

UNIVERSIDAD SAN PABLO DE GUATEMALA
Facultad de Ciencias Empresariales
Escuela de Ingeniería en Ciencias y Sistemas de la Computación.



[Retail Sales Forecasting](#)

en el curso de Sistemas Expertos

Impartido por el ingeniero Marco Alfredo Orozco de Paz

Henry Morales
2200304

Contenido

INTRODUCCIÓN.....	3
BASE MATEMÁTICA	3
REGRESIÓN LINEAL EN MACHINE LEARNING	4
1. Relación Lineal (Linealidad).....	4
2. Independencia Residual (Ausencia de Autocorrelación)	4
3. Normalidad de los Residuos (Distribución Normal)	4
4. Homocedasticidad (Varianza Constante de los Residuos).....	5
IMPLEMENTACIÓN	5
PARTE 1: CARGA DE LIBRERÍAS Y DATASET	5
PARTE 2: PREPROCESAMIENTO Y CREACIÓN VARIABLES.....	6
PARTE 3: SEPARACION DE DATOS Y MODELADO	7
PARTE 4: PREDICCIONES Y EVALUACIÓN	8
PARTE 5: PRUEBAS Y PREDICCIONES CASOS ESPECÍFICOS.....	11
PARTE 6: ANÁLISIS DE RESIDUOS Y PROPUESTAS DE MODELOS.....	12
PARTE 7: CONCLUSIONES Y VERIFICACION DE LAS PREDICCIONES	13
PARTE 8: MODELO ALTERNATIVO XGBOOST	13
PARTE 9: PRUEBA CON CASOS ESPECÍFICOS XGBoost.....	14
PARTE 10: CONCLUSIONES REGRESIÓN LINEAL VRS XGBOOST	15
PARTE 11: DESPLIEGUE Y VISUALIZACIÓN WEB DEL MODELO	16

INTRODUCCIÓN

El presente laboratorio consisten en la carga del dataset mock_kaggle.csv que contiene datos “ Retail Sales Forecasting” de Kaggle, este archivo debe cargarse al entorno de Google Colab , hacer la exploración del dataset , identificar la necesidad de limpieza (sí aplica), generar la transformación (nuevas variables), realizar el entrenamiento de un modelo predictivo de Regresión Lineal y finalmente concluir sí el modelo de Regresión Lineal es optimo para el tipo de información de Retail Sales Forecasting.

BASE MATEMÁTICA

De acuerdo con AWS Amazon (<https://aws.amazon.com/es/what-is/linear-regression/>) la regresión lineal es una técnica de análisis de datos con el cual se busca predecir valores de datos desconocidos a través del uso de otro(s) valor(es) de datos relacionados y conocidos. Se modela matemáticamente la variable a predecir (desconocida) y la variable conocida a través de una ecuación lineal.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Donde:

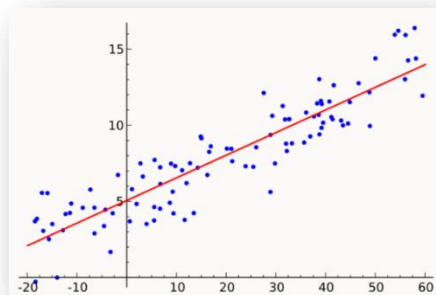
Y: Variable dependiente (desconocida).

X: Variable independiente (predictora, conocida).

β_0 : Intercepto (punto de corte con el eje Y).

β_1 : Coeficiente de pendiente (es el cambio de Y por cada unidad de cambio en X)

ϵ : Término de error (residuo, representa la parte de Y que no es explicada por X)



Una aplicación con datos de gastos e ingresos del año pasado, con regresión lineal se analiza los datos y se determina que los gastos corresponden a la mitad de los ingresos. Luego con este análisis y correlación se calcula un gasto futuro desconocido al reducir la mitad de un ingreso conocido futuro.

REGRESIÓN LINEAL EN MACHINE LEARNING

Los algoritmos analizan grandes conjuntos de datos y trabajan a partir de estos datos para calcular la ecuación de regresión lineal. Se entrena un algoritmo con un conjunto de datos conocidos o etiquetados y a continuación se utiliza el algoritmo para predecir valores desconocidos. En aplicaciones reales el análisis es mucho más complejo y requiere modificar y transformar matemáticamente los valores de los datos para cumplir con cuatro supuestos:

1. Relación Lineal (Linealidad)

Este supuesto es la base de la regresión.

- ¿Qué Implica? Debe existir una relación directa que se pueda modelar con una línea recta entre las variables independientes (x) y la variable dependiente (y). Si la relación es curva (ej. exponencial o cuadrática), el modelo Lineal la modelará pobremente.
- Verificación: Se utiliza un Gráfico de Dispersión (Scatter Plot) para observar la forma de la nube de puntos. También, se revisa el mapa de calor de correlaciones para asegurar que las variables tienen una correlación de Pearson significativa.

2. Independencia Residual (Ausencia de Autocorrelación)

Este supuesto se refiere a que los errores de las predicciones no deben estar relacionados entre sí.

- ¿Qué Implica? El error cometido por el modelo en la predicción de hoy no debe influir ni estar correlacionado con el error que cometerá en la predicción de mañana. Cada residuo debe ser un evento aleatorio e independiente.
- Verificación: Se utiliza una gráfica de Residuos en Secuencia (especialmente en series de tiempo). No debe existir un patrón identificable (ej. si cinco residuos positivos seguidos indican que el sexto también será positivo, hay autocorrelación).

3. Normalidad de los Residuos (Distribución Normal)

Este supuesto asegura que los errores son aleatorios y no sesgados.

- ¿Qué Implica? Los errores de predicción deben seguir una Distribución Normal (forma de campana) con media cero. Esto significa que los errores grandes son raros, y la mayoría de los errores son pequeños y cercanos a cero.
- Verificación: Histograma de Residuos: Debe tener forma de campana y estar centrado en cero.

4. Homocedasticidad (Varianza Constante de los Residuos)

Este es el supuesto más violado en la implementación del dataset Retail Sales Forecasting:

- ¿Qué Implica? La variación o dispersión de los residuos debe ser constante para todos los valores de la variable X. En otras palabras, la precisión del modelo (la magnitud del error) no debe depender de si la venta predicha es alta o baja.
- Verificación: Se utiliza la gráfica de Residuos vs. Predicciones.
 - Resultado Ideal (Homocedasticidad): Los puntos forman una nube de puntos aleatoria y uniforme centrado en $y=0$.
 - Resultado NO Ideal (Heterocedasticidad): Los puntos forman un patrón, típicamente una forma de embudo donde el error aumenta (o disminuye) a medida que aumentan las predicciones.

IMPLEMENTACIÓN

PARTE 1: CARGA DE LIBRERÍAS Y DATASET

En la parte 1 se cargaron las librerías que serán utilizadas en la implementación:

- pandas: Librería fundamental para la manipulación y análisis de datos (DataFrames).
- numpy: Para manejo de operaciones numéricas y arrays de alto rendimiento.
- matplotlib.pyplot: Para la creación de gráficos y visualizaciones estáticas.
- seaborn: Librería basada en matplotlib para crear gráficos estadísticos más atractivos e informativos.
- train_test_split: Función para dividir el dataset en conjuntos de entrenamiento y prueba.
- LinearRegression: El modelo de regresión lineal a implementar.
- Métricas de evaluación: Para medir el rendimiento del modelo (Error Absoluto Medio, Error Cuadrático Medio, Coeficiente R2).
- os: Para interactuar con el sistema operativo, útil en entornos como Colab para verificar archivos.

En la exploración inicial se constató que el dataset consta de 4 campos:

- data : fecha de venta tipo object.
- Venda: Ventas tipo int64.
- Estoque: Stock tipo int64
- Preco: Precio unitario tipo float64.

No se identificaron datos nulos

```
--- INICIO DE CARGA DEL DATASET ---
Dataset 'mock_kaggle.csv' cargado exitosamente.

--- Primeras 5 filas del Dataframe (df.head()) ---
   data  venda  estoque  preco
0  2014-01-01    0    4972    1.29
1  2014-01-02   70    4902    1.29
2  2014-01-03   59    4843    1.29
3  2014-01-04   93    4750    1.29
4  2014-01-05   96    4654    1.29

--- Información general del Dataframe (df.info()) ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 937 entries, 0 to 936
Data columns (total 4 columns):
#   column  Non-Null Count  Dtype
---  -
0   data    937 non-null    object
1   venda   937 non-null    int64
2   estoque 937 non-null    int64
3   preco   937 non-null    float64
dtypes: float64(1), int64(2), object(1)
memory usage: 29.4+ KB

--- FIN DE LA PRIMERA ETAPA: CARGA E INSPECCIÓN BÁSICA ---
```

PARTE 2: PREPROCESAMIENTO Y CREACIÓN VARIABLES

Esta fase incluyó conversión del campo “data” (fecha de venta) de tipo *object* a *datetime*. También se generaron variables (componentes de fecha) a partir de “data”: Adicionalmente se crea una variable categoría sobre cada día de la semana con valores de 0 a 6:

```
# a) Extracción de Componentes de la Fecha
# Crear la variable 'Ano' (Año)
df['Ano'] = df['data'].dt.year
# Crear la variable 'Mes' (Mes)
df['Mes'] = df['data'].dt.month
# Crear la variable 'Dia_Semana' (Día de la Semana): 0 = Lunes, 6 = Domingo
df['Dia_Semana'] = df['data'].dt.dayofweek
# Crear la variable 'Dia_Ano' (Día del Año): Representa la posición dentro del ciclo anual (1 a 366).
df['Dia_Ano'] = df['data'].dt.dayofyear
```

```
# b) Creación de Variable Categórica (Día Hábil/Feriado o Fin de Semana)
# Asumimos que 5 y 6 (Sábado y Domingo) son Fin de Semana, y el resto (0 a 4) son Días Hábiles.
df['Es_Fin_Semana'] = df['Dia_Semana'].apply(lambda x: 1 if x >= 5 else 0)
print(" Variables de tiempo (Ano, Mes, Dia_Semana, Es_Fin_Semana) creadas.")
```

```
--- Vista Previa del DataFrame con Nuevas Variables ---
   data  venda  estoque  preco  Ano  Mes  Dia_Semana  Dia_Ano
0 2014-01-01    0    4972   1.29  2014    1         2         1
1 2014-01-02    70    4902   1.29  2014    1         3         2
2 2014-01-03    59    4843   1.29  2014    1         4         3
3 2014-01-04    93    4750   1.29  2014    1         5         4
4 2014-01-05    96    4654   1.29  2014    1         6         5

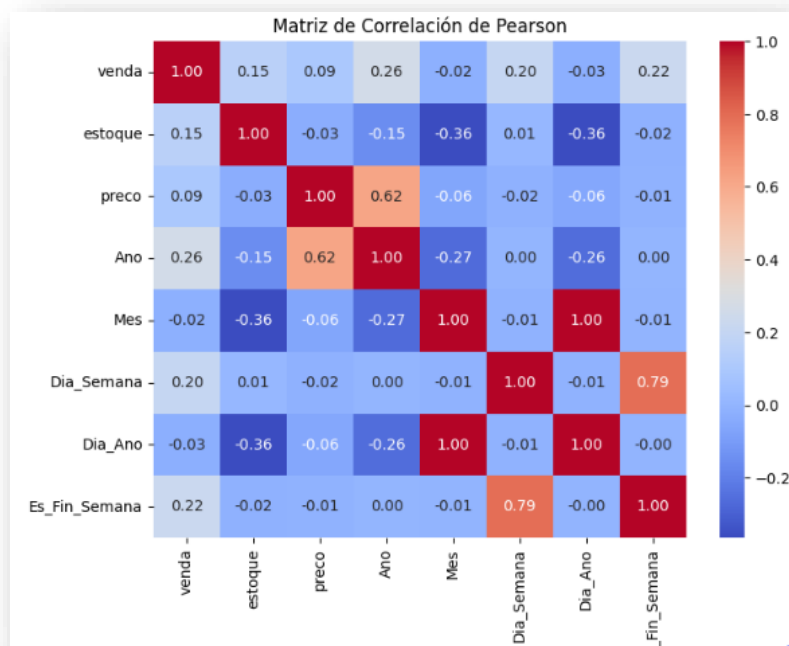
   Es_Fin_Semana
0              0
1              0
2              0
3              1
4              1
```

Una vez generada las variables, se determina la correlación de Pearson entre variables numéricas y la variable objetivo “ventas”:

```
--- Correlación con la variable 'venda' (Target) ---
venda          1.000000
Ano            0.264477
Es_Fin_Semana  0.224862
Dia_Semana     0.201810
estoque        0.153659
preco          0.094779
Mes           -0.020992
Dia_Ano        -0.025433
Name: venda, dtype: float64
```

La tabla de correlación de Pearson muestra la fuerza y dirección de la relación lineal entre cada variable y las Ventas (venta):

Variable	Correlación	Dirección	Interpretación para el Modelo
venta	1	—	es la variable <i>target</i> , es normal correlacion 1
Año	0.26	Positiva	Las ventas tienden a aumentar ligeramente con el tiempo (cada año).
Es_Fin_Semana	0.22	Positiva	Las ventas son significativamente más altas durante los fines de semana (cuando Es_Fin_Semana es 1).
Dia_Semana	0.2	Positiva	Relacionada con la anterior. A medida que avanza la semana (de 0 a 6), las ventas aumentan (sugiriendo picos el fin de semana).
estoque	0.15	Positiva	Relación Confusa/Débil: A mayor inventario, ligeramente mayor venta. Esto puede indicar que cuando hay más <i>stock</i> , las ventas aún están siendo altas, o viceversa, lo que requiere más análisis.
preco	0.09	Positiva	Relación Muy Débil: Un precio ligeramente más alto se asocia con ventas ligeramente más altas.
Mes	-0.02	Negativa	Correlación Nula: El mes del año, por sí solo, no tiene una relación lineal fuerte con las ventas.
Dia_Año	-0.03	Negativa	Correlación Nula: El día del año tampoco tiene una relación lineal fuerte.



Las variables temporales (Año, Fines de Semana) parecen ser las más importantes. Las variables operacionales (estoque, preco) tienen una correlación más débil. Esto indica que el modelo de regresión lineal deberá confiar fuertemente en los factores de tiempo.

PARTE 3: SEPARACION DE DATOS Y MODELADO

Esta sección implicó las siguientes fases:

- 1) Selección de Características (X): Selección de las variables predictoras (todas las creadas).
- 2) Selección del Target (Y): La variable a predecir (venta).
- 3) Separación de Datos: Dividir el *dataset* en conjuntos de entrenamiento (80%) y prueba (20%).

4) Entrenamiento del Modelo: Creación y entrenamiento del modelo de Regresión Lineal.

```
--- INICIO DE ENTRENAMIENTO DEL MODELO ---
Características (X) seleccionadas: ['estoque', 'preco', 'Ano', 'Mes', 'Dia_Semana', 'Es_Fin_Semana']
Datos separados: Entrenamiento (749 filas), Prueba (188 filas).

Modelo de Regresión Lineal entrenado exitosamente.

--- Coeficientes del Modelo (Importancia de cada Feature) ---
                Coeficiente
Ano              45.135155
Es_Fin_Semana    22.578000
Mes              4.307854
Dia_Semana       2.272728
estoque          0.017712
preco           -24.302401

--- FIN DE LA ETAPA DE ENTRENAMIENTO ---
```

Los coeficientes dan una visión preliminar del modelo:

- Ano (45.14): Por cada año que pasa, las ventas predichas aumentan en aproximadamente 45 unidades (suponiendo que las otras variables se mantienen constantes). Es el factor más fuerte.
- Es_Fin_Semana (22.58): Estar en un fin de semana aumenta las ventas predichas en unas 23 unidades.
- preco (-24.30): El precio tiene un coeficiente negativo. Por cada unidad de aumento en el precio, las ventas predichas disminuyen en 24.30 unidades (lo cual es lógico en economía: a mayor precio, menor demanda, aunque su correlación inicial fue baja).

PARTE 4: PREDICCIONES Y EVALUACIÓN

Este bloque realiza las predicciones en el conjunto de prueba (X_{test}) y luego calcula las métricas de rendimiento (MAE, RMSE, R^2) :

1. Error Absoluto Medio (MAE)

El MAE es la medida más intuitiva del error del modelo.

- ¿Qué Mide?: El promedio de la magnitud de los errores cometidos por el modelo en las predicciones, sin importar la dirección (si el modelo predijo de más o de menos).
- Interpretación: Se expresa en las mismas unidades que la variable objetivo (unidades de venta).

2. Raíz del Error Cuadrático Medio (RMSE)

El RMSE es la métrica más utilizada en muchos campos, ya que penaliza más severamente los errores grandes.

- ¿Qué Mide?: La desviación estándar de los residuos. Primero eleva al cuadrado los errores (para eliminar signos negativos y magnificar errores grandes), luego calcula el promedio, y finalmente toma la raíz cuadrada para devolver la métrica a las unidades originales.
- Interpretación: Al igual que el MAE, se expresa en unidades de venta. Es siempre igual o mayor que el MAE. Una gran diferencia entre el RMSE y el MAE indica que existen muchos *outliers* o errores muy grandes.

3. Coeficiente de Determinación R^2

R^2 es la métrica principal para evaluar la capacidad explicativa del modelo.

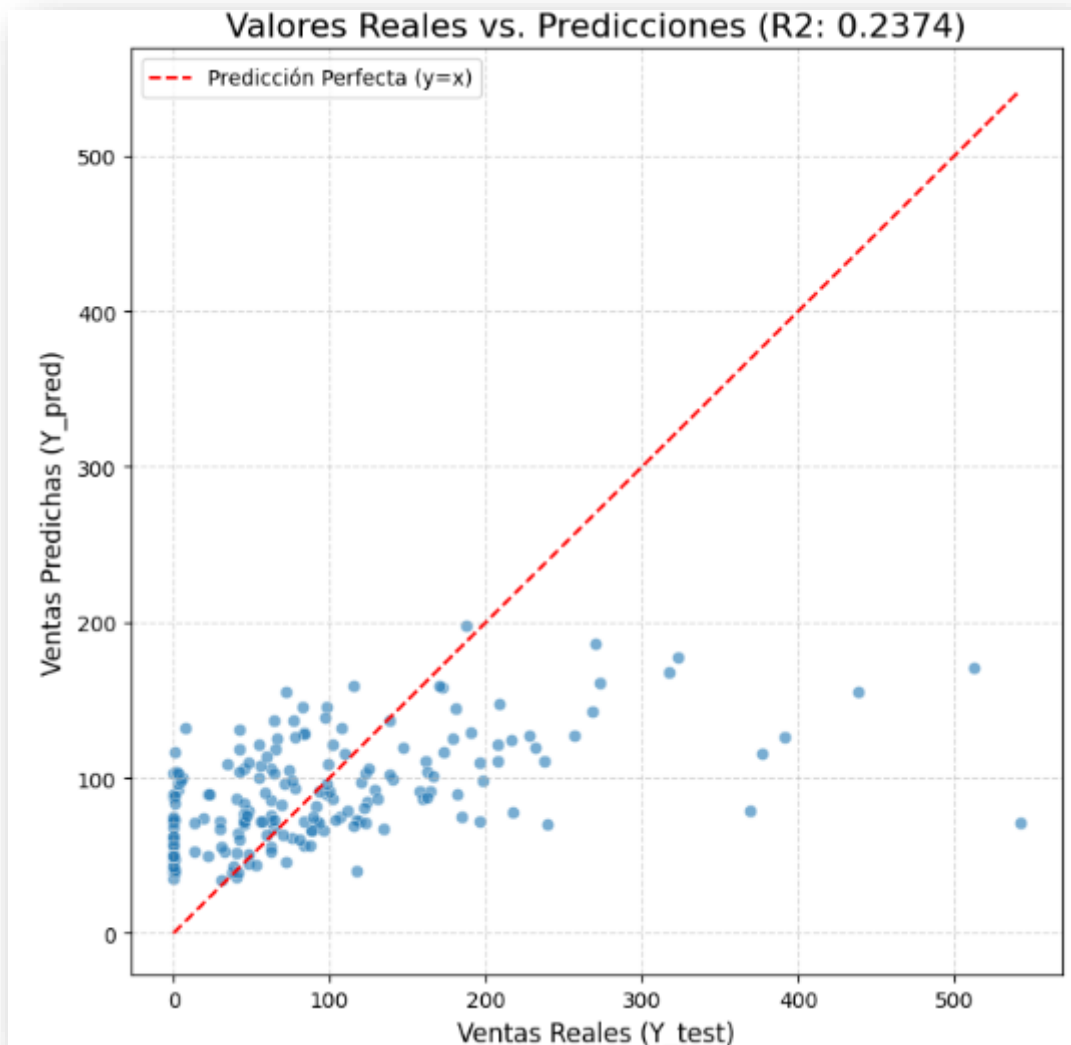
- ¿Qué Mide?: El porcentaje de la variabilidad de la variable objetivo (Y) que es explicado por las variables predictoras (X) incluidas en el modelo.
- Interpretación: El valor varía típicamente entre 0 y 1 (o 0% a 100%).
 - R^2 de 1 (100%): El modelo explica toda la variabilidad; es un ajuste perfecto.
 - R^2 de 0: El modelo no explica nada, es tan bueno como tomar el promedio de Y.

```

--- INICIO DE PREDICCIONES Y EVALUACIÓN ---
Predicciones generadas en el conjunto de prueba (y_pred).

--- Métricas de Desempeño del Modelo ---
R-cuadrado (R2): 0.2374 (Porcentaje de varianza de Y explicado por X)
Error Absoluto Medio (MAE): 58.08 (Error promedio en unidades de venta)
Raíz del Error Cuadrático Medio (RMSE): 83.39 (Error estandarizado en unidades de venta)

```



Coefficiente R-cuadrado (R^2): 0.2374

- Interpretación: Esto significa que las variables predictoras (estoque, precio, Año, Mes, Día_Semana, Es_Fin_Semana) logran explicar solo el 23.74% de la variabilidad total en las ventas (venta).
- Conclusión: Un valor de R^2 bajo (cercano a cero) para un modelo de predicción de ventas indica que la Regresión Lineal no es el modelo más adecuado para este conjunto de datos, o que hay factores muy importantes que no se han incluido (ej. promociones, días festivos específicos, competencia, etc.).

Error Absoluto Medio (MAE): 58.08

- Interpretación: En promedio, la predicción del modelo se equivoca en 58.08 unidades de venta respecto al valor real. Si las ventas promedio son altas, este error puede ser aceptable; si las ventas promedio son bajas, el error es significativo. (Basado en la inspección inicial, las ventas están en el rango de 0 a 400, por lo que 58.08 es un error considerable).

Gráfica (Predicciones vs. Reales)

- Interpretación de la Gráfica:
 - El eje X muestra las Ventas Reales (y_{test}).
 - El eje Y muestra las Ventas Predichas (y_{pred}).
 - La línea diagonal roja punteada es la línea de predicción perfecta ($y=x$).
 - Si los puntos se agrupan firmemente alrededor de la línea roja, el modelo es excelente.
 - Lo que se observa: Los puntos están muy dispersos y forman una nube que no está fuertemente alineada con la diagonal. Esto confirma el R^2 bajo y el alto MAE: el modelo no logra predecir consistentemente los valores reales.

PARTE 5: PRUEBAS Y PREDICCIONES CASOS ESPECÍFICOS

Realizar pruebas enviando ejemplos de datos al modelo para obtener predicciones específicas. Esto permite simular un escenario de uso real.

Se crear un *DataFrame* con cinco casos de prueba que cubran diferentes escenarios (diferentes precios, stock y días de la semana/fin de semana).

```
--- INICIO DE PRUEBAS CON DATOS NUEVOS ---
--- Datos de Prueba Generados ---
```

	estoque	preco	Ano	Mes	Dia_Semana
Caso 1 (Lunes Alto Stock/Bajo Precio)	4500	1.50	2017	1	0
Caso 2 (Sábado Bajo Stock/Alto Precio)	100	3.50	2016	12	5
Caso 3 (Jueves Stock/Precio Medio)	2500	2.00	2016	6	3
Caso 4 (Venta Baja Enero 2014)	4972	1.29	2014	1	2
Caso 5 (Lunes 2018)	3000	2.50	2018	5	0

	Es_Fin_Semana
Caso 1 (Lunes Alto Stock/Bajo Precio)	0
Caso 2 (Sábado Bajo Stock/Alto Precio)	1
Caso 3 (Jueves Stock/Precio Medio)	0
Caso 4 (Venta Baja Enero 2014)	0
Caso 5 (Lunes 2018)	0

Se utilizan los datos del dataframe para predecir las ventas:

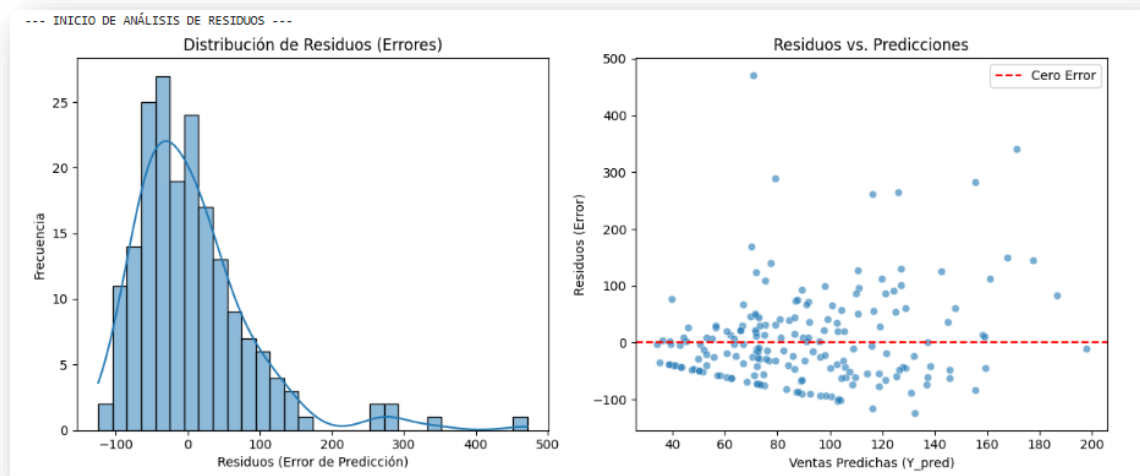
```
# Usar el modelo entrenado para predecir las ventas con los nuevos datos
y_nuevos_pred = model.predict(X_nuevos)
```

--- Resultados de Predicciones para Casos Específicos ---					
	estoque	preco	Ano	Mes	Dia_Semana
Caso 1 (Lunes Alto Stock/Bajo Precio)	4500	1.50	2017	1	0
Caso 2 (Sábado Bajo Stock/Alto Precio)	100	3.50	2016	12	5
Caso 3 (Jueves Stock/Precio Medio)	2500	2.00	2016	6	3
Caso 4 (Venta Baja Enero 2014)	4972	1.29	2014	1	2
Caso 5 (Lunes 2018)	3000	2.50	2018	5	0

	Es_Fin_Semana	Ventas_Predichas
Caso 1 (Lunes Alto Stock/Bajo Precio)	0	205.50
Caso 2 (Sábado Bajo Stock/Alto Precio)	1	115.15
Caso 3 (Jueves Stock/Precio Medio)	0	141.15
Caso 4 (Venta Baja Enero 2014)	0	88.10
Caso 5 (Lunes 2018)	0	217.00

PARTE 6: ANÁLISIS DE RESIDUOS Y PROPUESTAS DE MODELOS

En esta fase se genera el gráfico de Residuos para verificar si los errores del modelo están distribuidos aleatoriamente (un supuesto clave de la Regresión Lineal).



Elemento	Observación en la Gráfica	Implicación para el Modelo
Patrón	La nube de puntos es muy dispersa y no está uniformemente distribuida.	El error del modelo es alto y no constante (existe Heterocedasticidad).
Concentración	Hay una mayor concentración de residuos positivos (errores grandes) en el lado derecho, donde las ventas predichas son altas.	El modelo tiende a subestimar (predicción es menor que lo real) el valor de las ventas reales cuando estas son altas.
Cero Error	La línea roja ($y=0$) no está en el centro de la dispersión de puntos.	El modelo no es imparcial. Confirma el bajo R^2 (0.2374) obtenido anteriormente.

PARTE 7: CONCLUSIONES Y VERIFICACION DE LAS PREDICCIONES

La gráfica de residuos permite verificar cómo se comporta el modelo:

1. Modelo Teóricamente Válido (Supuestos Rotos): La Regresión Lineal asume que los residuos son aleatorios y tienen varianza constante (Homocedasticidad). La gráfica muestra un patrón y una variación no uniforme, lo que indica que se violan los supuestos de la Regresión Lineal.
2. Verificación de Predicciones: Las predicciones no se acercan consistentemente a los datos de entrenamiento y prueba.
 - Para las ventas más bajas, los errores (puntos) están más cerca de cero.
 - Para las ventas más altas (predichas a la derecha del eje X), el error aumenta significativamente, lo que significa que la confianza en esas predicciones es baja.

La gráfica confirma que la Regresión Lineal no es el mejor modelo para este conjunto de datos, y que las variables predictoras no son suficientes para capturar la complejidad de las ventas.

PARTE 8: MODELO ALTERNATIVO XGBOOST

XGBoost es un algoritmo basado en árboles de decisión (específicamente un algoritmo de *Gradient Boosting*), y es extremadamente robusto y versátil.

¿Por qué funcionará mejor?

- Manejo de No Linealidad: El XGBoost no está limitado por el supuesto de Relación Lineal. Puede aprender automáticamente las complejas interacciones y las curvas en los datos.
- Manejo de Heterocedasticidad: La Regresión Lineal falló porque su error era inconsistente (Heterocedasticidad). XGBoost es menos sensible a la distribución de los residuos y puede manejar mejor la varianza no constante.
- Uso de Variables Temporales: El modelo aprovechará al máximo las variables de tiempo creadas (Año, Mes, Día_Semana, Es_Fin_Semana) al encontrar umbrales y reglas complejas (ej., "si es diciembre Y fin de semana, la venta aumenta x3").

```
--- INICIO DEL MODELO XGBOOST ---  
  
--- Comparación de Desempeño ---  
R² de XGBoost: 0.3406  
MAE de XGBoost: 47.73 unidades de venta  
  
--- Desempeño de Regresión Lineal (Anterior) ---  
R² Lineal: 0.2374  
MAE Lineal: 58.08
```

Métrica	Regresión Lineal (RL)	XGBoost (Alternativa)	Mejora	Interpretación
R^2	0.2374	0.3406	43.50%	XGBoost explica un 43.5% más de la variabilidad de las ventas que el modelo Lineal.
MAE	58.08	47.73	-17.80%	El error promedio de predicción se redujo en más de 10 unidades de venta, siendo ahora approx 47.73

PARTE 9: PRUEBA CON CASOS ESPECÍFICOS XGBoost

```

--- INICIO DE PRUEBAS XGBOOST EN CASOS ESPECÍFICOS ---

--- TABLA DE COMPARACIÓN: RL vs. XGBOOST ---
                                estoque  preco  Es_Fin_Semana  \
Caso 1 (Lunes Alto Stock/Bajo Precio)    4500    1.50             0
Caso 2 (Sábado Bajo Stock/Alto Precio)     100    3.50             1
Caso 3 (Jueves Stock/Precio Medio)        2500    2.00             0
Caso 4 (Venta Baja Enero 2014)           4972    1.29             0
Caso 5 (Lunes 2018)                      3000    2.50             0

                                RL_Predichas  XGB_Predichas
Caso 1 (Lunes Alto Stock/Bajo Precio)      205.50       69.00
Caso 2 (Sábado Bajo Stock/Alto Precio)      115.15       59.37
Caso 3 (Jueves Stock/Precio Medio)          141.15      164.77
Caso 4 (Venta Baja Enero 2014)              88.10       41.26
Caso 5 (Lunes 2018)                        217.00      118.16

--- FIN DE LA ETAPA DE COMPARACIÓN ---

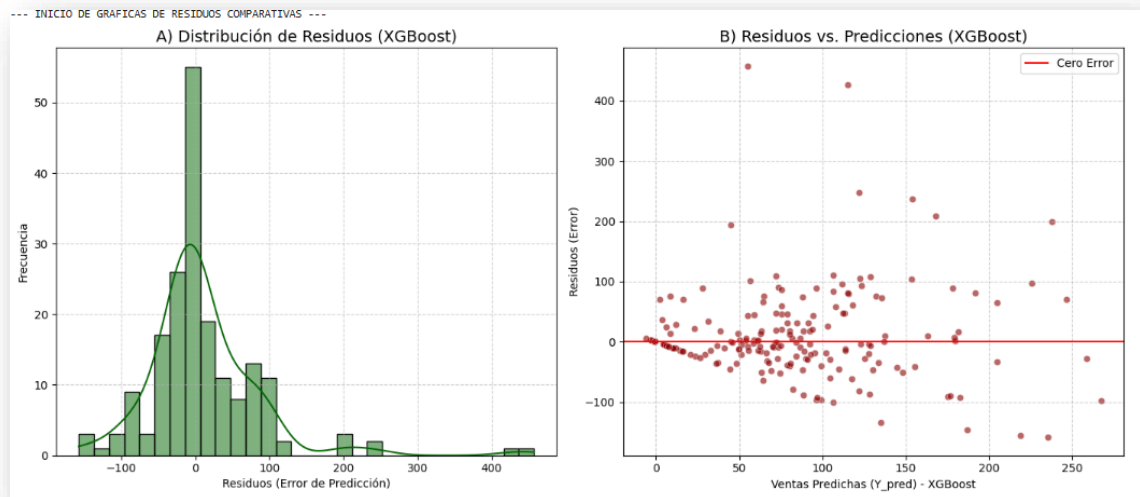
```

```

--- Resultados de Predicciones para Casos Específicos ---
                                estoque  preco  Ano  Mes  Dia_Semana
Caso 1 (Lunes Alto Stock/Bajo Precio)    4500    1.50  2017    1         0
Caso 2 (Sábado Bajo Stock/Alto Precio)     100    3.50  2016   12         5
Caso 3 (Jueves Stock/Precio Medio)        2500    2.00  2016    6         3
Caso 4 (Venta Baja Enero 2014)           4972    1.29  2014    1         2
Caso 5 (Lunes 2018)                      3000    2.50  2018    5         0

                                Es_Fin_Semana  Ventas_Predichas
Caso 1 (Lunes Alto Stock/Bajo Precio)         0         205.50
Caso 2 (Sábado Bajo Stock/Alto Precio)         1         115.15
Caso 3 (Jueves Stock/Precio Medio)             0         141.15
Caso 4 (Venta Baja Enero 2014)                 0          88.10
Caso 5 (Lunes 2018)                           0         217.00

```



Característica	Observación en la Gráfica de XGBoost	Implicación (Mejora)
Patrón	La nube de puntos es mucho más uniforme y aleatoria a lo largo del eje X. No hay una forma de "embudo" o "U" clara.	El modelo corrige la Heterocedasticidad (varianza no constante del error) que plagaba a la Regresión Lineal. Esto significa que el error del modelo es más consistente.
Centrado en Cero	La nube de puntos está muy bien centrada en la línea horizontal $y=0$ (Cero Error).	El modelo no tiene un sesgo sistemático (no subestima ni sobreestima consistentemente). Se cumple mejor el supuesto de Media Cero de los residuos.
Magnitud	Los puntos están más apretados alrededor de la línea $y=0$ que en la gráfica de RL.	El error es menor. Esto visualmente justifica la reducción del MAE de 58.08 a 47.73 .

PARTE 10: CONCLUSIONES REGRESIÓN LINEAL VRS XGBOOST

El análisis comparativo entre la Regresión Lineal Múltiple (RL) y el modelo XGBoost reveló diferencias fundamentales en su capacidad para predecir las ventas con el *dataset* proporcionado:

Regresión Lineal (Modelo Inicial)

El modelo de Regresión Lineal se basa en una ecuación fija y lineal, lo que resultó ser una limitación.

- **Desempeño y Calidad del Ajuste:** El Coeficiente de Determinación (R^2) fue bajo (0.2374), indicando que las variables seleccionadas apenas explican el 23.74% de la variabilidad en las ventas.
- **Análisis de Residuos:** La gráfica de residuos mostró un patrón de embudo (Heterocedasticidad), lo que implica una violación del supuesto de varianza constante. Los errores eran mucho mayores para las ventas predichas altas.
- **Conclusión:** El modelo se considera inadecuado para este *dataset*. Sus predicciones no son confiables, especialmente en escenarios de ventas altas, debido a errores estructurales y su incapacidad para capturar las relaciones no lineales.

XGBoost (Modelo Alternativo Propuesto)

El modelo XGBoost, siendo un algoritmo de *machine learning* basado en árboles, opera con una ecuación no lineal y adaptable que permite crear reglas de decisión complejas.

- Desempeño y Capacidad Explicativa: El R^2 mejoró sustancialmente a 0.3406 (un aumento del 43.5% en la capacidad explicativa), y el Error Absoluto Medio (MAE) se redujo de 58.08 a 47.73.
- Análisis de Residuos: La gráfica de residuos se transformó en una nube aleatoria y uniforme centrada en el eje cero. Esto demuestra que el modelo corrige la Heterocedasticidad, cumple mejor los supuestos del error y muestra un error imparcial y consistente.
- Conclusión: XGBoost se establece como el Modelo Recomendado. Corrige los principales errores de la Regresión Lineal y ofrece una precisión superior en la predicción de ventas.

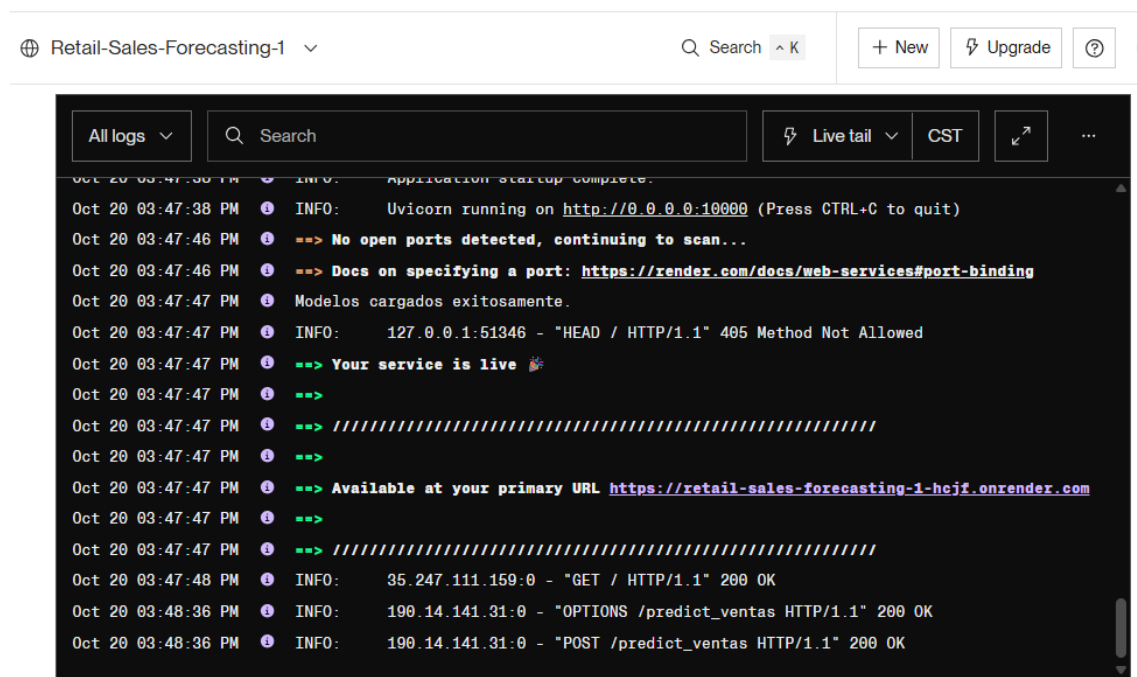
PARTE 11: DESPLIEGUE Y VISUALIZACIÓN WEB DEL MODELO

Esta fase final del proyecto se centró en llevar los modelos de Machine Learning a un entorno de producción accesible generando una interfaz de usuario (frontend) para contextualizar las predicciones mediante visualización de datos.

1. Despliegue y Corrección de la API (Backend)

Se desplegó el servicio de predicción, construido con FastAPI y alojado en Render.com, para operar de forma remota como un api de servicio.

<https://retail-sales-forecasting-1-hcjf.onrender.com>



```
Oct 20 03:47:30 PM INFO: Application startup complete.
Oct 20 03:47:38 PM INFO: Uvicorn running on http://0.0.0.0:10000 (Press CTRL+C to quit)
Oct 20 03:47:46 PM INFO: ==> No open ports detected, continuing to scan...
Oct 20 03:47:46 PM INFO: ==> Docs on specifying a port: https://render.com/docs/web-services#port-binding
Oct 20 03:47:47 PM INFO: Modelos cargados exitosamente.
Oct 20 03:47:47 PM INFO: 127.0.0.1:51346 - "HEAD / HTTP/1.1" 405 Method Not Allowed
Oct 20 03:47:47 PM INFO: ==> Your service is live 🎉
Oct 20 03:47:47 PM INFO: ==>
Oct 20 03:47:47 PM INFO: ==> //////////////////////////////////////
Oct 20 03:47:47 PM INFO: ==>
Oct 20 03:47:47 PM INFO: ==> Available at your primary URL https://retail-sales-forecasting-1-hcjf.onrender.com
Oct 20 03:47:47 PM INFO: ==>
Oct 20 03:47:47 PM INFO: ==> //////////////////////////////////////
Oct 20 03:47:48 PM INFO: 35.247.111.159:0 - "GET / HTTP/1.1" 200 OK
Oct 20 03:48:36 PM INFO: 190.14.141.31:0 - "OPTIONS /predict_ventas HTTP/1.1" 200 OK
Oct 20 03:48:36 PM INFO: 190.14.141.31:0 - "POST /predict_ventas HTTP/1.1" 200 OK
```


2. Desarrollo del Portal Web (Frontend)

Se creo una interfaz de usuario (index.html) para ofrecer un análisis dinámico y visual de las predicciones :

https://ingesistemas.sisweb.site/API_VENTAS/index.html.

- Integración de Chart.js: Se incluyó la librería Chart.js para generar una gráfica interactiva de líneas en el frontend con el objetivo de almacenar y mostrar el historial de consultas y predicciones.
- Conexión Asíncrona: La función JavaScript obtenerPredicciones() consume la API remota y actualizar el DOM (los resultados numéricos y la gráfica).

3. Contextualización Estadística y Visualización

Construcción de la gráfica de contexto de negocio al incorporar líneas de referencia estáticas:

- Cálculo de Benchmarks: Se utilizó el resumen estadístico (.describe()) del dataset de entrenamiento (937 filas) para determinar los valores de referencia clave de la columna ventas.
- Visualización de la Distribución: Se agregaron cuatro líneas de referencia estáticas al gráfico que representan la distribución histórica de las ventas, permitiendo al usuario ver el impacto de sus variables de entrada en el contexto del historial:
 - Promedio (Media): 90.53
 - Mediana (Q2): 76.00
 - Tercer Cuartil (Q3): 127.00
 - Primer Cuartil (Q1): 33.00

Esta funcionalidad permite al usuario determinar inmediatamente si las predicciones del modelo (líneas Roja y Azul) caen dentro del 50% central de las ventas históricas (entre Q1 y Q3) o si representan un escenario atípico de bajo o alto rendimiento.

Retail Sales Forecast

API desplegada en Render conectada a modelos de Regresión Lineal y XGBoost.

1. Stock (estoque):

2. Precio (precio):

3. Año (Ano):

4. Mes (Mes, 1-12):

5. Día de la Semana (0=Dom, 6=Sáb):

6. Es Fin de Semana (0=No, 1=Si):

Obtener Predicciones

Resultados de la Predicción

Valores Enviados: {"estoque":1500,"precio":1.62,"Ano":2019,"Mes":3,"Dia_Semana":3,"Es_Fin_Semana":0}

Regresión Lineal: **255.15**

XGBoost: **42.55**

Predicciones vs. Distribución Histórica (Percentiles)

