UNIVERSIDAD SAN PABLO DE GUATEMALA

Facultad de Ciencias Empresariales

Escuela de Ingeniería en Ciencias y Sistemas de la Computación

INFORME PREDICCIÓN DE VENTAS RETAIL

Trabajo presentado en el curso de **Sistemas Expertos**

Impartido por **Ing. Marcos Alfredo Orozco De Paz**

Emerson Batista 2200078

Guatemala, octubre de 2025

Informe Predicción de Ventas Retail

Modelo Regresión Lineal

El enfoque fue predecir ventas y unidades vendidas

Se manejaron **dos escenarios** para validar el resultado considerando la no estacionalidad de los datos:

Sin manejo de Outliers (se excluyen valores negativos en caso de existir)

Resultado de las predicciones vs. valores reales:

--- 8. Casos de Prueba Aleatorios (5 por Período) ---

```
--- Casos de Prueba: total_venta ---


Por Fecha (5 casos diarios aleatorios del Test Set):
     fecha  Real_TV  Pred_TV
2016-02-13   416.10   223.37
2016-07-26   231.83   194.00
2016-04-03   108.78   239.83
2016-03-08   137.27   214.31
2016-07-13   544.32   335.77

Por Día de la Semana (5 casos aleatorios, media del Test Set):
            Real_TV  Pred_TV
nombre_dia
Dom          327.81   364.25
Mié          345.35   249.03
Mar          224.10   220.01
Lun          215.02   260.61
Vie          293.81   269.28
```

Por Semana del Año (5 casos aleatorios, media del Test Set):

num_semana_ano	Real_TV	Pred_TV
21	406.26	316.06
18	396.76	345.11
22	589.41	445.30
7	108.25	185.44
27	338.01	341.18

Por Mes (5 casos aleatorios, media del Test Set):

nombre_mes	Real_TV	Pred_TV
Mar	167.52	198.57
Abr	317.14	283.58
May	318.07	287.03
Nov	NaN	NaN
Feb	206.61	212.43

--- Casos de Prueba: venta_unidades ---

Por Fecha (5 casos diarios aleatorios del Test Set):

fecha	Real_VU	Pred_VU
2016-03-16	127	82.23
2016-04-16	202	163.89
2016-03-21	74	105.53
2016-02-09	113	94.22
2016-05-22	113	117.28

Por Día de la Semana (5 casos aleatorios, media del Test Set):

nombre_dia	Real_VU	Pred_VU
Vie	133.89	136.45
Mar	101.73	107.61

```
Jue          120.70   145.94
Mié          159.46   127.54
Dom          148.81   183.56

Por Semana del Año (5 casos aleatorios, media del Test Set):
                Real_VU  Pred_VU
num_semana_ano
29               146.00   152.36
4                207.25   154.43
13                50.29    92.80
24               299.14   255.94
20                86.00    75.68

Por Mes (5 casos aleatorios, media del Test Set):
            Real_VU  Pred_VU
nombre_mes
Oct             NaN      NaN
Jun          211.57   201.74
May          142.16   140.98
Ago             NaN      NaN
Feb           97.66   110.55
```

--- Métricas de Desempeño (Modelo total_venta) ---

```
MAE (total_venta):   99.53
RMSE (total_venta):  132.97
R² (total_venta):    0.4493
```

--- Métricas de Desempeño (Modelo venta_unidades) ---

```
MAE (venta_unidades):   47.80
RMSE (venta_unidades):  62.23
R² (venta_unidades):    0.5212
```

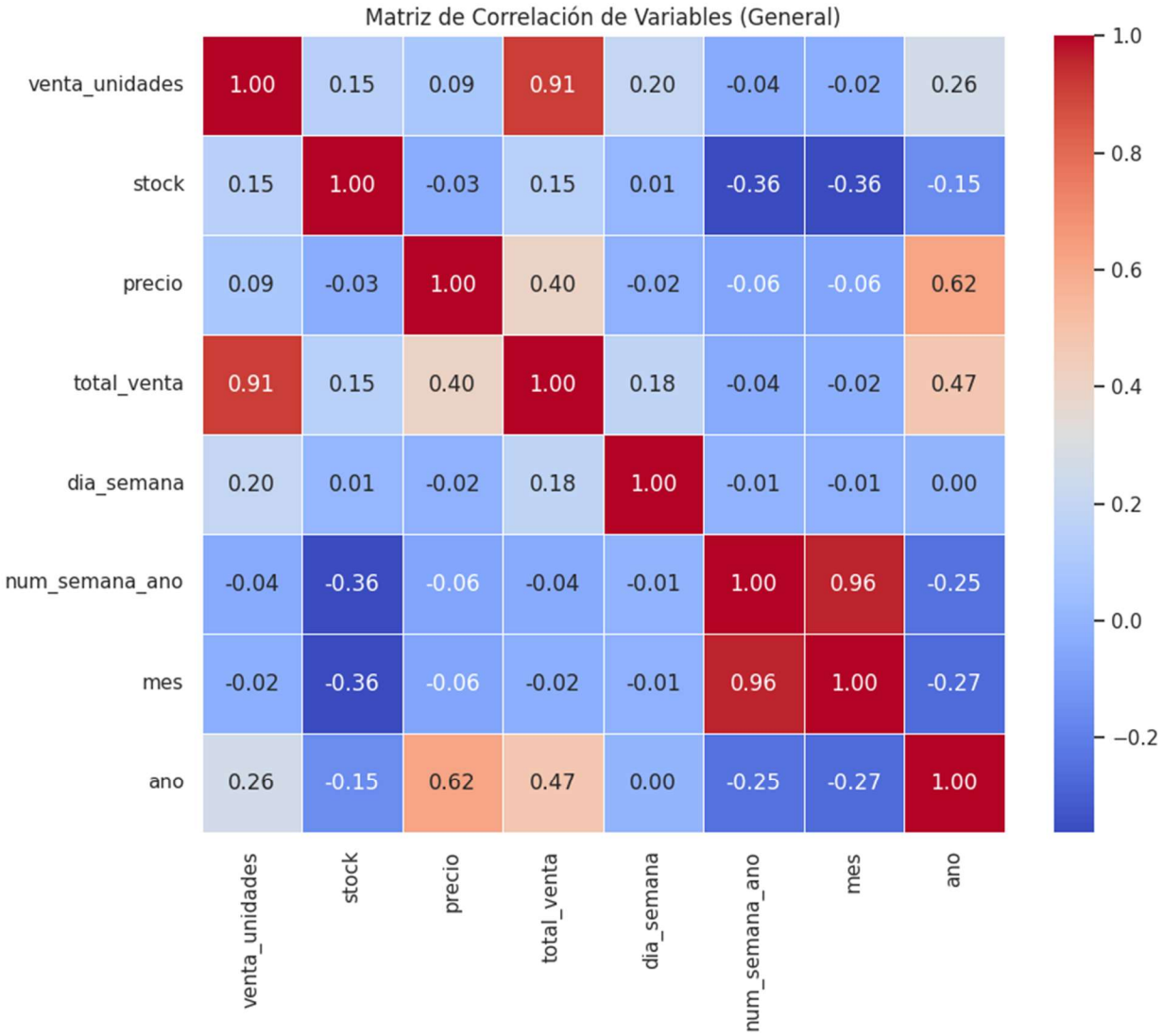
Código clave:

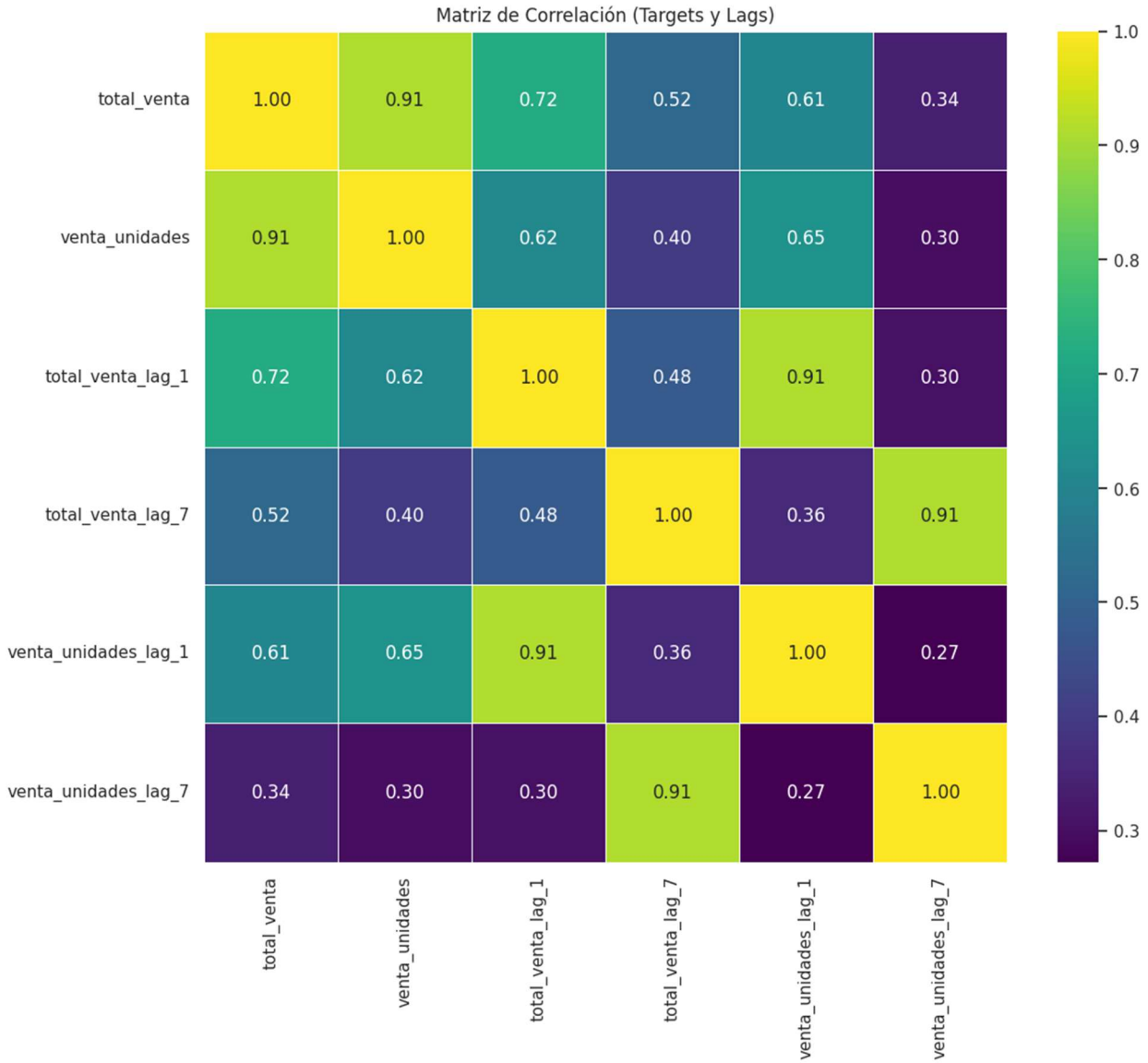
```
# --- 6. Preprocesamiento y Entrenamiento (Doble Modelo) ---

print("\n--- 6. Preprocesamiento y Entrenamiento (Doble Modelo) ---")


# --- 6.1 Preprocesamiento ---


# Se definen las variables predictoras (X)

features = [

    'dia_semana', 'num_semana_ano', 'mes', 'ano',

    'precio', 'stock',

    'precio_roll_7', 'stock_roll_7',

    'total_venta_lag_1', 'total_venta_lag_7', 'total_venta_roll_30',
'total_venta_roll_365',

    'venta_unidades_lag_1', 'venta_unidades_lag_7', 'venta_unidades_roll_30',
'venta_unidades_roll_365'

]

# Se definen los dos targets (y)

target_tv = 'total_venta'

target_vu = 'venta_unidades'


# Se asegura que todas las features existen

features_presentes = [col for col in features if col in df_model.columns]

X = df_model[features_presentes]

y_tv = df_model[target_tv]

y_vu = df_model[target_vu]


print(f"Se usarán {len(features_presentes)} variables predictoras.")

print(f"Target 1: {target_tv}")

print(f"Target 2: {target_vu}")


# División de datos (train/test) para AMBOS targets

# Se usa 'shuffle=False' por ser series temporales

X_train, X_test, y_tv_train, y_tv_test, y_vu_train, y_vu_test = train_test_split(

    X, y_tv, y_vu, test_size=0.2, shuffle=False
```

```
)

print(f"Tamaño de datos de entrenamiento: {X_train.shape[0]} filas")
print(f"Tamaño de datos de prueba: {X_test.shape[0]} filas")

# Escalado de características
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
print("Datos de entrenamiento y prueba escalados (StandardScaler).")

# --- 6.2 Entrenamiento ---

# Modelo 1: total_venta
print("\nEntrenando Modelo 1 (para total_venta)...")
model_tv = LinearRegression()
model_tv.fit(X_train_scaled, y_tv_train)
print("Modelo 1 (total_venta) entrenado.")

# Modelo 2: venta_unidades
print("Entrenando Modelo 2 (para venta_unidades)...")
model_vu = LinearRegression()
model_vu.fit(X_train_scaled, y_vu_train)
print("Modelo 2 (venta_unidades) entrenado.")

# --- 6.3 Evaluación ---

# Predicciones de ambos modelos
y_pred_tv = model_tv.predict(X_test_scaled)
y_pred_vu = model_vu.predict(X_test_scaled)

# Métricas para total_venta
print("\n--- Métricas de Desempeño (Modelo total_venta) ---")
mae_tv = mean_absolute_error(y_tv_test, y_pred_tv)
```

```
mse_tv = mean_squared_error(y_tv_test, y_pred_tv)
rmse_tv = np.sqrt(mse_tv) # Corrección para versiones antiguas de sklearn
r2_tv = r2_score(y_tv_test, y_pred_tv)

print(f"MAE (total_venta):   {mae_tv:.2f}")
print(f"RMSE (total_venta):  {rmse_tv:.2f}")
print(f"R² (total_venta):    {r2_tv:.4f}")

# Métricas para venta_unidades
print("\n--- Métricas de Desempeño (Modelo venta_unidades) ---")
mae_vu = mean_absolute_error(y_vu_test, y_pred_vu)
mse_vu = mean_squared_error(y_vu_test, y_pred_vu)
rmse_vu = np.sqrt(mse_vu) # Corrección para versiones antiguas de sklearn
r2_vu = r2_score(y_vu_test, y_pred_vu)

print(f"MAE (venta_unidades):   {mae_vu:.2f}")
print(f"RMSE (venta_unidades):  {rmse_vu:.2f}")
print(f"R² (venta_unidades):    {r2_vu:.4f}")
```

Gráficas


Matriz de Correlación de Variables (General)
venta_unidades
stock
precio
total_venta
dia_semana
num_semana_ano
mes
ano
1.00 0.15 0.09 0.91 0.20 -0.04 -0.02 0.26
0.15 1.00 -0.03 0.15 0.01 -0.36 -0.36 -0.15
0.09 -0.03 1.00 0.40 -0.02 -0.06 -0.06 0.62
0.91 0.15 0.40 1.00 0.18 -0.04 -0.02 0.47
0.20 0.01 -0.02 0.18 1.00 -0.01 -0.01 0.00
-0.04 -0.36 -0.06 -0.04 -0.01 1.00 0.96 -0.25
-0.02 -0.36 -0.06 -0.02 -0.01 0.96 1.00 -0.27
0.26 -0.15 0.62 0.47 0.00 -0.25 -0.27 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Matriz de Correlación (Targets y Lags)
total_venta
venta_unidades
total_venta_lag_1
total_venta_lag_7
venta_unidades_lag_1
venta_unidades_lag_7
1.00 0.91 0.72 0.52 0.61 0.34
0.91 1.00 0.62 0.40 0.65 0.30
0.72 0.62 1.00 0.48 0.91 0.30
0.52 0.40 0.48 1.00 0.36 0.91
0.61 0.65 0.91 0.36 1.00 0.27
0.34 0.30 0.30 0.91 0.27 1.00
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3

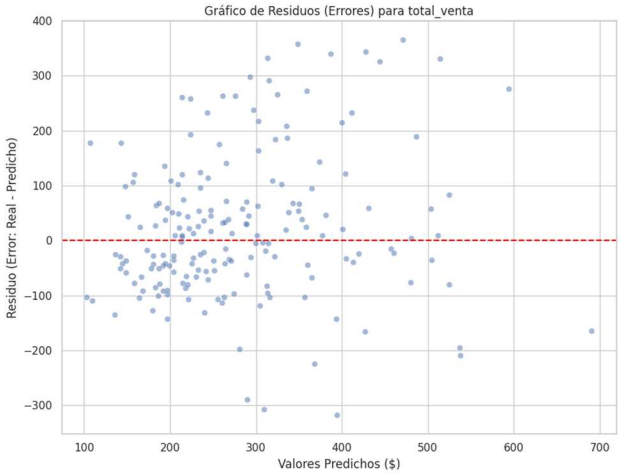


Gráfico de Residuos (Errores) para total_venta
Residuo (Error: Real - Predicho)
Valores Predichos ($)
400
300
200
100
0
−100
−200
−300
100
200
300
400
500
600
700

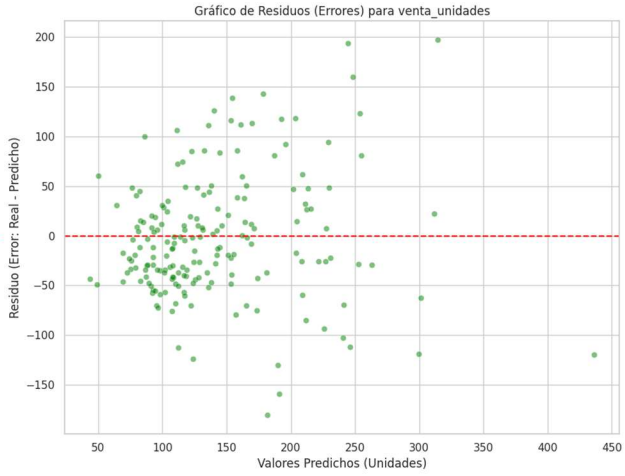
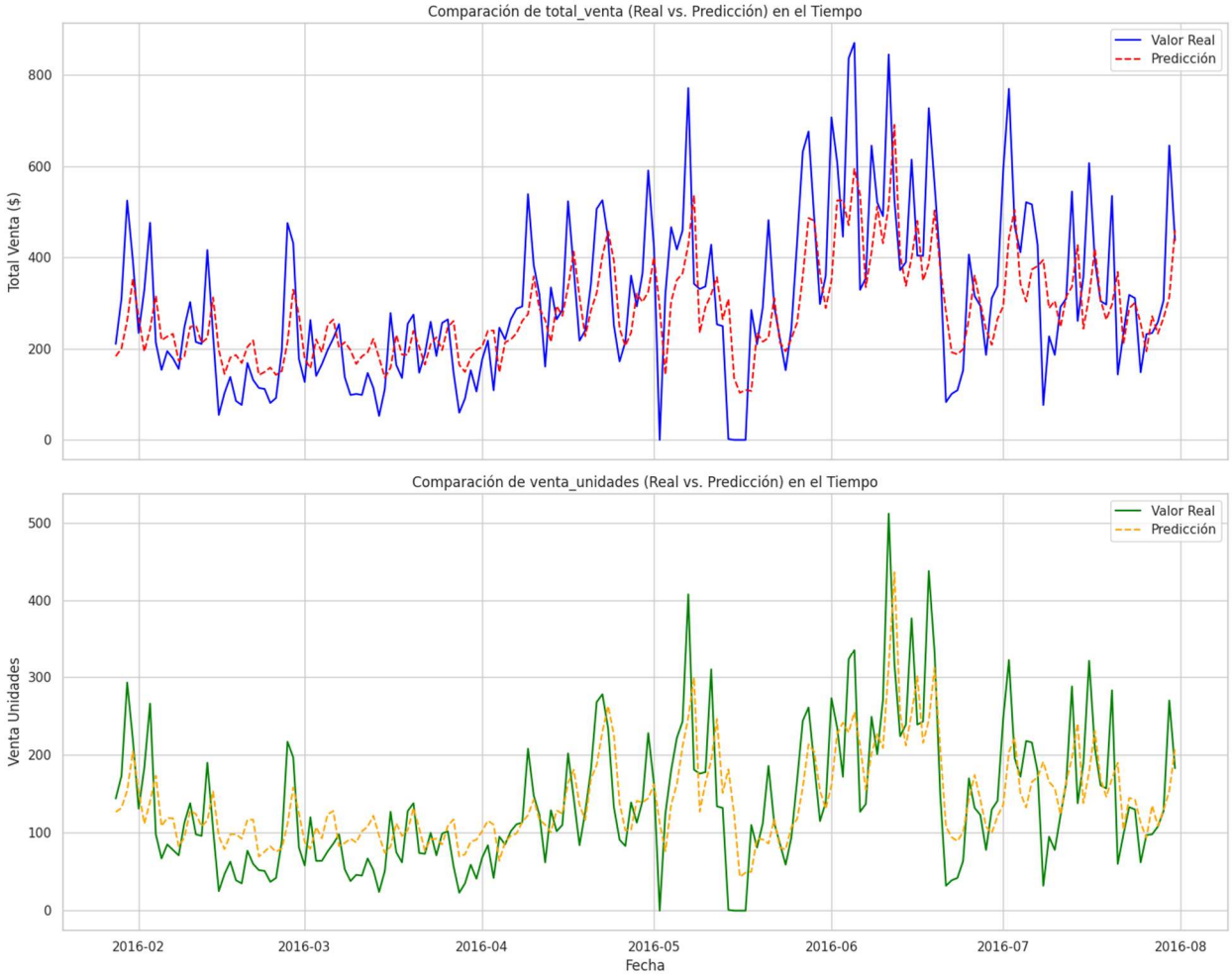
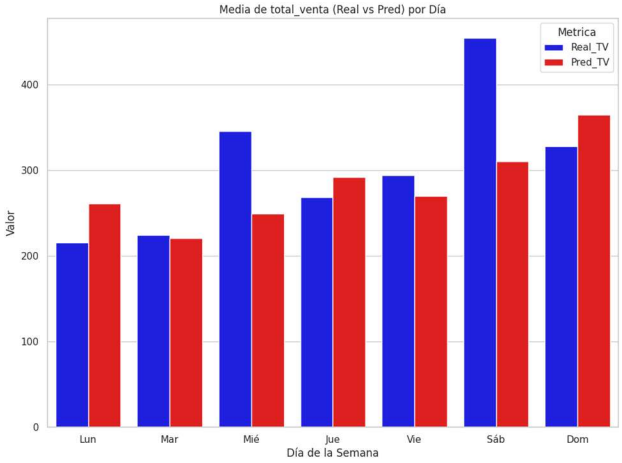
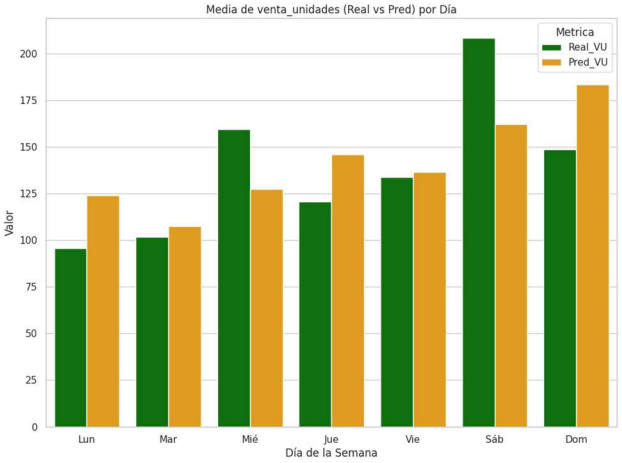
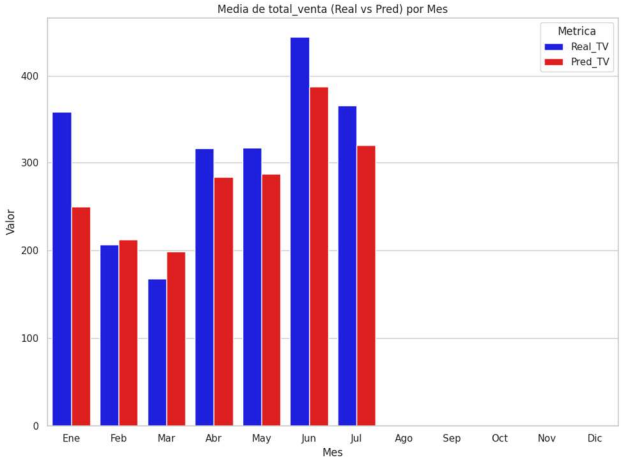


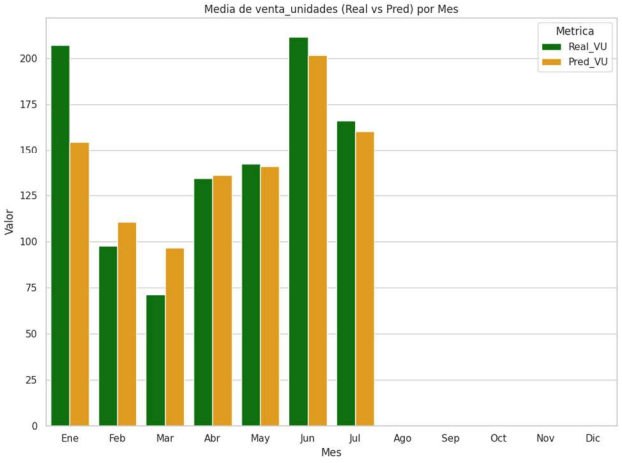
Gráfico de Residuos (Errores) para venta_unidades
Residuo (Error: Real - Predicho)
Valores Predichos (Unidades)
200
150
100
50
0
−50
−100
−150
50
100
150
200
250
300
350
400
450


Comparación de total_venta (Real vs. Predicción) en el Tiempo
Valor Real
Predicción
Total Venta ($)
0
200
400
600
800
Comparación de venta_unidades (Real vs. Predicción) en el Tiempo
Valor Real
Predicción
Venta Unidades
0
100
200
300
400
500
2016-02
2016-03
2016-04
2016-05
2016-06
2016-07
2016-08
Fecha


Media de total_venta (Real vs Pred) por Día
Metrica
Real_TV
Pred_TV
Valor
0
100
200
300
400
Lun
Mar
Mié
Jue
Vie
Sáb
Dom
Día de la Semana


Media de venta_unidades (Real vs Pred) por Día
Metrica
Real_VU
Pred_VU
Valor
0
25
50
75
100
125
150
175
200
Lun
Mar
Mié
Jue
Vie
Sáb
Dom
Día de la Semana


Media de total_venta (Real vs Pred) por Mes
Metrica
Real_TV
Pred_TV
Valor
0
100
200
300
400
Ene
Feb
Mar
Abr
May
Jun
Jul
Ago
Sep
Oct
Nov
Dic
Mes


Media de venta_unidades (Real vs Pred) por Mes
Metrica
Real_VU
Pred_VU
Valor
0
25
50
75
100
125
150
175
200
Ene
Feb
Mar
Abr
May
Jun
Jul
Ago
Sep
Oct
Nov
Dic
Mes

Con manejo de Outliers (Criterio Interquartiles)

Resultado de las predicciones vs. valores reales:

--- 8. Casos de Prueba Aleatorios (5 por Período) ---

```
--- 8. Casos de Prueba Aleatorios (5 por Período) ---

--- Casos de Prueba: total_venta ---

Por Fecha (5 casos diarios aleatorios del Test Set):
      fecha  Real_TV  Pred_TV
```

2016-02-13   416.10   232.17

2016-07-26   231.83   215.75

2016-04-03   108.78   245.60

2016-03-08   137.27   227.28

2016-07-13   506.52   328.28

Por Día de la Semana (5 casos aleatorios, media del Test Set):

nombre_dia	Real_TV	Pred_TV
Dom	314.17	349.48
Mié	333.25	257.17
Mar	224.10	235.74
Lun	215.02	267.26
Vie	292.91	282.50

Por Semana del Año (5 casos aleatorios, media del Test Set):

num_semana_ano	Real_TV	Pred_TV
21	406.26	337.21
18	358.96	336.68
22	542.05	454.30
7	108.25	191.56
27	338.01	355.00

Por Mes (5 casos aleatorios, media del Test Set):

nombre_mes	Real_TV	Pred_TV
Mar	167.52	206.22
Abr	316.51	297.11
May	307.66	291.92
Nov	NaN	NaN
Feb	206.61	219.75

--- Casos de Prueba: venta_unidades ---

Por Fecha (5 casos diarios aleatorios del Test Set):

fecha	Real_VU	Pred_VU
2016-03-16	127	85.59
2016-04-16	202	173.26
2016-03-21	74	115.41
2016-02-09	113	94.88
2016-05-22	113	134.28

Por Día de la Semana (5 casos aleatorios, media del Test Set):

nombre_dia	Real_VU	Pred_VU
Vie	133.41	147.08
Mar	101.73	120.04
Jue	120.70	152.82
Mié	152.12	134.17
Dom	142.04	178.77

Por Semana del Año (5 casos aleatorios, media del Test Set):

num_semana_ano	Real_VU	Pred_VU
29	143.86	157.52
4	201.00	162.01
13	50.29	94.82
24	249.86	236.84
20	86.00	78.05

Por Mes (5 casos aleatorios, media del Test Set):

nombre_mes	Real_VU	Pred_VU
Oct	NaN	NaN
Jun	185.93	199.04
May	136.29	146.72

```
Ago              NaN      NaN
Feb            97.66   115.89
```

--- Métricas de Desempeño (Modelo total_venta) ---

```
MAE (total_venta):   86.63
RMSE (total_venta):  113.38
R² (total_venta):    0.4862
```

--- Métricas de Desempeño (Modelo venta_unidades) ---

```
MAE (venta_unidades):   42.46
RMSE (venta_unidades):  52.52
R² (venta_unidades):    0.5172
```

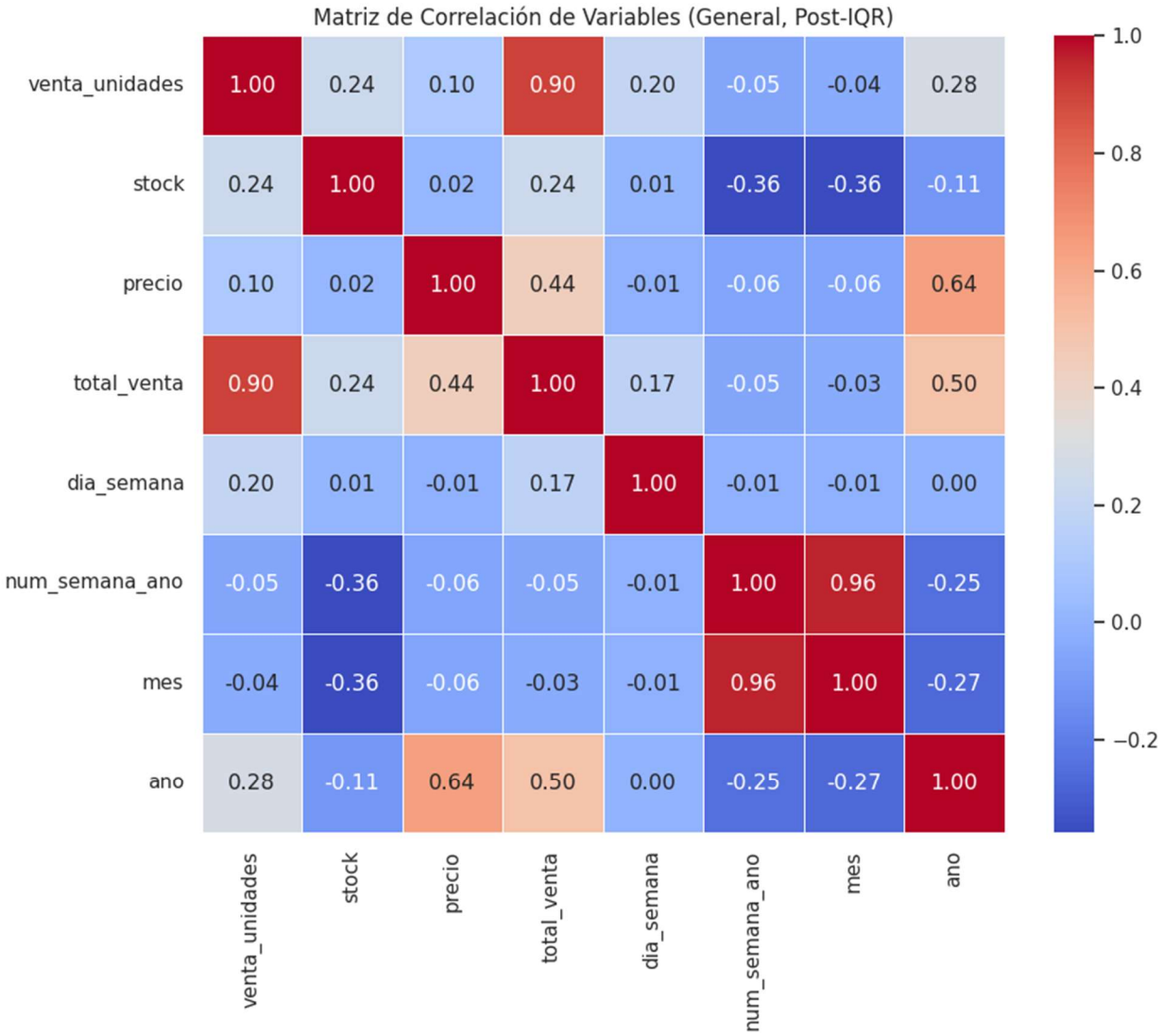
Código clave:

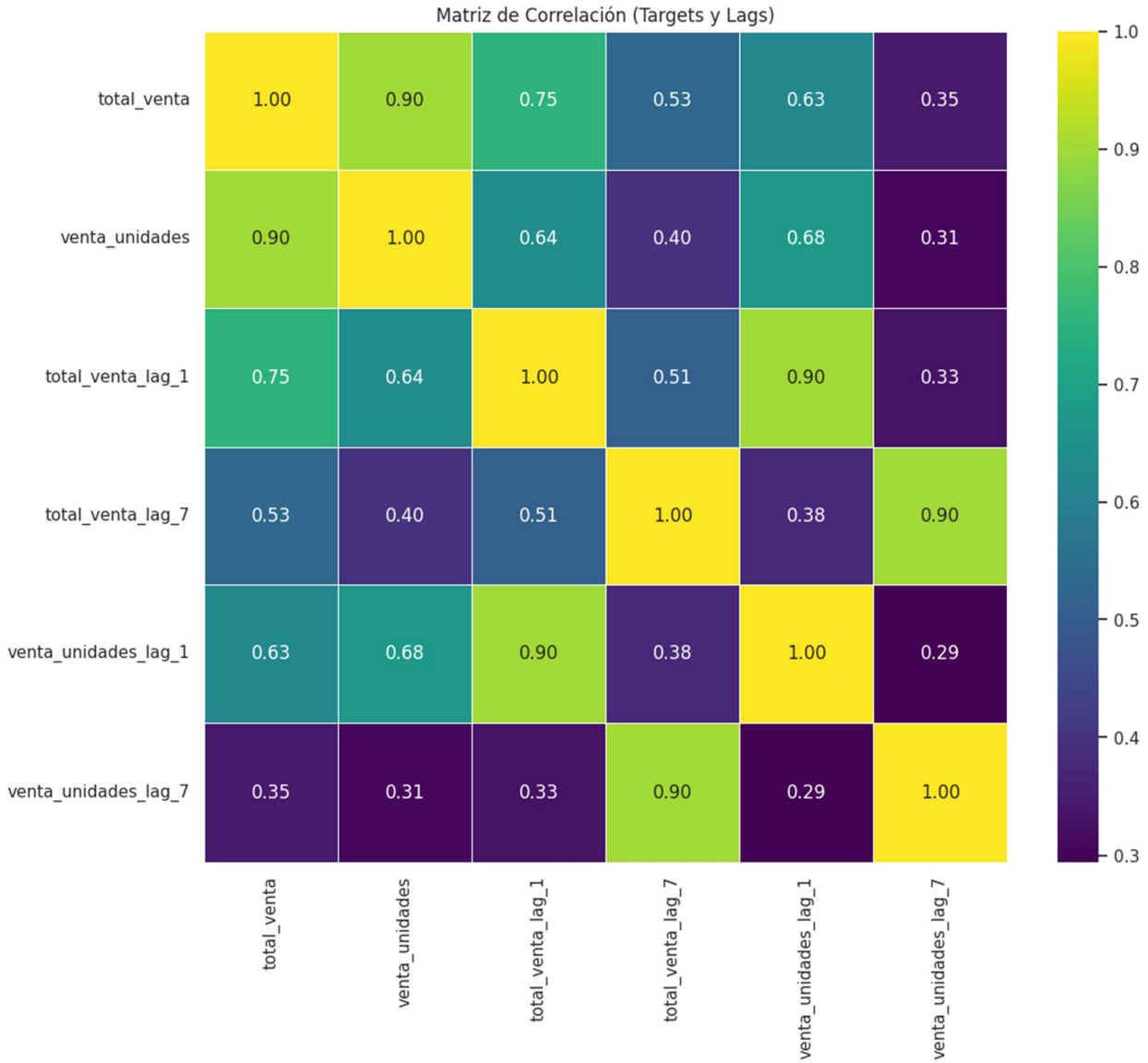
```
# --- 6. Preprocesamiento y Entrenamiento (Doble Modelo) ---
print("\n--- 6. Preprocesamiento y Entrenamiento (Doble Modelo) ---")

# --- 6.1 Preprocesamiento ---
print("Preparando datos para el modelo...")

# Se definen las variables predictoras (X)
features = [
    'dia_semana', 'num_semana_ano', 'mes', 'ano',
    'precio', 'stock',
    'precio_roll_7', 'stock_roll_7',
    'total_venta_lag_1', 'total_venta_lag_7', 'total_venta_roll_7',
    'venta_unidades_lag_1', 'venta_unidades_lag_7', 'venta_unidades_roll_7'
]
# Se definen los dos targets (y)
target_tv = 'total_venta'
target_vu = 'venta_unidades'

features_presentes = [col for col in features if col in df_model.columns]
```

```
X = df_model[features_presentes]
y_tv = df_model[target_tv]
y_vu = df_model[target_vu]

print(f"Se usarán {len(features_presentes)} variables predictoras.")
print(f"Target 1: {target_tv}")
print(f"Target 2: {target_vu}")

# División de datos (train/test)
X_train, X_test, y_tv_train, y_tv_test, y_vu_train, y_vu_test = train_test_split(
    X, y_tv, y_vu, test_size=0.2, shuffle=False
)

print(f"Tamaño de datos de entrenamiento: {X_train.shape[0]} filas")
print(f"Tamaño de datos de prueba: {X_test.shape[0]} filas")

# Escalado de características
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
print("Datos de entrenamiento y prueba escalados (StandardScaler).")

# --- 6.2 Entrenamiento ---

# Modelo 1: total_venta
print("\nEntrenando Modelo 1 (para total_venta)...")
model_tv = LinearRegression()
model_tv.fit(X_train_scaled, y_tv_train)
print("Modelo 1 (total_venta) entrenado.")

# Modelo 2: venta_unidades
print("Entrenando Modelo 2 (para venta_unidades)...")
model_vu = LinearRegression()
model_vu.fit(X_train_scaled, y_vu_train)
```

```
print("Modelo 2 (venta_unidades) entrenado.")



# --- 6.3 Evaluación ---
print("\nEvaluando modelos...")

# Predicciones de ambos modelos
y_pred_tv = model_tv.predict(X_test_scaled)
y_pred_vu = model_vu.predict(X_test_scaled)

# Métricas para total_venta
print("\n--- Métricas de Desempeño (Modelo total_venta) ---")
mae_tv = mean_absolute_error(y_tv_test, y_pred_tv)
mse_tv = mean_squared_error(y_tv_test, y_pred_tv)
rmse_tv = np.sqrt(mse_tv) # Corrección para versiones antiguas de sklearn
r2_tv = r2_score(y_tv_test, y_pred_tv)
print(f"MAE (total_venta):   {mae_tv:.2f}")
print(f"RMSE (total_venta):  {rmse_tv:.2f}")
print(f"R² (total_venta):    {r2_tv:.4f}")

# Métricas para venta_unidades
print("\n--- Métricas de Desempeño (Modelo venta_unidades) ---")
mae_vu = mean_absolute_error(y_vu_test, y_pred_vu)
mse_vu = mean_squared_error(y_vu_test, y_pred_vu)
rmse_vu = np.sqrt(mse_vu) # Corrección para versiones antiguas de sklearn
r2_vu = r2_score(y_vu_test, y_pred_vu)
print(f"MAE (venta_unidades):   {mae_vu:.2f}")
print(f"RMSE (venta_unidades):  {rmse_vu:.2f}")
print(f"R² (venta_unidades):    {r2_vu:.4f}")
```

Gráficas


Matriz de Correlación de Variables (General, Post-IQR)
venta_unidades
stock
precio
total_venta
dia_semana
num_semana_ano
mes
ano
1.00 0.24 0.10 0.90 0.20 -0.05 -0.04 0.28
0.24 1.00 0.02 0.24 0.01 -0.36 -0.36 -0.11
0.10 0.02 1.00 0.44 -0.01 -0.06 -0.06 0.64
0.90 0.24 0.44 1.00 0.17 -0.05 -0.03 0.50
0.20 0.01 -0.01 0.17 1.00 -0.01 -0.01 0.00
-0.05 -0.36 -0.06 -0.05 -0.01 1.00 0.96 -0.25
-0.04 -0.36 -0.06 -0.03 -0.01 0.96 1.00 -0.27
0.28 -0.11 0.64 0.50 0.00 -0.25 -0.27 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Matriz de Correlación (Targets y Lags)
total_venta
venta_unidades
total_venta_lag_1
total_venta_lag_7
venta_unidades_lag_1
venta_unidades_lag_7
1.00 0.90 0.75 0.53 0.63 0.35
0.90 1.00 0.64 0.40 0.68 0.31
0.75 0.64 1.00 0.51 0.90 0.33
0.53 0.40 0.51 1.00 0.38 0.90
0.63 0.68 0.90 0.38 1.00 0.29
0.35 0.31 0.33 0.90 0.29 1.00
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3

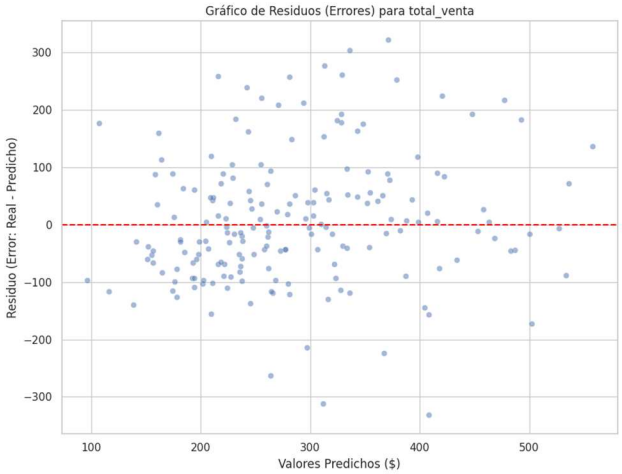


Gráfico de Residuos (Errores) para total_venta
Residuo (Error: Real - Predicho)
Valores Predichos ($)
300
200
100
0
−100
−200
−300
100
200
300
400
500

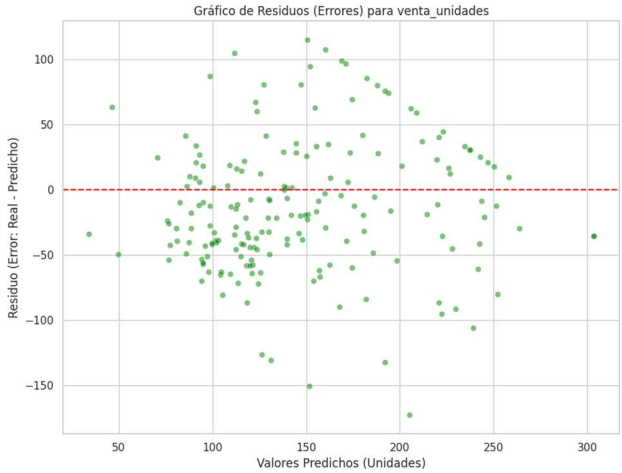
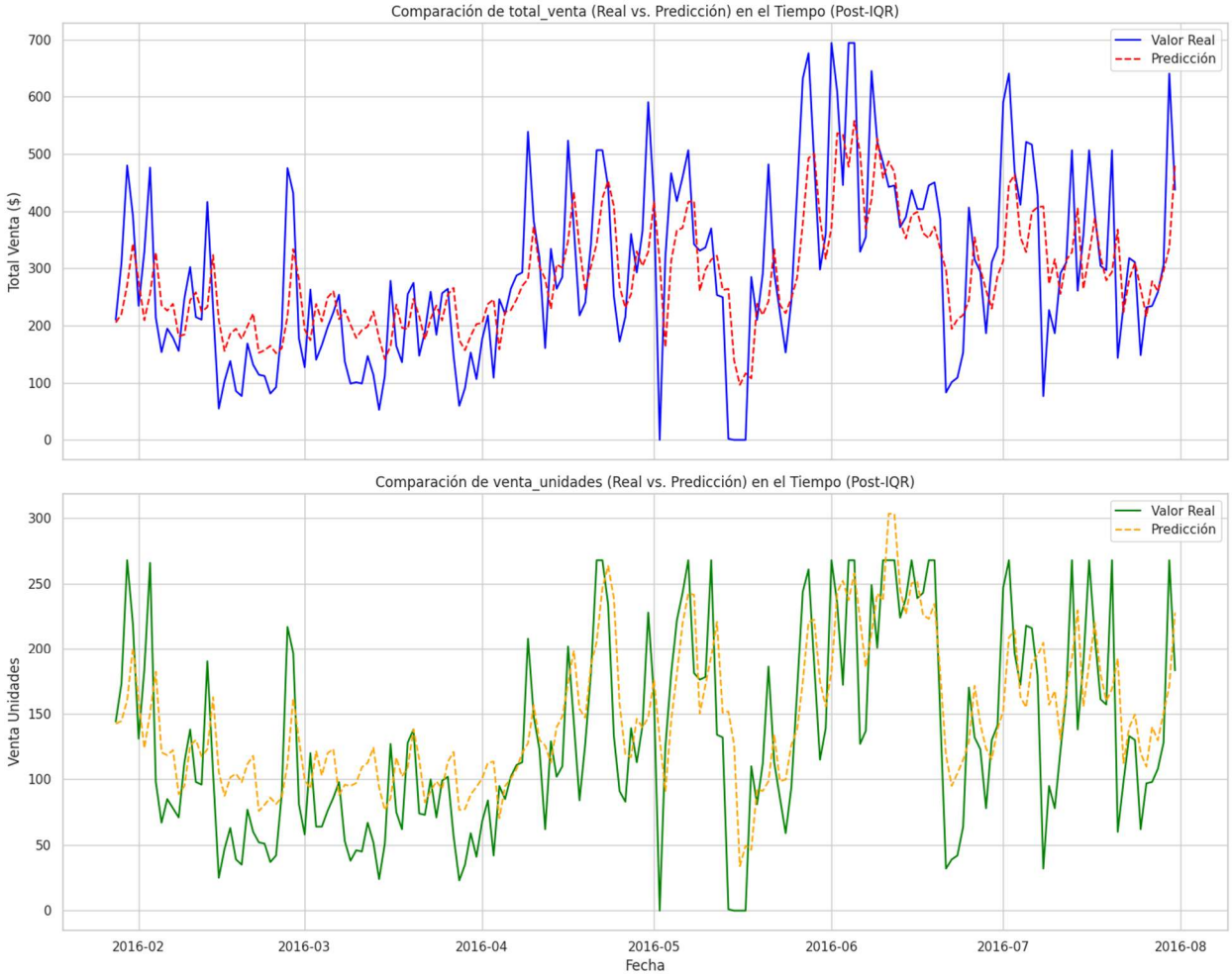
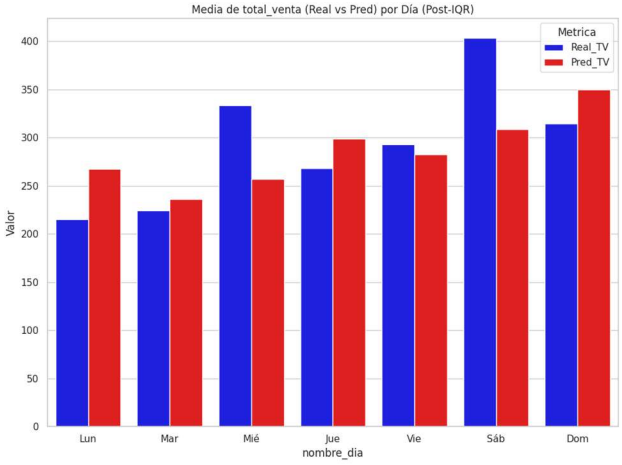
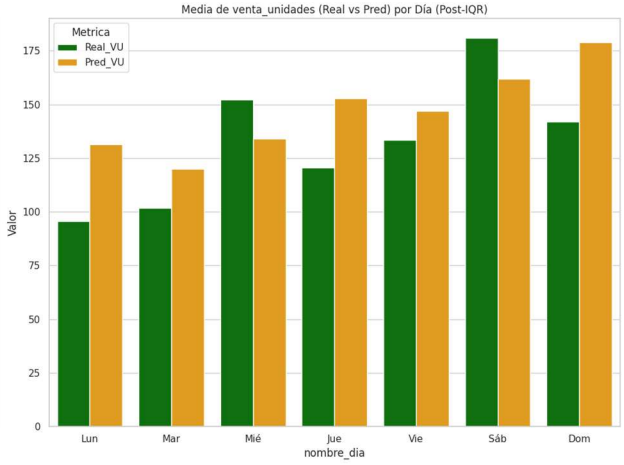
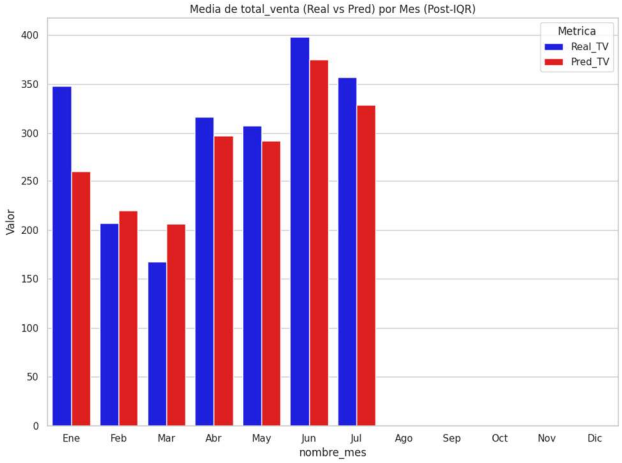


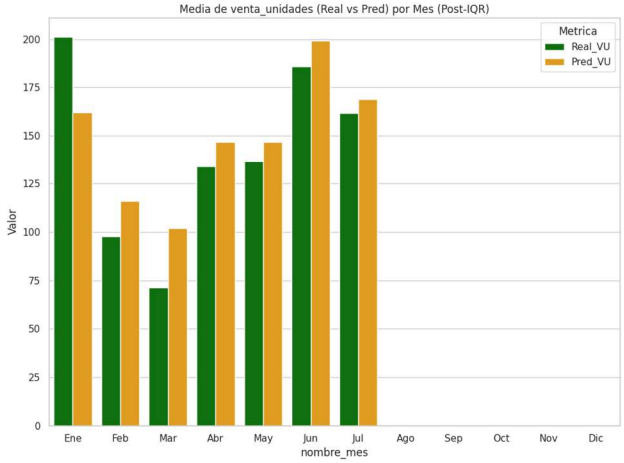
Gráfico de Residuos (Errores) para venta_unidades
Residuo (Error: Real - Predicho)
Valores Predichos (Unidades)
100
50
0
−50
−100
−150
50
100
150
200
250
300


Comparación de total_venta (Real vs. Predicción) en el Tiempo (Post-IQR)
Valor Real
Predicción
Total Venta ($)
0
100
200
300
400
500
600
700
Comparación de venta_unidades (Real vs. Predicción) en el Tiempo (Post-IQR)
Valor Real
Predicción
Venta Unidades
0
50
100
150
200
250
300
2016-02
2016-03
2016-04
2016-05
2016-06
2016-07
2016-08
Fecha


Media de total_venta (Real vs Pred) por Día (Post-IQR)
Metrica
Real_TV
Pred_TV
Valor
nombre_dia
Lun
Mar
Mié
Jue
Vie
Sáb
Dom


Media de venta_unidades (Real vs Pred) por Día (Post-IQR)
Metrica
Real_VU
Pred_VU
Valor
nombre_dia
Lun
Mar
Mié
Jue
Vie
Sáb
Dom


Media de total_venta (Real vs Pred) por Mes (Post-IQR)
Metrica
Real_TV
Pred_TV
Valor
nombre_mes
Ene
Feb
Mar
Abr
May
Jun
Jul
Ago
Sep
Oct
Nov
Dic


Media de venta_unidades (Real vs Pred) por Mes (Post-IQR)
Metrica
Real_VU
Pred_VU
Valor
nombre_mes
Ene
Feb
Mar
Abr
May
Jun
Jul
Ago
Sep
Oct
Nov
Dic

Resumen interpretativo

Hallazgos Obtenidos del Análisis de Datos

El análisis exploratorio (EDA) reveló patrones claros, los gráficos de boxplot y de agregación mostraron una fuerte estacionalidad semanal (las ventas fluctúan consistentemente según el día de la semana) y mensual/anual (las ventas suben y bajan en ciertos meses del año).

La ingeniería de variables y la matriz de correlación demostraron que las ventas son altamente dependientes del pasado inmediato. Las ventas de ayer (lag_1) y las de la semana pasada (lag_7) fueron fuertes predictores de las ventas de hoy.

Los datos no son "normales" y tienen dos tipos de outliers que definen el problema, este fue el hallazgo más importante:

Existen días con ventas extremadamente altas (ej. venda_unidades de 542), que actúan como picos que distorsionan los promedios.

Casi el 12% del dataset (112 filas) tienen ventas de 0. Estos no son errores, sino eventos reales (días cerrados, festivos, etc.) que forman una parte fundamental del comportamiento de los datos.

Comportamiento del Modelo de Regresión Lineal

El modelo de Regresión Lineal actuó como un buen detector de patrones, pero como un predictor pobre.

Fortalezas:

Gracias a las variables de tiempo (mes, día) y los lags, el modelo entendió la "ola" general de las ventas. En las gráficas de predicción agregadas (por mes o día de la semana), se observa que la línea de predicción (naranja/roja) sigue la *forma* de la línea real (azul/verde).

Se logró un R^2 de ~0.53 (para venta_unidades). Esto nos dice que, usando una simple línea recta, pudimos explicar el 53% de la variabilidad de las ventas. Esto confirma que los datos son predecibles.

Limitaciones:

El modelo falla por completo en predecir la volatilidad diaria. Las gráficas de serie temporal (diarias) muestran una línea de predicción "suave" que pasa por el medio, incapaz de alcanzar los picos de ventas altos o de bajar a los valles de cero ventas.

Se probaron tres estrategias diferentes para manejar los atípicos (ignorarlos, recortarlos con IQR, y recalcularlos de forma coherente) y en todos los casos, el R^2 se estancó en ~0.50. Esto prueba que el problema no son los datos, sino el modelo. La Regresión Lineal es una herramienta demasiado simple para un problema no lineal.

Conclusiones sobre la Calidad de las Predicciones

Las predicciones diarias NO SON FIABLES. Las métricas de error (MAE de 42 unidades y RMSE de 52 unidades) son demasiado altas en comparación con la venta media. Los casos de prueba "Por Fecha" confirman que el modelo puede fallar por 40 o 50 unidades en un día específico. No se deben usar estas predicciones para gestionar inventario diario.

Predicciones Agregadas MODERADAMENTE ACEPTABLES. Los casos de prueba "Por Mes" (ej. Junio: Real 185.93 vs Pred 199.04) y "Por Semana" fueron mucho más precisos. El modelo es significativamente mejor prediciendo el promedio de ventas de un período largo (como un mes o una semana) que un día individual.

El modelo de Regresión Lineal actual sirve como una línea base que captura la mitad del patrón, pero es insuficiente para una predicción de negocio fiable debido a su incapacidad para manejar los días cero y los picos de venta. El análisis concluye que para superar el R^2 de 0.53, es obligatorio usar un modelo más avanzado.

Propuesta de Otros Modelos

El análisis previo concluyó que el modelo de Regresión Lineal, si bien es una línea base funcional, tiene un rendimiento limitado (R^2 ~0.50) debido a la naturaleza no lineal de los datos (presencia de 12% de valores cero y picos de venta atípicos).

Regresión Ridge y Lasso

Son regularizaciones de la Regresión Lineal. Añaden una penalización al modelo para reducir la complejidad y prevenir el sobreajuste (overfitting).

Es poco probable que ofrezcan una mejora significativa para este dataset. El problema identificado no es el sobreajuste, sino el subajuste (underfitting): el modelo es demasiado simple. Dado que Ridge y Lasso siguen siendo fundamentalmente modelos lineales, heredarían la misma incapacidad para gestionar los picos de venta y los días cero que se observó en la regresión simple.

Se emplean cuando se trabaja con datos de alta dimensionalidad (cientos o miles de variables predictoras) y se sospecha que muchas de esas variables son redundantes o introducen ruido.

Random Forest (Bosque Aleatorio)

Es un modelo de ensamblaje (ensemble) que construye múltiples árboles de decisión durante el entrenamiento. La predicción final es el promedio de las predicciones de todos los árboles individuales.

Es un candidato ideal y el siguiente paso lógico en el análisis de este dataset. Su arquitectura le permite superar las limitaciones de la Regresión Lineal:

Es muy robusto con los valores atípicos. Un pico de venta extremo (ej. 542 unidades) quedará aislado en una "rama" de un árbol sin distorsionar las predicciones del resto de los datos.

Puede aprender reglas específicas para los días cero (ej. "si es lunes, la venta es 0") sin que esto afecte las reglas para los días de ventas altas.

Captura relaciones complejas que la regresión no puede (ej. "las ventas suben los sábados, *solo si* el precio es bajo").

Es el modelo de referencia para datos tabulares (como un CSV) donde se sospecha la existencia de relaciones no lineales e interacciones complejas entre variables.

XGBoost (Extreme Gradient Boosting)

Es un modelo avanzado de ensamblaje basado en "boosting". Construye árboles de forma secuencial, donde cada nuevo árbol se entrena para corregir los errores cometidos por el árbol anterior.

Es el candidato de más alto rendimiento, ofrece todas las ventajas de Random Forest (manejo de outliers, ceros y no linealidad), pero su método de corrección secuencial suele llevar a una precisión superior. Es particularmente efectivo en datasets con distribuciones de datos desbalanceadas o "infladas en cero", como es este caso.

Se emplea cuando el objetivo principal es la máxima precisión predictiva en problemas de datos tabulares. Es el estándar en competencias de ciencia de datos por su alto desempeño.

Prophet

Es una librería de pronóstico desarrollada por Meta (Facebook), diseñada específicamente para analizar series temporales.

Representa un enfoque alternativo muy bueno para este análisis. En lugar de requerir una ingeniería de variables manual (como los lags), Prophet descompone automáticamente la serie temporal en sus componentes principales: tendencia a largo plazo, estacionalidad semanal y estacionalidad anual. Maneja bien los datos faltantes y los outliers.

Es apropiado para problemas donde el tiempo es el predictor dominante. Es ideal para pronosticar la demanda basándose en patrones de calendario (días festivos, fines de semana, temporadas) de forma rápida y robusta.

Redes Neuronales Recurrentes (LSTM)

Long Short-Term Memory (LSTM) es un tipo de red neuronal profunda (Deep Learning) diseñada para aprender patrones en secuencias largas de datos, donde el contexto y la memoria a largo plazo son cruciales.

No es un modelo apropiado para este escenario. Los modelos LSTM requieren *grandes cantidades de datos* (decenas de miles o millones de registros) para aprender eficazmente. Con un dataset de solo 930 filas, el modelo no tendría suficientes ejemplos, sufriría de sobreajuste y su rendimiento sería, casi con seguridad, inferior al de un Random Forest.

Se reserva para problemas de secuencias muy largas y complejas, como la traducción de idiomas, el reconocimiento de voz, el análisis de sentimiento en texto o el pronóstico de mercados financieros con datos de alta frecuencia (minuto a minuto).