Máquinas de Vectores de Soporte (kernel = rbf)	Entrenamiento	4.88	6.51	42.44	1.00	4.83	6.69	44.76	1.00
	Prueba	9.03	11.44	130.95	0.98	6.20	8.37	70.01	0.99
Regresor de Bosques Aleatorios	Entrenamiento	8.97	12.62	159.25	1.00	9.40	13.36	178.64	1.00
	Prueba	15.47	21.04	442.66	0.92	12.89	17.01	289.39	0.95
Regresor de Gradiente Aumentado	Entrenamiento	7.44	10.11	102.12	1.00	8.73	12.30	151.32	1.00
	Prueba	15.48	21.00	440.98	0.92	13.38	17.87	319.33	0.95

Se puede observar que, aunque los resultados del conjunto de entrenamiento en algunos modelos como el de Regresión con Máquinas de Vectores de Soporte (SVR) presentan un ligero aumento en el RMSE y MSE, esto sugiere que el modelo está evitando el sobreajuste. Sin embargo, en el conjunto de prueba los resultados muestran mejoras significativas, particularmente en el SVR, donde el MAE disminuye de 9.03 a 6.20, el RMSE de 11.44 a 8.37 y el MSE de 130.95 a 70.01; indicado que el modelo optimizado con menos variables predice con mayor precisión los datos no vistos. Por otro lado, en los modelos Regresor de Bosques Aleatorios y Regresor de Gradiente Aumentado, el efecto es similar, mientras que el rendimiento del conjunto de entrenamiento varía ligeramente, en el conjunto de prueba se observan mejoras notables en todas las métricas, reflejando una mejor capacidad de generalización.

En base a este análisis, se corrobora lo indicado por Fosca (2020), quien resalta que llevar a cabo un análisis detallado y seleccionar adecuadamente las variables, es fundamental para lograr una mejora notable en el rendimiento de los algoritmos. Este hallazgo también coincide con lo señalado por Huillca y Quispe (2019), quienes afirman que la precisión de los modelos depende en gran medida de la correcta elección tanto de las variables como de las técnicas empleadas.

#### **4.1.3. Análisis de Selección del Mejor Algoritmo**

En el presente estudio, se compararon tres algoritmos: Regresión con Máquinas de Vectores de Soporte (SVR), Regresor de Bosques Aleatorios (RFR) y Regresor de Gradiente Aumentado (GBR). Los resultados mostraron que el SVR con un kernel radial fue el mejor modelo para predecir el precio del oro, ya que mostró un mejor desempeño con los datos de entrenamiento con: MAE = 4.83, RMSE = 6.69, MSE = 44.76 y  $R^2 = 1.00$  y también con los datos de prueba con: MAE = 6.20, RMSE = 8.37, MSE = 70.01 y  $R^2 = 0.99$ .

Este hallazgo coincide con los resultados de Castillo (2022), quien también identificó al SVR como el modelo más efectivo para predecir el precio del oro, mostrando mejor precisión ante la regresión lineal y el árbol de regresión. Adicionalmente, el estudio de Fosca (2020), aunque enfocado en el precio del cobre, también identificó al SVR como el algoritmo más efectivo para la predicción, superando a la regresión lineal y al modelo convencional ARIMA. Esta

coincidencia refuerza la efectividad del SVR en el desarrollo de modelos predictivos, confirmando su relevancia en diferentes contextos del mercado de metales preciosos.

#### 4.2. CONTRASTACIÓN DE LA HIPÓTESIS

De acuerdo con la hipótesis planteada, que buscaba mantener el error absoluto medio (MAE) del modelo predictivo por debajo de 10 USD/onza troy utilizando nuevos datos, se evaluó el modelo desarrollado, con datos recopilados desde el 01/01/2023 hasta el 31/01/2023, logrando un MAE de 5.19 USD/onza troy, como se muestra en la Tabla 37. Este resultado no solo respalda la hipótesis, sino que también demuestra que el método utilizado en este estudio realmente funciona.

Tabla 37. Evaluación de Errores de Predicción con Nuevos Datos

Fecha	Precio de Oro Real	Precio de Oro Predicho	Error Absoluto
3/01/2023	1839.49	1833.74	5.75
4/01/2023	1854.09	1855.67	1.58
5/01/2023	1833.19	1847.38	14.19
6/01/2023	1865.71	1867.69	1.98
9/01/2023	1871.59	1871.58	0.01
10/01/2023	1877.27	1876.96	0.31
11/01/2023	1876.38	1879.81	3.43
12/01/2023	1896.86	1898.92	2.06
13/01/2023	1920.21	1909.80	10.41
17/01/2023	1908.39	1903.68	4.71
18/01/2023	1903.76	1906.64	2.88
19/01/2023	1931.39	1917.82	13.57
20/01/2023	1926.57	1929.38	2.81
23/01/2023	1931.30	1915.54	15.76
24/01/2023	1937.45	1939.28	1.83
25/01/2023	1945.93	1945.37	0.56
26/01/2023	1928.99	1937.19	8.20
27/01/2023	1927.34	1918.90	8.44
30/01/2023	1922.52	1919.92	2.60
31/01/2023	1927.88	1925.22	2.66
<b>Error Absoluto Medio (MAE)</b>			<b>5.19</b>

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. CONCLUSIONES**

- ✓ Se logró desarrollar un modelo predictivo que permite anticipar la fluctuación del precio del oro utilizando técnicas de aprendizaje automático (machine learning), proporcionando una herramienta fundamental para el análisis del mercado de metales preciosos y su influencia en la industria minera.
- ✓ Las principales variables que influyen y tienen mayor relación con la fluctuación del precio del oro en base a los tres algoritmos analizados son: el Precio del Oro del día anterior, iShares Silver Trust ETF, VanEck Gold Miners ETF, Rentabilidad del Bono de Estados Unidos, Índice del Precio del Dólar, Producción Mundial de Oro y Demanda Mundial de Oro.
- ✓ Se logró identificar los valores óptimos de los hiperparámetros de cada algoritmo a través de la aplicación de técnicas de aprendizaje automático, lo que permitió ajustar los modelos de manera más precisa a los datos históricos, mejorando la precisión y eficiencia de las predicciones.
- ✓ El mejor algoritmo para predecir el precio del oro es el de Regresión con Máquinas de Vectores de Soporte con un kernel radial, ya que fue el que mostró el mejor rendimiento en comparación con los otros dos algoritmos analizados. Este modelo obtuvo métricas destacadas tanto en los datos de entrenamiento como en los de prueba, lo que demuestra su efectividad superior en las predicciones.

## 5.2. RECOMENDACIONES

- ✓ Se recomienda a estudiantes e investigadores interesados en desarrollar modelos predictivos que continúen mejorando sus metodologías, como por ejemplo: incorporando datos en tiempo real. Del mismo modo, se deben realizar pruebas regulares para determinar cómo responden los modelos en diversos escenarios.
- ✓ Se sugiere realizar estudios adicionales para explorar otras variables que puedan influir o tener relación con la fluctuación del precio del oro, ya que esto ayudará a profundizar el análisis y a entender mejor los factores que influyen en este mercado.
- ✓ Se aconseja a quienes tienen interés en el desarrollo de modelos predictivos a ajustar regularmente los hiperparámetros conforme se disponen de nuevos datos. Además, sería bueno explorar otros métodos de aprendizaje automático, como la optimización bayesiana y el uso de algoritmos genéticos para optimizar este proceso.
- ✓ Se recomienda emplear el algoritmo SVR como base para futuras investigaciones y modelos predictivos, considerando su aplicación en otros metales preciosos o materias primas.

## REFERENCIAS BIBLIOGRÁFICAS

- Agrawal, T. 2020. Introduction to Hyperparameters. In *Hyperparameter Optimization in Machine Learning: Make Your Machine Learning and Deep Learning Models More Efficient*. Primera edición. Apress Berkeley. p. 1-30.
- Asante, D; Arbi, TO; Oghenerurie, J; Nii, B; Yassir, M. 2022. Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions. *Journal of Petroleum Science and Engineering* 208:109244.
- Bardales, CD y Zamora, C. 2019. Evaluación de los criterios de selección de un método de explotación mediante la inteligencia artificial. Caso práctico yacimiento de hierro en Ventanillas, Yonán, Cajamarca, 2019. Tesis de pre-grado. Cajamarca, Perú, Universidad Privada del Norte.
- Breiman, L. 2001. Random Forests. *Machine Learning* 45:5–32.
- Camastra, F y Vinciarelli, A. 2008. Machine Learning. In *Machine Learning for Audio, Image and Video Analysis: Theory and Applications*. London, Springer. p. 83–89.
- Carles, J. 2023. Modelización de los factores que inciden en el rendimiento académico de los estudiantes universitarios con técnicas de estadística multivariante y de machine learning. Trabajo de fin de grado. Valencia, España, Universidad Politécnica de Valencia.
- Carrión, AK. 2023. Análisis del precio del oro mediante Inteligencia Artificial – proyecto integrador. Tesis de pregrado. Quito, Ecuador, Universidad Central del Ecuador.
- Castellano, K. 2020. Análisis de la evolución del precio del oro. Memoria de trabajo de grado. San Cristóbal de La Laguna, España, Universidad de la Laguna.
- Castillo, OA. 2022. Desarrollo de modelos predictivos de regresión en la industria minera mediante el uso de algoritmo de machine learning. Tesis de pre-grado. Lima, Perú, Universidad Nacional Mayor de San Marcos.
- Díaz, MH. 2017. Uso de las redes neuronales artificiales en el modelado del ensayo de resistencia a compresión de concreto de construcción según la norma ASTM C39/C39M. Tesis de pre-grado. Cajamarca, Perú, Universidad Nacional de Cajamarca.

- Fosca, A. 2020. Desarrollo de un modelo para la predicción del precio del cobre empleando herramientas de Machine Learning. Tesis de pre-grado. Lima, Perú, Pontificia Universidad Católica del Perú.
- Gopaul, K. 22 jun. 2023. The evolving picture of global gold production (en línea, blog). World Gold Council. Consultado 27 jul. 2023. Disponible en <https://www.gold.org/>.
- Guo, J; Farhang-Razi, V; Algra, P. 2019. AI: A Glossary of Terms. In Ranschaert, E., Morozov, S., Algra, P. (eds). Artificial Intelligence in Medical Imaging. Springer. p. 347-373.
- Hernández, R; Fernández, C; Baptista, P. 2014. Metodología de la investigación. Sexta edición. México D.F., México, McGRAW-HILL.
- Idrogo, YP. 2022. Machine learning aplicado al control de la fragmentación de rocas en la voladura de minas a tajo abierto. Tesis de pre-grado. Lima, Perú, Universidad Nacional de Ingeniería.
- Investing. 2024. Datos históricos oro (en línea, sitio web). Consultado el 15 oct. 2024. Disponible en <https://es.investing.com/>.
- Jadue, RA. 2021. Entrenamiento de algoritmos de aprendizaje de máquinas para predecir los band gaps en paneles de metamateriales. Tesis de pre-grado. Santiago de Chile, Chile, Universidad de Chile.
- Kwak, S; Kim, J; Ding, H; Xu, X; Chen, R; Guo, J; Fu, H. 2022. Machine learning prediction of the mechanical properties of  $\gamma$ -TiAl alloys produced using random forest regression model. Journal of Materials Research and Technology 18:520-530.
- Laudon, KC y Laudon, JP. 2012. Sistemas de Información Gerencial. Decimosegunda edición. México D.F., México, Pearson Educación.
- López, A. 2007. Metales preciosos: El Oro. Boletín de la Real Academia de Córdoba de Ciencias, Bellas Letras y Nobles Artes 86(152):345-353.
- Madhavan, S; Sturdevant, M; Kienzler, R. 2019. Introduction to machine learning (en línea). International Business Machines (IBM). Consultado 15 jun. 2023. Disponible en <https://www.ibm.com/mx-es>.
- Makala, D y Li, Z. 2021. Prediction of gold price with ARIMA and SVM. Journal of Physics: Conference Series 1767:012022.

- Manjula, KA y Karthikeyan, P. 2019. Gold Price Prediction using Ensemble based Machine Learning Techniques. 3rd International Conference on Trends in Electronics and Informatics (ICOEI). Tirunelveli, India. p. 1360-1364.
- Pan, F. 2023. Machine Learning. In Daya Sagar, BS; Cheng, Q; McKinley, J; Agterberg, F. (eds). Encyclopedia of Mathematical Geosciences. Encyclopedia of Earth Sciences Series. Springer. p. 781–784.
- Raschka, S y Mirjalili, V. 2019. Python Machine Learning: Machine Learning and Deep Learning with Python, scikitlearn and TensorFlow. Tercera edición. Birmingham, Reino Unido, Packt Publishing Ltd.
- Real Academia Española. 2023. Diccionario de la lengua española (en línea). Consultado 18 jun. 2023. Disponible en <https://www.rae.es/>.
- Saavedra, F. 2014. El precio del oro durante la gran recesión desde una perspectiva austriaca. Procesos de Mercado: Revista Europea de Economía Política 11(1):137-188.
- Schölkopf, B y Smola, AJ. 2002. Learning with Kernels. Cambridge. MIT Press.
- Scikit-learn. 2020. scikit-learn user guide (Release 0.23.2). Scikit-learn.
- Shafiee, S y Topal, E. 2010. An overview of global gold market and gold price forecasting. Resources Policy 35(3):178-189.
- Universidad Anáhuac. 2023. Modelos predictivos con Machine Learning (en línea, curso). edX. Consultado 31 jul. 2023. Disponible en <https://www.edx.org/es>.
- Vega, JF. 2019. Modelo de pronóstico de rendimiento académico de alumnos en los cursos del programa de estudios básicos de la Universidad Ricardo Palma usando algoritmos de Machine Learning. Tesis de maestría. Lima, Perú, Universidad Ricardo Palma.
- World Gold Council. 2023a. Annual gold demand soars to a new decade high in 2022 (en línea, sitio web). Consultado 28 jul. 2023. Disponible en <https://www.gold.org/>.
- World Gold Council. 2023b. Gold Supply (en línea, sitio web). Consultado 28 jul. 2023. Disponible en <https://www.gold.org/>.
- World Gold Council. 2023c. Historical demand and supply (en línea, sitio web). Consultado 26 jul. 2023. Disponible en <https://www.gold.org/>.

- World Gold Council. 2023d. Gold Demand Trends Full Year 2022 (en línea, sitio web). Consultado 27 jul. 2023. Disponible en <https://www.gold.org/>.
- World Gold Council. 2024. Gold Market Commentary: Gold hit new highs in 2023 (en línea, sitio web). Consultado 19 set. 2024. Disponible en <https://www.gold.org/>.
- Zhang, H; Peng, J; Wang, R; Zhang, M; Gao, C; Yu, Y. 2023. Use of random forest based on the effects of urban governance elements to forecast CO2 emissions in Chinese cities. *Heliyon* 9(6):e16693.
- Zhang, T; Lin, W; Vogelmann, AM; Zhang, M; Xie, S; Qin, Y; Golaz, JC. 2021. Improving Convection Trigger Functions in Deep Convective Parameterization Schemes Using Machine Learning. *Journal of Advances in Modeling Earth Systems* 13(5):e2020MS002365.

## **ANEXOS**

Anexo N°1. Gráficos de Dispersión del Precio del Oro con las Variables de Entrada

Anexo N°2. Predicción del Precio del Oro con el Algoritmo SVR con Kernel Radial

Anexo N°3. Predicción del Precio del Oro con el Algoritmo RFR

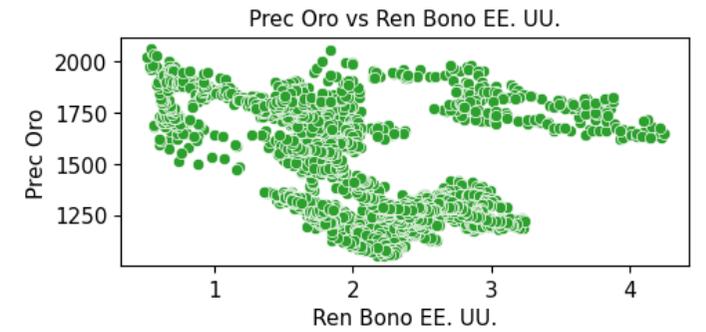
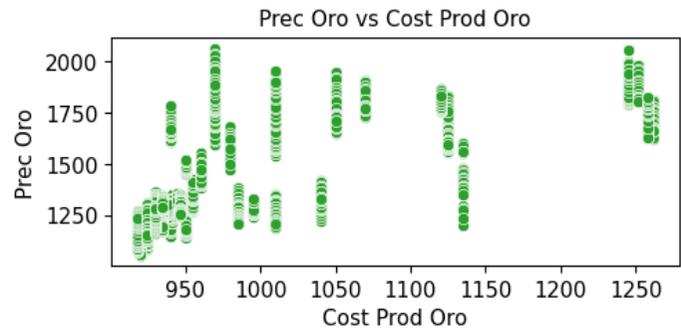
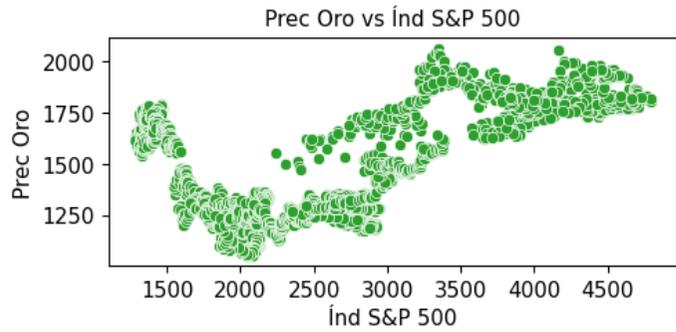
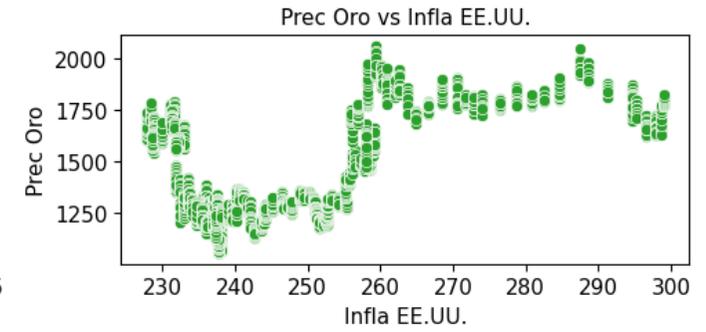
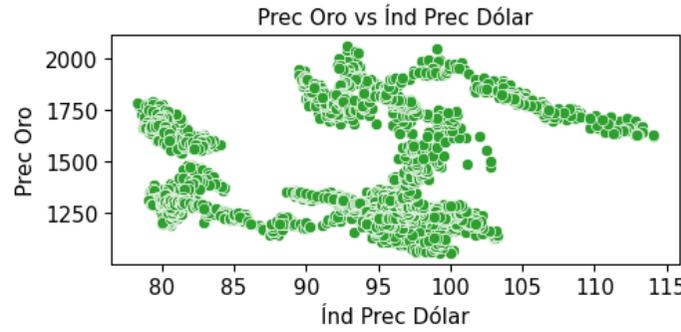
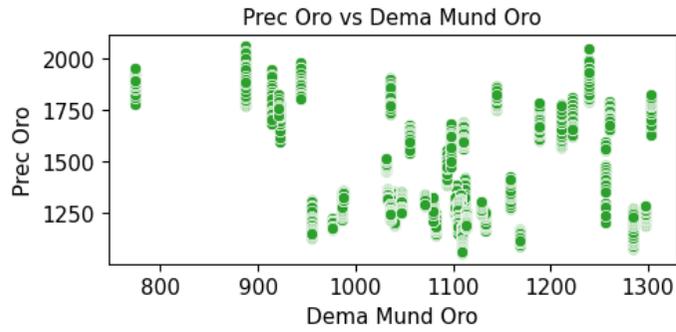
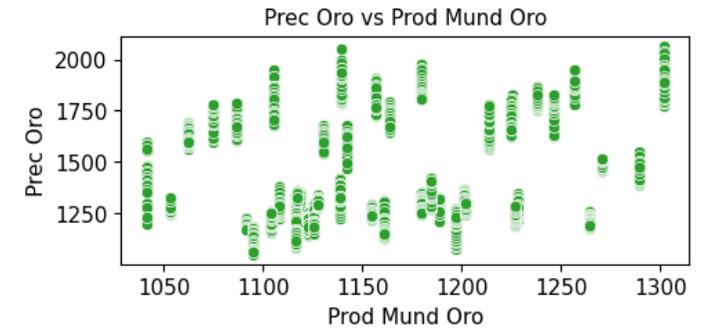
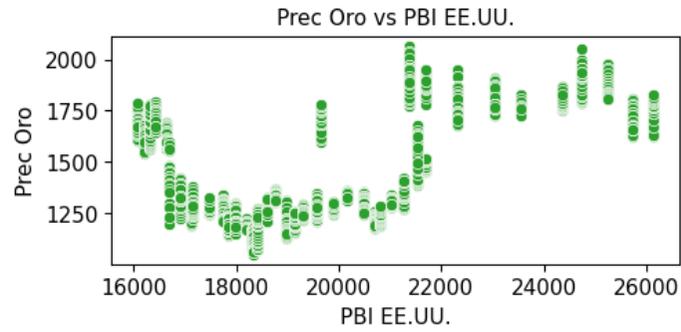
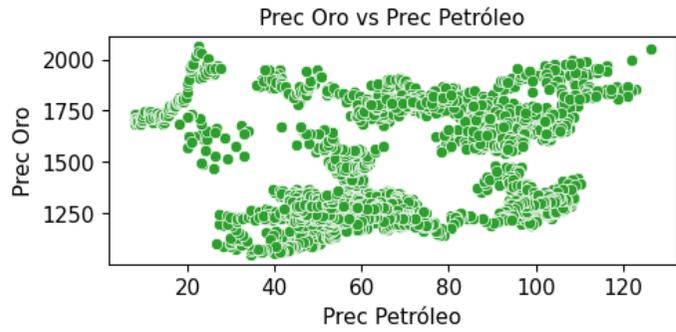
Anexo N°4. Predicción del Precio del Oro con el Algoritmo GBR

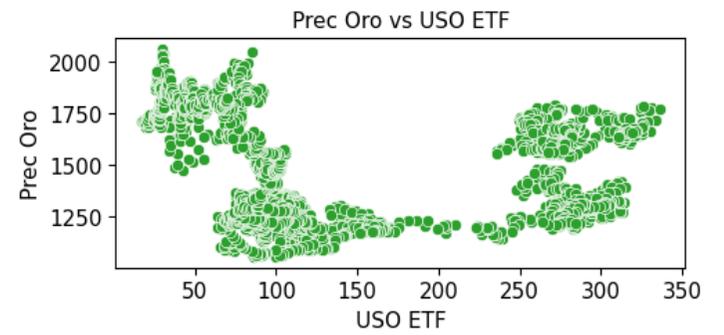
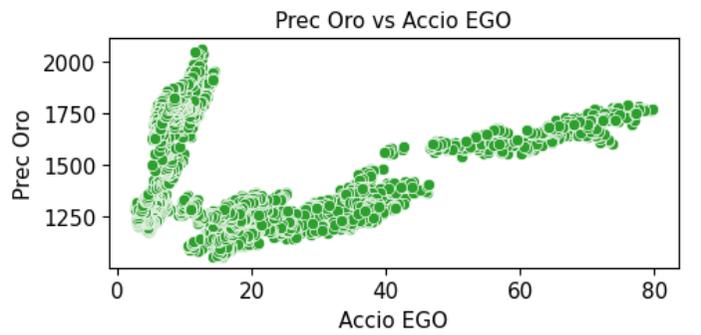
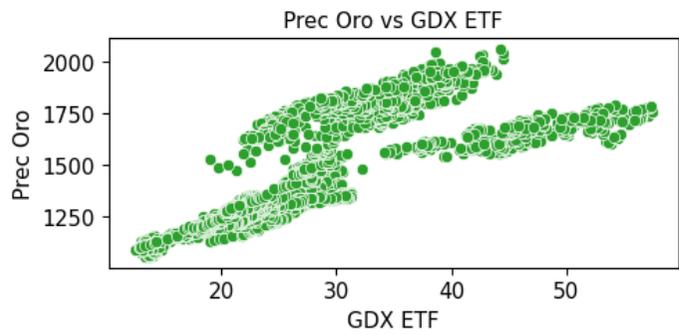
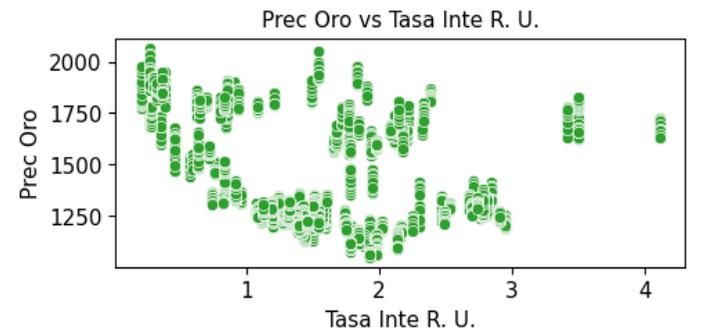
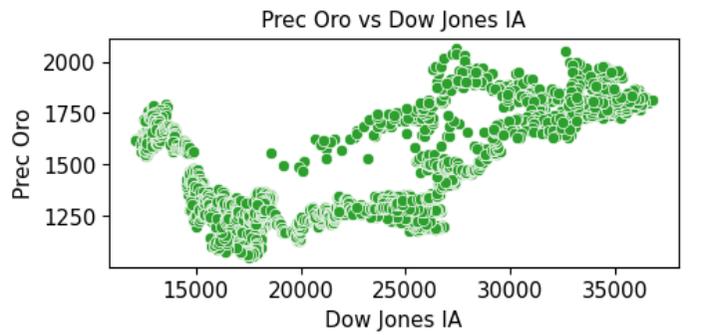
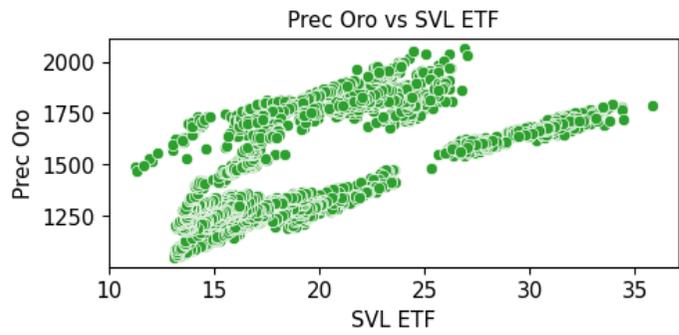
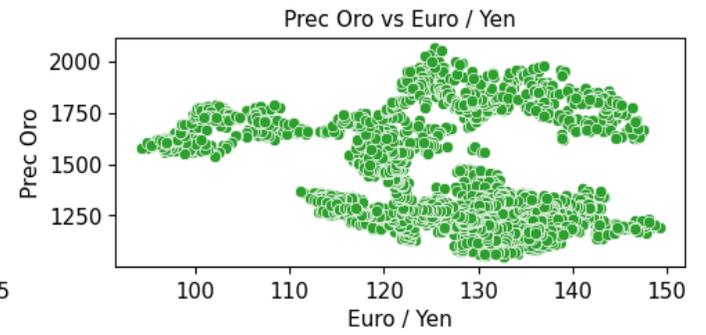
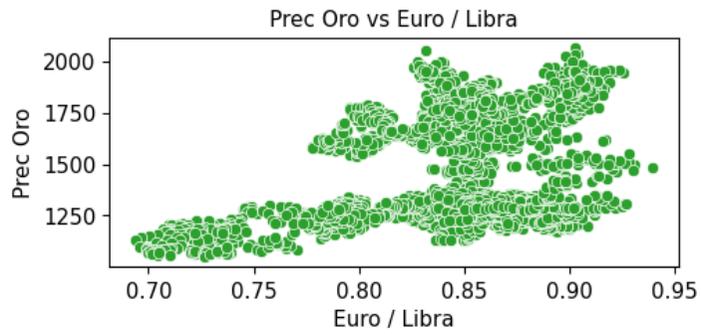
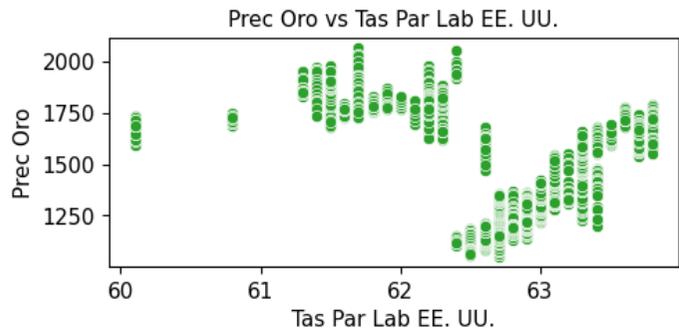
Anexo N°5. Modelo Predictivo SVR con Kernel Radial

Anexo N°6. Resumen Mensual de Datos utilizados para el Modelo Predictivo

# Anexo N°1

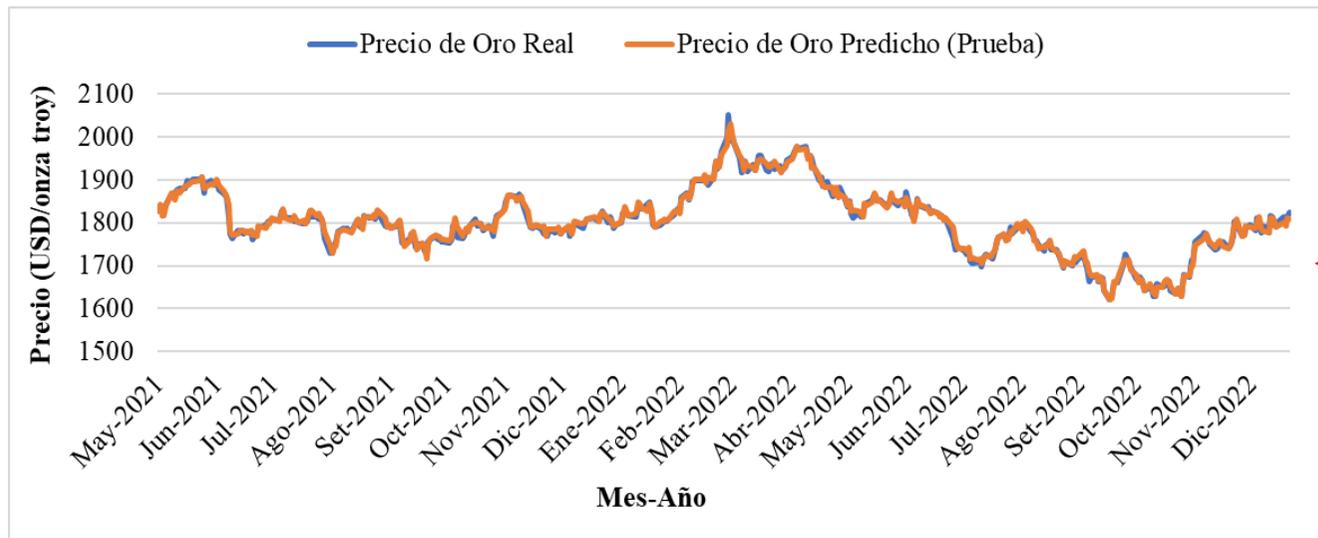
## Gráficos de Dispersión del Precio del Oro con las Variables de Entrada





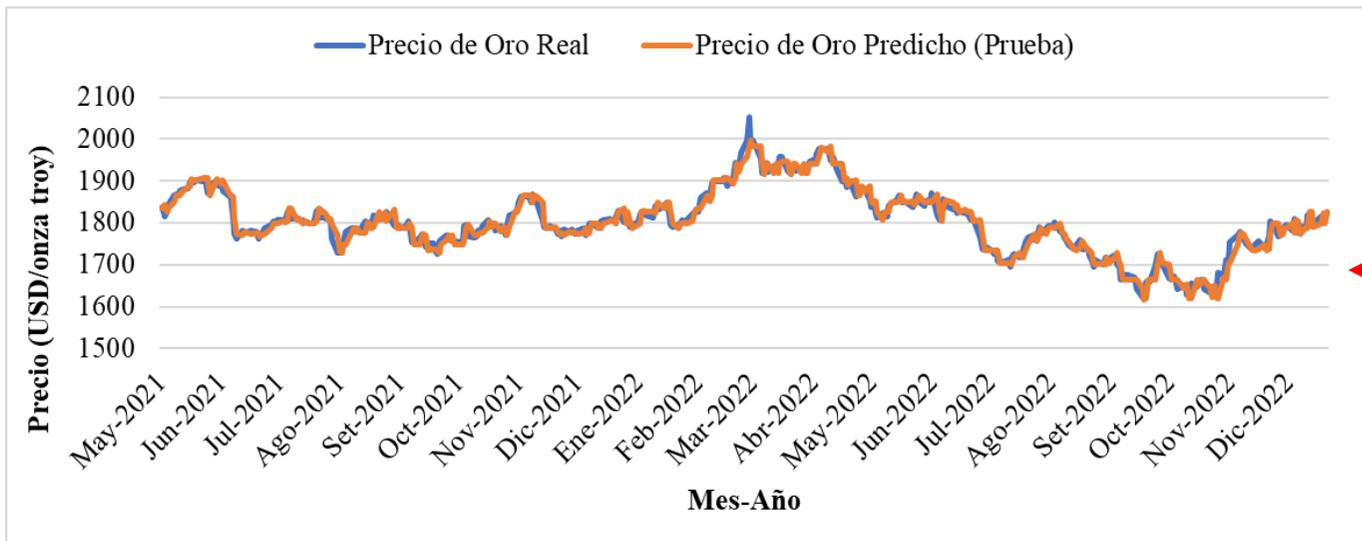
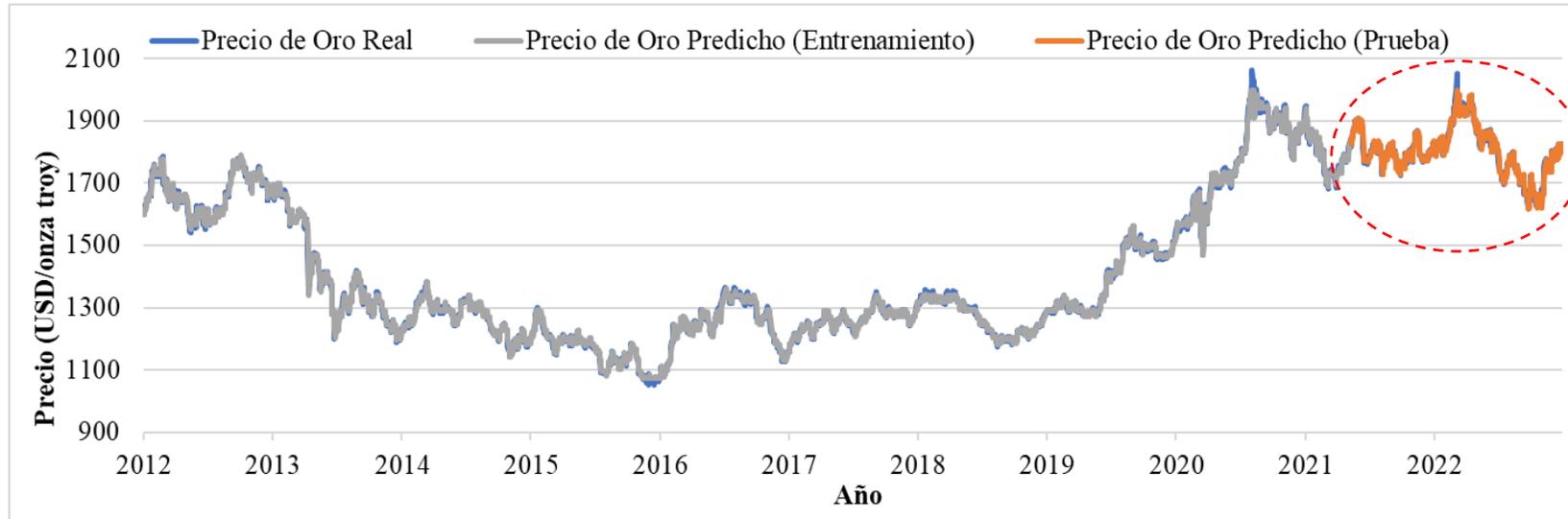
Anexo N°2

Predicción del Precio del Oro con el Algoritmo SVR con Kernel Radial



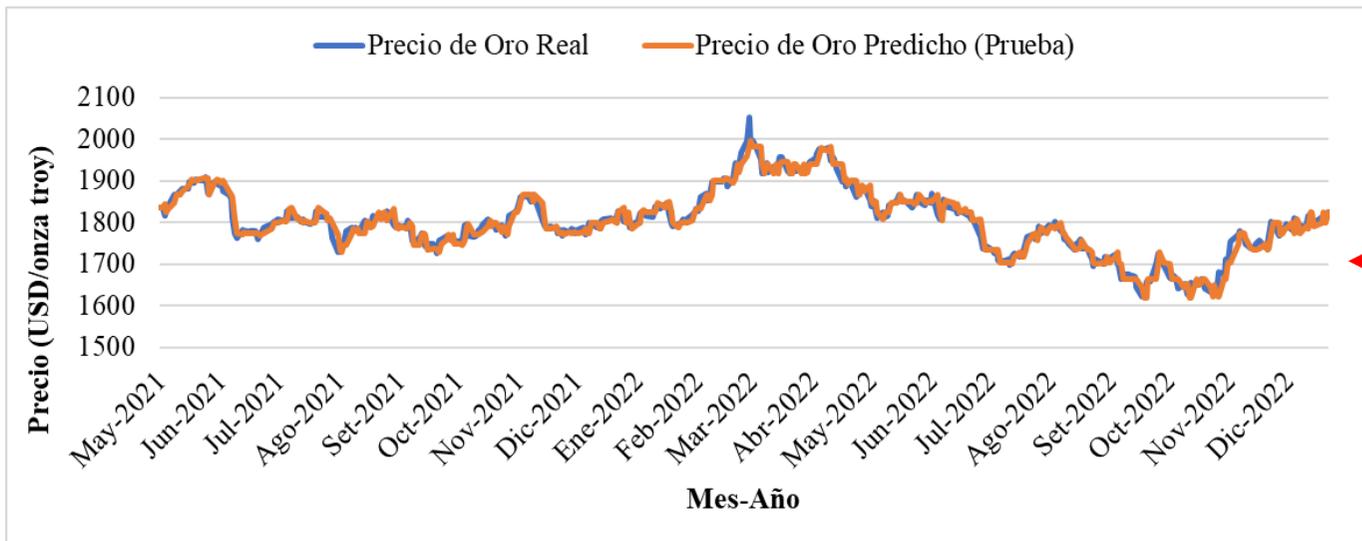
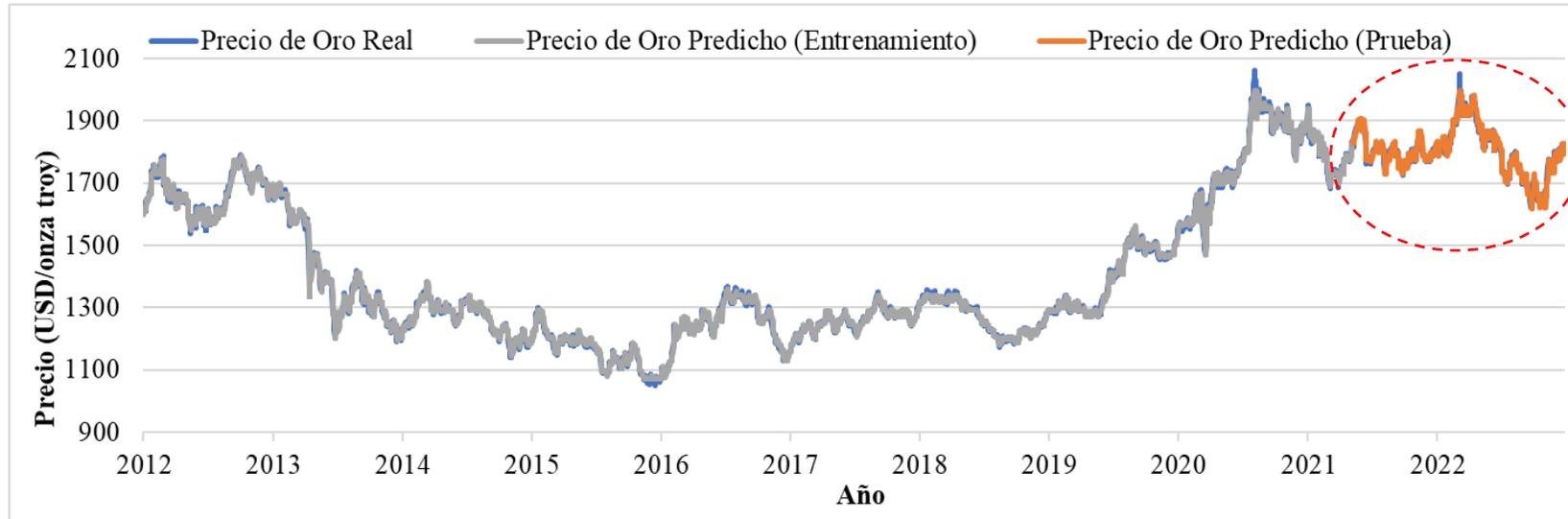
### Anexo N°3

#### Predicción del Precio del Oro con el Algoritmo RFR



Anexo N°4

Predicción del Precio del Oro con el Algoritmo GBR



## Anexo N°5

### Modelo Predictivo SVR con Kernel Radial

```
# Importación y preparación de base de datos
```

```
import pandas as pd
```

```
df = pd.read_excel('/content/DB Oro.xlsx')
```

```
num_lags = 1
```

```
df_lags = pd.DataFrame()
```

```
for lag in range(1, num_lags + 1):
```

```
    df_shifted = df.shift(lag)
```

```
    df_shifted.columns = [f'{col}_lag{lag}' for col in df.columns]
```

```
    df_lags = pd.concat([df_lags, df_shifted], axis=1)
```

```
df_with_lags = pd.concat([df, df_lags], axis=1)
```

```
df_with_lags = df_with_lags.dropna()
```

```
# Definición de variables de entrada y variable de salida
```

```
x = df_with_lags.drop(["Fecha", "Prec Oro", "Fecha_lag1"], axis=1)
```

```
x = pd.DataFrame(x)
```

```
x.columns = x.columns
```

```
y = df_with_lags["Prec Oro"].values.reshape(-1, 1)
```

```
# Estandarización de datos
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
scaler_x = MinMaxScaler()
```

```
scaler_y = MinMaxScaler()
```

```
x = scaler_x.fit_transform(x)
```

```
y = scaler_y.fit_transform(y)
```

```
# División de datos (entrenamiento y prueba)
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.15, shuffle=False)
```

```
# Algoritmo SVR (kernel = rbf) con hiperparámetros óptimos
```

```
from sklearn.svm import SVR
```

```
svr_rbf = SVR(kernel='rbf', gamma=0.01, C=100, epsilon=0.01)
```

```
svr_rbf.fit(x_train, y_train)
```

```

# Entrenamiento de algoritmo
y_train_pred = svr_rbf.predict(x_train)

# Validación de algoritmo
y_test_pred = svr_rbf.predict(x_test)

# Cálculo de métricas de evaluación
import numpy as np
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error

print("MSE Test: ", mean_squared_error(scaler_y.inverse_transform(y_test.reshape(-1, 1)),
scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("MSE Train: ", mean_squared_error(scaler_y.inverse_transform(y_train.reshape(-1, 1)),
scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("RMSE Test: ", np.sqrt(mean_squared_error(scaler_y.inverse_transform
(y_test.reshape(-1, 1)), scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("RMSE Train: ", np.sqrt(mean_squared_error(scaler_y.inverse_transform
(y_train.reshape(-1, 1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("MAE Test: ", mean_absolute_error(scaler_y.inverse_transform(y_test.reshape(-1, 1)),
scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("MAE Train: ", mean_absolute_error(scaler_y.inverse_transform(y_train.reshape(-1,
1)), scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))
print("R2 Test: ", r2_score(scaler_y.inverse_transform(y_test.reshape(-1, 1)),
scaler_y.inverse_transform(y_test_pred.reshape(-1, 1))))
print("R2 Train: ", r2_score(scaler_y.inverse_transform(y_train.reshape(-1, 1)),
scaler_y.inverse_transform(y_train_pred.reshape(-1, 1))))

```

Anexo N°6

Resumen Mensual de Datos utilizados para el Modelo Predictivo

Año	Mes	Prec Petróleo	PBI EE.UU.	Prod Mund Oro	Dema Mund Oro	Índ Prec Dólar	Infla EE.UU.	Índ S&P 500	Cost Prod Oro	Ren Bono EE.UU.	Tas Par Lab EE.UU.	Euro/ Libra	Euro/ Yen	SVL ETF	Dow Jones IA	Tasa Inte R. U.	GDX ETF	Accio EGO	USO ETF	Prec Oro
2012	Ene	100.4	16068.8	1086.8	1187.9	80.3	227.8	1300.6	940.0	1.9	63.7	0.8	99.4	30.1	12550.9	2.1	54.1	71.3	309.1	1662.6
2012	Feb	102.3	16068.8	1086.8	1187.9	78.9	228.3	1352.5	940.0	2.0	63.8	0.8	104.0	33.3	12889.0	2.2	55.7	72.0	314.0	1742.0
2012	Mar	106.3	16068.8	1086.8	1187.9	79.6	228.8	1389.2	940.0	2.2	63.8	0.8	109.1	32.0	13079.5	2.3	51.2	68.9	324.7	1676.4
2012	Abr	103.3	16207.1	1130.3	1054.8	79.4	229.2	1386.4	1010.0	2.0	63.7	0.8	107.0	30.6	13030.8	2.2	46.8	68.8	314.2	1650.3
2012	May	94.6	16207.1	1130.3	1054.8	80.9	228.7	1341.3	1010.0	1.8	63.7	0.8	102.0	28.0	12721.1	1.9	43.1	58.6	286.2	1588.4
2012	Jun	82.4	16207.1	1130.3	1054.8	82.2	228.5	1323.5	1010.0	1.6	63.8	0.8	99.6	27.3	12544.9	1.8	46.1	61.5	248.1	1600.6
2012	Jul	87.7	16319.5	1213.4	1210.9	83.1	228.6	1359.8	1010.0	1.5	63.7	0.8	97.0	26.6	12814.1	1.7	42.8	55.4	263.3	1592.5
2012	Ago	94.1	16319.5	1213.4	1210.9	82.2	229.9	1403.4	1010.0	1.7	63.5	0.8	97.6	28.1	13134.9	1.7	45.2	59.6	280.9	1630.3
2012	Set	94.5	16319.5	1213.4	1210.9	79.8	231.0	1443.4	1010.0	1.7	63.6	0.8	100.7	32.8	13418.5	1.7	52.0	74.5	281.0	1748.7
2012	Oct	89.8	16420.4	1163.6	1259.9	79.7	231.6	1436.9	1050.0	1.7	63.8	0.8	102.5	32.2	13370.5	1.8	52.3	72.5	266.4	1746.7
2012	Nov	86.8	16420.4	1163.6	1259.9	80.7	231.2	1395.2	1050.0	1.6	63.6	0.8	104.1	31.8	12900.5	1.8	48.8	73.6	255.3	1722.4
2012	Dic	88.3	16420.4	1163.6	1259.9	79.8	231.2	1422.3	1050.0	1.7	63.7	0.8	110.1	30.7	13144.2	1.8	45.9	66.1	258.2	1684.3
2013	Ene	94.8	16629.1	1062.4	1110.7	79.9	231.7	1480.4	1125.0	1.9	63.7	0.8	118.7	30.1	13615.3	2.1	44.3	62.3	275.6	1670.0
2013	Feb	95.4	16629.1	1062.4	1110.7	80.6	232.9	1512.3	1125.0	2.0	63.4	0.9	124.1	29.2	13967.3	2.2	40.3	52.7	275.5	1627.8
2013	Mar	93.0	16629.1	1062.4	1110.7	82.6	232.3	1550.8	1125.0	1.9	63.3	0.9	123.0	27.9	14418.3	2.0	37.3	47.8	266.9	1593.2
2013	Abr	92.0	16699.6	1041.5	1255.4	82.5	231.8	1570.7	1135.0	1.7	63.4	0.9	127.3	24.4	14675.9	1.8	31.8	38.5	263.1	1485.3
2013	May	94.7	16699.6	1041.5	1255.4	83.2	231.9	1639.8	1135.0	1.9	63.4	0.8	131.1	22.3	15172.2	1.9	28.7	36.7	269.6	1417.4
2013	Jun	95.8	16699.6	1041.5	1255.4	81.9	232.4	1618.8	1135.0	2.3	63.4	0.9	128.4	20.4	15035.8	2.3	27.0	35.2	271.7	1342.2
2013	Jul	104.8	16911.1	1138.5	1112.2	82.8	232.9	1668.7	1040.0	2.6	63.3	0.9	130.5	19.0	15390.2	2.5	25.4	34.8	296.9	1289.9
2013	Ago	106.5	16911.1	1138.5	1112.2	81.5	233.5	1670.1	1040.0	2.7	63.3	0.9	130.3	21.3	15195.3	2.7	27.9	41.9	303.5	1353.5
2013	Set	106.2	16911.1	1138.5	1112.2	81.2	233.5	1687.2	1040.0	2.8	63.2	0.8	132.6	21.7	15269.8	2.8	26.4	37.1	304.5	1348.8
2013	Oct	100.4	17133.1	1118.7	1040.4	79.9	233.7	1720.0	1010.0	2.6	62.8	0.8	133.4	21.2	15289.3	2.7	24.7	32.0	289.9	1315.7
2013	Nov	93.9	17133.1	1118.7	1040.4	80.9	234.1	1783.5	1010.0	2.7	63.0	0.8	134.9	20.0	15870.8	2.8	23.5	31.9	270.8	1276.5
2013	Dic	97.9	17133.1	1118.7	1040.4	80.3	234.7	1807.8	1010.0	2.9	62.9	0.8	141.9	18.9	16095.8	2.9	21.0	28.0	280.6	1222.3

2014	Ene	94.9	17144.3	1107.9	1102.8	80.8	235.3	1822.4	985.0	2.8	62.9	0.8	141.4	19.1	16243.7	2.9	22.7	31.2	271.5	1245.3
2014	Feb	100.8	17144.3	1107.9	1102.8	80.5	235.5	1817.0	985.0	2.7	62.9	0.8	139.6	20.0	15958.4	2.8	25.3	34.0	288.0	1301.1
2014	Mar	100.5	17144.3	1107.9	1102.8	79.9	236.0	1863.5	985.0	2.7	63.1	0.8	141.5	19.9	16308.6	2.8	25.7	32.8	288.7	1335.0
2014	Abr	102.0	17462.7	1053.0	1102.4	79.9	236.5	1864.3	995.0	2.7	62.8	0.8	141.6	19.0	16399.5	2.7	24.2	29.5	294.8	1299.0
2014	May	101.8	17462.7	1053.0	1102.4	79.9	236.9	1889.8	995.0	2.6	62.9	0.8	139.9	18.6	16567.3	2.7	23.5	29.6	297.1	1288.1
2014	Jun	105.2	17462.7	1053.0	1102.4	80.4	237.2	1947.1	995.0	2.6	62.8	0.8	138.8	19.1	16843.8	2.8	24.3	32.7	308.9	1282.7
2014	Jul	102.3	17743.2	1116.5	1108.2	80.5	237.5	1973.1	985.0	2.5	62.9	0.8	137.7	20.1	16988.3	2.7	26.6	37.7	301.7	1311.3
2014	Ago	96.0	17743.2	1116.5	1108.2	81.9	237.5	1961.5	985.0	2.4	62.9	0.8	137.1	19.0	16775.2	2.5	26.6	40.7	284.4	1295.7
2014	Set	93.0	17743.2	1116.5	1108.2	84.4	237.5	1993.2	985.0	2.5	62.8	0.8	138.5	17.7	17098.1	2.5	23.5	36.7	278.4	1235.7
2014	Oct	84.2	17852.5	1122.6	1081.6	85.7	237.4	1937.3	950.0	2.3	62.9	0.8	137.0	16.5	16701.9	2.2	20.6	34.9	254.6	1223.1
2014	Nov	75.7	17852.5	1122.6	1081.6	87.7	237.0	2044.6	950.0	2.3	62.9	0.8	145.1	15.3	17649.0	2.2	18.7	29.9	230.1	1176.1
2014	Dic	59.3	17852.5	1122.6	1081.6	89.2	236.3	2054.3	950.0	2.2	62.8	0.8	147.0	15.6	17754.2	1.9	18.5	31.9	180.2	1199.5
2015	Ene	47.2	17991.3	1125.2	1104.8	93.0	234.7	2028.2	940.0	1.9	62.9	0.8	137.3	16.5	17542.3	1.6	21.3	32.1	142.3	1253.8
2015	Feb	50.7	17991.3	1125.2	1104.8	94.5	235.3	2082.2	940.0	2.0	62.7	0.7	134.9	16.1	17945.4	1.7	21.2	25.8	151.1	1226.9
2015	Mar	47.8	17991.3	1125.2	1104.8	97.9	236.0	2080.0	940.0	2.0	62.6	0.7	130.3	15.6	17931.7	1.8	18.9	24.5	139.3	1179.6
2015	Abr	54.6	18193.7	1091.8	976.1	97.6	236.2	2094.9	935.0	1.9	62.8	0.7	129.2	15.6	17970.5	1.8	19.6	24.5	153.1	1199.8
2015	May	59.4	18193.7	1091.8	976.1	95.2	237.0	2111.9	935.0	2.2	62.9	0.7	134.9	16.1	18124.7	2.0	20.1	24.8	162.1	1198.7
2015	Jun	59.8	18193.7	1091.8	976.1	95.2	237.7	2099.3	935.0	2.4	62.7	0.7	139.0	15.4	17927.2	2.2	18.7	21.6	161.2	1182.4
2015	Jul	50.9	18307.0	1116.3	1168.2	97.0	238.0	2094.1	925.0	2.3	62.6	0.7	135.6	14.4	17795.0	2.1	15.4	17.9	136.1	1129.0
2015	Ago	42.9	18307.0	1116.3	1168.2	96.4	238.0	2039.9	925.0	2.2	62.6	0.7	137.1	14.3	17061.6	2.0	14.2	17.4	113.3	1119.5
2015	Set	45.3	18307.0	1116.3	1168.2	95.8	237.5	1944.4	925.0	2.2	62.4	0.7	135.0	14.1	16340.0	1.9	13.6	14.5	118.5	1125.6
2015	Oct	46.4	18332.1	1095.1	1108.8	95.7	237.7	2024.8	920.0	2.1	62.5	0.7	134.8	15.0	17182.3	1.8	15.7	19.0	119.3	1157.9
2015	Nov	43.0	18332.1	1095.1	1108.8	99.0	238.0	2080.6	920.0	2.3	62.5	0.7	131.7	13.8	17723.8	2.0	13.8	16.1	109.2	1085.6
2015	Dic	37.3	18332.1	1095.1	1108.8	98.3	237.8	2054.1	920.0	2.2	62.7	0.7	132.5	13.4	17542.9	1.9	14.0	15.3	91.3	1068.6
2016	Ene	31.8	18425.3	1196.8	1284.5	99.0	237.7	1918.6	918.0	2.1	62.7	0.8	128.5	13.4	16305.3	1.8	13.7	12.7	74.8	1098.8
2016	Feb	30.7	18425.3	1196.8	1284.5	97.1	237.3	1904.4	918.0	1.8	62.8	0.8	127.3	14.4	16299.9	1.5	17.6	14.0	69.2	1201.2
2016	Mar	38.0	18425.3	1196.8	1284.5	96.3	238.1	2022.0	918.0	1.9	63.0	0.8	125.8	14.7	17302.1	1.5	19.9	16.4	79.6	1245.1
2016	Abr	41.2	18611.6	1188.9	1079.1	94.4	239.0	2075.5	930.0	1.8	62.9	0.8	124.3	15.5	17844.4	1.6	22.2	18.1	82.6	1242.2
2016	May	46.8	18611.6	1188.9	1079.1	94.5	239.6	2065.6	930.0	1.8	62.7	0.8	123.1	16.0	17692.3	1.5	24.0	22.0	91.6	1258.9
2016	Jun	48.8	18611.6	1188.9	1079.1	94.7	240.2	2083.9	930.0	1.6	62.7	0.8	118.6	16.4	17754.9	1.3	25.7	22.0	94.2	1276.2

2016	Jul	44.8	18775.5	1201.1	1032.0	96.6	240.1	2148.9	930.0	1.5	62.8	0.8	115.2	19.0	18341.2	1.0	29.6	23.1	85.2	1339.3
2016	Ago	44.8	18775.5	1201.1	1032.0	95.4	240.5	2177.5	930.0	1.6	62.9	0.9	113.5	18.5	18495.2	0.7	29.6	19.8	83.8	1338.5
2016	Set	45.2	18775.5	1201.1	1032.0	95.5	241.2	2157.7	930.0	1.6	62.9	0.9	114.1	18.3	18267.4	0.8	26.9	19.2	83.1	1326.6
2016	Oct	49.9	18968.0	1160.6	955.5	97.7	241.7	2143.0	925.0	1.8	62.8	0.9	114.5	16.8	18184.5	1.1	23.8	17.6	90.6	1265.9
2016	Nov	45.8	18968.0	1160.6	955.5	99.7	242.0	2165.0	925.0	2.1	62.7	0.9	117.0	16.5	18689.8	1.4	22.5	14.6	82.5	1237.0
2016	Dic	52.1	18968.0	1160.6	955.5	102.0	242.6	2246.6	925.0	2.5	62.7	0.8	122.4	15.5	19712.4	1.4	20.3	14.9	92.0	1151.2
2017	Ene	52.6	19148.2	1103.8	1133.4	101.1	243.6	2275.1	930.0	2.4	62.8	0.9	122.2	16.0	19908.2	1.4	22.9	17.1	91.1	1193.9
2017	Feb	53.4	19148.2	1103.8	1133.4	100.7	244.0	2329.9	930.0	2.4	62.9	0.9	120.2	17.0	20424.1	1.3	24.6	18.0	91.3	1235.0
2017	Mar	49.7	19148.2	1103.8	1133.4	100.8	243.9	2366.8	930.0	2.5	62.9	0.9	120.7	16.7	20823.1	1.2	22.4	15.4	84.1	1231.3
2017	Abr	51.1	19304.5	1154.4	1035.7	100.0	244.2	2359.3	930.0	2.3	63.0	0.8	118.1	17.1	20684.7	1.1	23.4	17.7	85.6	1270.3
2017	May	48.6	19304.5	1154.4	1035.7	98.3	244.0	2395.3	930.0	2.3	62.8	0.9	124.1	15.9	20936.8	1.1	22.4	17.2	80.6	1244.7
2017	Jun	45.1	19304.5	1154.4	1035.7	96.9	244.2	2434.0	930.0	2.2	62.8	0.9	124.7	16.0	21317.8	1.1	22.5	14.2	74.6	1260.0
2017	Jul	46.8	19561.9	1228.4	985.5	94.9	244.2	2454.1	941.0	2.3	62.9	0.9	129.6	15.3	21581.3	1.3	22.0	12.2	76.6	1238.2
2017	Ago	48.0	19561.9	1228.4	985.5	93.2	245.2	2456.2	941.0	2.2	62.9	0.9	129.8	16.0	21914.1	1.1	23.1	9.7	78.6	1284.3
2017	Set	49.7	19561.9	1228.4	985.5	92.3	246.4	2492.8	941.0	2.2	63.1	0.9	132.0	16.4	22173.4	1.2	24.1	11.0	81.2	1314.4
2017	Oct	51.5	19894.8	1201.6	1129.3	93.7	246.6	2557.0	938.0	2.4	62.7	0.9	132.8	16.0	23036.2	1.4	23.2	9.8	83.2	1280.9
2017	Nov	56.5	19894.8	1201.6	1129.3	94.0	247.3	2593.6	938.0	2.4	62.7	0.9	132.5	16.0	23557.9	1.3	22.7	6.1	90.9	1281.7
2017	Dic	57.9	19894.8	1201.6	1129.3	93.4	247.8	2664.3	938.0	2.4	62.7	0.9	133.6	15.3	24545.4	1.3	22.3	6.3	92.7	1266.5
2018	Ene	63.7	20155.5	1116.9	987.7	90.7	248.9	2789.8	945.0	2.6	62.7	0.9	135.3	16.2	25804.0	1.4	23.8	6.7	101.9	1331.7
2018	Feb	62.1	20155.5	1116.9	987.7	89.7	249.5	2705.2	945.0	2.9	63.0	0.9	133.2	15.6	24981.5	1.6	22.2	5.7	99.6	1331.0
2018	Mar	62.8	20155.5	1116.9	987.7	89.9	249.6	2702.8	945.0	2.8	62.9	0.9	130.9	15.5	24582.2	1.5	21.7	5.1	101.3	1326.4
2018	Abr	66.3	20470.2	1179.4	1047.1	90.3	250.2	2653.6	947.0	2.9	62.9	0.9	132.1	15.7	24304.2	1.5	22.5	4.8	107.1	1335.0
2018	May	70.0	20470.2	1179.4	1047.1	93.3	250.8	2701.5	947.0	3.0	62.9	0.9	129.7	15.5	24572.5	1.5	22.5	5.1	113.1	1303.5
2018	Jun	67.3	20470.2	1179.4	1047.1	94.4	251.0	2754.4	947.0	2.9	63.0	0.9	128.6	15.5	24790.1	1.4	22.2	5.5	108.9	1280.4
2018	Jul	71.0	20687.3	1264.5	1113.2	94.6	251.2	2793.6	935.0	2.9	63.0	0.9	130.3	14.8	24978.2	1.4	21.9	5.4	115.6	1236.9
2018	Ago	68.3	20687.3	1264.5	1113.2	95.5	251.7	2857.8	935.0	2.9	62.6	0.9	128.2	14.1	25630.0	1.4	19.6	5.0	113.3	1200.3
2018	Set	70.4	20687.3	1264.5	1113.2	94.7	252.2	2901.5	935.0	3.0	62.8	0.9	130.7	13.4	26232.7	1.5	18.3	4.4	118.1	1197.9
2018	Oct	70.8	20819.3	1226.8	1297.3	95.9	252.8	2785.5	930.0	3.2	62.9	0.9	129.5	13.7	25609.3	1.5	19.3	4.3	119.7	1215.3
2018	Nov	56.6	20819.3	1226.8	1297.3	96.8	252.6	2723.2	930.0	3.1	62.9	0.9	128.8	13.5	25258.7	1.5	19.2	3.2	96.0	1220.5
2018	Dic	48.7	20819.3	1226.8	1297.3	96.9	252.8	2567.3	930.0	2.8	63.0	0.9	127.7	13.9	23805.5	1.3	20.4	3.0	82.8	1252.6

2019	Ene	51.4	21013.1	1127.3	1070.7	95.9	252.7	2607.4	935.0	2.7	63.1	0.9	124.4	14.7	24157.8	1.3	21.1	3.0	87.0	1293.4
2019	Feb	54.9	21013.1	1127.3	1070.7	96.5	253.3	2754.9	935.0	2.7	63.1	0.9	125.4	14.8	25605.5	1.2	22.5	4.1	92.3	1318.7
2019	Mar	58.0	21013.1	1127.3	1070.7	96.8	254.2	2804.0	935.0	2.6	63.0	0.9	125.6	14.3	25722.6	1.2	22.4	4.6	97.0	1300.8
2019	Abr	63.7	21272.4	1184.6	1158.5	97.4	255.2	2903.8	955.0	2.5	62.9	0.9	125.4	14.1	26401.6	1.2	21.8	4.4	106.3	1286.2
2019	May	60.8	21272.4	1184.6	1158.5	97.7	255.3	2854.7	955.0	2.4	62.9	0.9	123.0	13.7	25744.8	1.1	20.6	3.6	101.3	1284.0
2019	Jun	54.7	21272.4	1184.6	1158.5	96.8	255.2	2890.2	955.0	2.1	63.0	0.9	122.0	14.1	26160.1	0.9	23.9	4.7	91.1	1361.7
2019	Jul	57.5	21531.8	1289.3	1093.6	97.4	255.7	2996.1	960.0	2.1	63.1	0.9	121.3	14.8	27089.2	0.8	26.7	7.2	95.6	1415.7
2019	Ago	54.9	21531.8	1289.3	1093.6	98.0	256.1	2897.5	960.0	1.6	63.1	0.9	118.1	16.1	26058.2	0.6	29.0	8.7	91.2	1503.5
2019	Set	57.0	21531.8	1289.3	1093.6	98.6	256.5	2982.2	960.0	1.7	63.2	0.9	118.4	16.9	26900.2	0.6	28.3	8.6	94.9	1507.1
2019	Oct	53.9	21706.5	1270.9	1031.8	98.2	257.2	2977.7	950.0	1.7	63.3	0.9	119.6	16.5	26736.8	0.6	27.3	7.8	90.2	1495.0
2019	Nov	57.0	21706.5	1270.9	1031.8	98.1	257.8	3104.9	950.0	1.8	63.3	0.9	120.3	16.0	27797.0	0.8	26.9	8.0	95.4	1470.8
2019	Dic	59.8	21706.5	1270.9	1031.8	97.4	258.6	3176.7	950.0	1.9	63.3	0.8	121.2	16.0	28167.0	0.8	27.8	7.5	100.0	1482.0
2020	Ene	57.7	21538.0	1141.9	1097.1	97.4	259.0	3278.2	980.0	1.8	63.3	0.8	121.3	16.8	28880.0	0.7	28.7	7.5	96.7	1561.5
2020	Feb	50.6	21538.0	1141.9	1097.1	98.9	259.2	3277.3	980.0	1.5	63.3	0.8	120.0	16.7	28519.7	0.6	28.8	8.0	84.9	1596.3
2020	Mar	30.8	21538.0	1141.9	1097.1	98.8	258.1	2652.4	980.0	0.9	62.6	0.9	119.1	13.9	22637.4	0.5	24.8	7.2	51.5	1591.9
2020	Abr	16.8	19636.7	1074.8	921.8	99.9	256.1	2762.0	970.0	0.7	60.1	0.9	117.1	14.1	23293.9	0.4	30.0	8.3	31.3	1686.3
2020	May	10.4	19636.7	1074.8	921.8	99.6	255.9	2919.6	970.0	0.7	60.8	0.9	116.9	15.2	24271.0	0.3	34.8	9.0	22.9	1718.2
2020	Jun	14.9	19636.7	1074.8	921.8	97.1	257.0	3104.7	970.0	0.7	61.5	0.9	121.1	16.5	26062.3	0.3	33.9	8.7	27.7	1735.9
2020	Jul	19.2	21362.4	1301.6	887.1	95.5	258.3	3207.6	970.0	0.6	61.5	0.9	122.5	19.3	26385.8	0.2	39.8	11.3	29.1	1849.9
2020	Ago	24.1	21362.4	1301.6	887.1	93.0	259.4	3391.7	970.0	0.6	61.7	0.9	125.4	25.1	27821.4	0.3	41.8	11.5	30.4	1970.1
2020	Set	32.6	21362.4	1301.6	887.1	93.4	260.0	3365.5	970.0	0.7	61.4	0.9	124.4	24.0	27733.4	0.3	40.7	10.8	28.3	1922.9
2020	Oct	39.7	21704.7	1256.9	773.2	93.4	260.3	3418.7	1010.0	0.8	61.7	0.9	123.8	22.5	28005.1	0.3	39.1	12.5	27.9	1899.8
2020	Nov	41.5	21704.7	1256.9	773.2	92.6	260.8	3549.0	1010.0	0.9	61.5	0.9	123.5	22.4	29124.0	0.4	37.0	13.0	28.8	1868.2
2020	Dic	47.1	21704.7	1256.9	773.2	90.5	262.0	3695.3	1010.0	0.9	61.5	0.9	126.3	23.3	30148.6	0.3	35.8	13.0	32.2	1859.4
2021	Ene	52.0	22313.9	1105.1	914.4	90.2	262.7	3793.7	1050.0	1.1	61.3	0.9	126.3	24.1	30821.4	0.4	35.9	12.1	35.2	1866.1
2021	Feb	59.0	22313.9	1105.1	914.4	90.6	263.6	3883.4	1050.0	1.2	61.4	0.9	127.4	25.3	31283.9	0.6	34.0	12.3	39.6	1807.5
2021	Mar	62.3	22313.9	1105.1	914.4	92.0	264.9	3910.5	1050.0	1.6	61.5	0.9	129.3	23.8	32373.3	0.8	32.6	11.2	42.2	1721.4
2021	Abr	61.7	23046.9	1156.8	1035.2	91.6	266.7	4141.2	1070.0	1.6	61.6	0.9	130.5	23.8	33803.3	0.9	35.1	11.1	42.2	1760.5
2021	May	65.2	23046.9	1156.8	1035.2	90.3	268.4	4167.8	1070.0	1.6	61.5	0.9	132.5	25.5	34270.3	0.9	38.1	11.3	44.6	1849.8
2021	Jun	71.4	23046.9	1156.8	1035.2	91.0	270.6	4238.5	1070.0	1.5	61.7	0.9	132.6	25.0	34289.9	0.9	36.5	11.0	48.5	1831.9

2021	Jul	72.6	23550.4	1225.6	920.9	92.5	271.8	4363.7	1125.0	1.3	61.8	0.9	130.3	23.8	34798.8	0.7	34.1	9.6	49.5	1806.3
2021	Ago	67.8	23550.4	1225.6	920.9	92.8	272.9	4454.2	1125.0	1.3	61.7	0.9	129.2	22.2	35244.0	0.6	32.6	8.6	47.2	1786.2
2021	Set	71.6	23550.4	1225.6	920.9	93.0	274.0	4445.5	1125.0	1.4	61.7	0.9	129.5	21.5	34688.4	0.8	31.1	8.3	50.1	1775.1
2021	Oct	81.4	24349.1	1238.3	1144.1	93.9	276.5	4460.7	1120.0	1.6	61.8	0.8	131.2	21.6	35055.5	1.1	31.8	9.1	56.5	1776.3
2021	Nov	79.0	24349.1	1238.3	1144.1	95.3	278.7	4667.5	1120.0	1.6	61.9	0.8	130.1	22.4	35848.6	0.9	33.1	9.6	55.1	1819.3
2021	Dic	72.1	24349.1	1238.3	1144.1	96.2	280.9	4674.8	1120.0	1.5	62.0	0.8	128.8	20.8	35641.3	0.8	31.0	9.0	51.7	1791.8
2022	Ene	83.2	24740.5	1139.3	1238.5	96.0	282.6	4573.8	1245.0	1.8	62.2	0.8	129.9	21.4	35456.1	1.2	31.0	9.2	59.2	1815.8
2022	Feb	92.3	24740.5	1139.3	1238.5	96.0	284.6	4436.0	1245.0	1.9	62.2	0.8	130.7	21.7	34648.5	1.5	32.6	10.1	64.7	1856.6
2022	Mar	109.7	24740.5	1139.3	1238.5	98.5	287.5	4391.3	1245.0	2.1	62.4	0.8	130.7	23.4	34029.7	1.5	37.7	11.3	76.4	1951.5
2022	Abr	102.4	25248.5	1179.7	943.2	100.7	288.6	4391.3	1252.0	2.7	62.2	0.8	136.5	22.6	34315.0	1.8	38.3	11.3	76.2	1935.1
2022	May	109.7	25248.5	1179.7	943.2	103.2	291.3	4040.6	1252.0	2.9	62.3	0.8	136.2	20.2	32380.0	1.9	32.5	8.5	80.7	1847.7
2022	Jun	115.4	25248.5	1179.7	943.2	103.8	294.7	3898.9	1252.0	3.1	62.2	0.9	141.6	19.8	31446.9	2.4	30.6	7.6	86.0	1836.7
2022	Jul	100.4	25723.9	1224.9	1221.8	107.0	294.6	3911.7	1262.0	2.9	62.1	0.8	139.0	17.6	31535.3	2.1	26.1	5.8	76.1	1732.8
2022	Ago	92.2	25723.9	1224.9	1221.8	107.1	295.3	4158.6	1262.0	2.9	62.3	0.8	137.0	18.1	33009.9	2.3	25.9	6.1	74.5	1763.1
2022	Set	84.2	25723.9	1224.9	1221.8	110.7	296.5	3850.5	1262.0	3.5	62.3	0.9	141.8	17.4	30650.4	3.5	23.8	5.7	68.9	1679.3
2022	Oct	87.5	26138.0	1246.2	1302.8	111.9	298.0	3726.1	1258.0	4.0	62.2	0.9	144.8	17.9	30571.8	4.1	24.2	6.2	71.4	1665.6
2022	Nov	84.7	26138.0	1246.2	1302.8	108.2	298.6	3913.7	1258.0	3.9	62.2	0.9	145.1	19.3	33401.5	3.4	26.8	6.8	71.4	1725.9
2022	Dic	77.0	26138.0	1246.2	1302.8	104.5	299.0	3912.4	1258.0	3.6	62.3	0.9	142.9	21.5	33482.2	3.5	29.0	8.3	66.9	1797.5