



UNIVERSIDAD COMPLUTENSE DE MADRID

Facultad de Estudios Estadísticos

Máster big data, data science & inteligencia artificial

MODELIZACIÓN Y PREDICCIÓN DE PRECIOS DE VIVIENDAS DE 3 HABITACIONES - MODELOS DE SERIES TEMPORALES

Tarea de Tarea minería de datos y modelización predictiva

Autor(a): Daniel Olmedo Rawlins Poveda

Profesor(a): Juan Antonio Guevara

Curso académico: 2025-2026

Madrid, 26 de febrero de 2026

Índice

1. Introducción	2
2. Representación gráfica y descomposición	3
3. Transformación logarítmica y diferenciación	4
4. Reserva de datos para validación	6
5. Modelo de Suavizado Exponencial	6
5.1. Aplicación sobre la serie original	7
5.1.1. Suavizado Exponencial Simple (SES)	7
5.1.2. Modelo Holt-Winters	8
5.2. Aplicación sobre la serie transformada	9
5.2.1. Suavizado Exponencial Simple	9
5.2.2. Holt-Winters sobre la serie transformada	9
6. Análisis ACF, PACF y ajuste ARIMA	10
6.1. Análisis de la serie original	10
6.2. Análisis tras diferenciación ($d = 1$)	11
6.3. Modelos candidatos evaluados	12
6.4. Modelo seleccionado	13
7. Expresión algebraica del modelo	14
8. Predicciones e intervalos de confianza	15
9. Comparación de modelos y conclusiones	16
9.1. Criterios de comparación	16
9.2. Resultados	17
9.3. Análisis comparativo	17
9.4. Conclusión	18
10. Repositorio y reproducibilidad del análisis	19
11. Bibliografía	19

1. Introducción

El presente trabajo tiene como objetivo el análisis y modelización de una serie temporal con estacionalidad, cumpliendo el requisito de trabajar con aproximadamente 150 observaciones, tal y como se establece en el enunciado de la práctica [1].

La serie analizada se construye a partir del conjunto de datos *Property Sales Time Series*, disponible en Kaggle [2]. Dicho dataset contiene información sobre transacciones inmobiliarias registradas entre los años 2007 y 2019.

Listing 1: Carga inicial del dataset

```
1 import pandas as pd
2 df = pd.read_csv("/data/raw/raw_sales.csv")
```

A partir de los datos originales diarios, se construye una serie temporal mensual definiendo como variable de interés el **precio medio mensual de venta** de viviendas tipo *house* con tres habitaciones.

Listing 2: Construcción de pandas series

```
1 # 1) Parseo de fecha (ajusta el nombre de columna si no es '
    datesold')
2 df["datesold"] = pd.to_datetime(df["datesold"], errors="coerce")
3
4 # 2) Filtrar a "houses" y 3 bedrooms (ajusta nombres si es distinto
    ) y eliminar postcode
5 df = df[(df["propertyType"].str.lower() == "house") & (df["bedrooms"
    "] == 3)].copy()
6 df.drop(columns='postcode',axis=0)df.drop(columns='postcode',axis
    =0)
7
8 # 3) Agregar a frecuencia mensual (~150 obs) e interpolar
9 serie = (df
10         .set_index("datesold")["price"]
11         .resample("MS")           # Month Start
12         .mean()
13         .asfreq("MS")).interpolate(limit_direction="both")
```

La agregación a frecuencia mensual permite obtener aproximadamente 156 observaciones, lo que cumple el requisito de tamaño muestral.

El objetivo del estudio es:

- Analizar las componentes estructurales de la serie (tendencia y estacionalidad).
- Evaluar su estacionariedad.

- Ajustar un modelo de suavizado exponencial adecuado.
- Identificar y estimar un modelo ARIMA estacional siguiendo la metodología Box-Jenkins [3].
- Comparar la capacidad predictiva de ambos enfoques.

2. Representación gráfica y descomposición

En primer lugar, se representa gráficamente la serie temporal correspondiente al precio medio mensual de venta de viviendas tipo *house* con tres habitaciones.

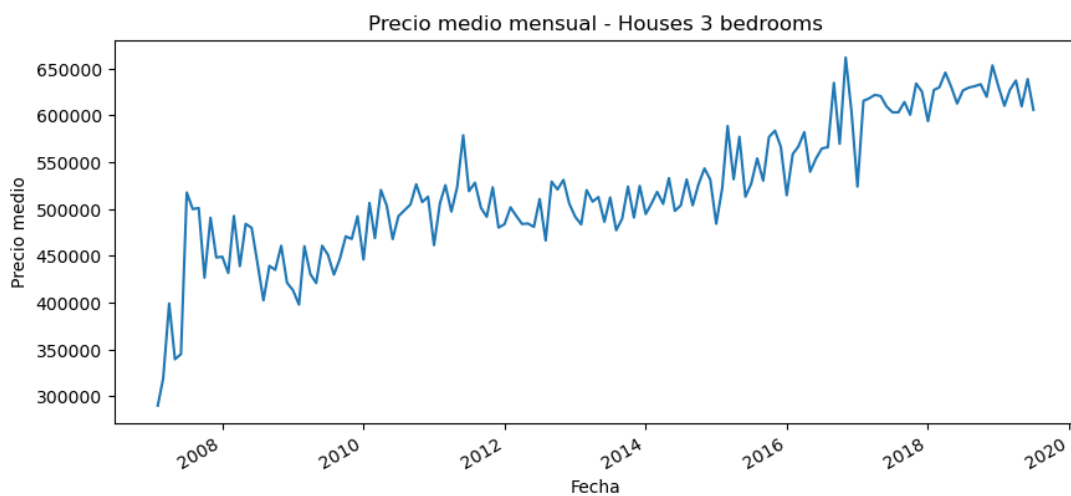


Figura 1: Precio medio mensual de viviendas tipo *house* con tres habitaciones.

Tal como se aprecia en la Figura 1, la serie presenta una clara tendencia creciente a lo largo del periodo 2007–2019, la serie no es estacionaria en media, ya que presenta una tendencia creciente a lo largo del periodo analizado. Se observa igualmente una caída significativa en los años 2008–2009, coincidente con la crisis financiera, seguida de una fase de recuperación progresiva y un crecimiento más pronunciado a partir de 2014.

Dado que la amplitud de las fluctuaciones estacionales parece aumentar ligeramente conforme crece el nivel de la serie, se opta por realizar una descomposición bajo un esquema multiplicativo.

Listing 3: Seasonal Decompose

```
1 from statsmodels.tsa.seasonal import seasonal_decompose
2 result = seasonal_decompose(serie, model='multiplicative', period
    =12)
```

La descomposición permite separar la serie en las siguientes componentes:

- **Tendencia:** muestra un comportamiento claramente creciente, especialmente a partir del año 2014.
- **Estacionalidad:** se observa un patrón repetitivo anual (periodicidad $s = 12$), con coeficientes que oscilan aproximadamente entre 0.95 y 1.03, lo que indica variaciones estacionales moderadas en torno al 5 %.
- **Componente residual:** no presenta patrones sistemáticos evidentes y fluctúa alrededor de la unidad, lo que sugiere que la descomposición captura adecuadamente la estructura principal de la serie.

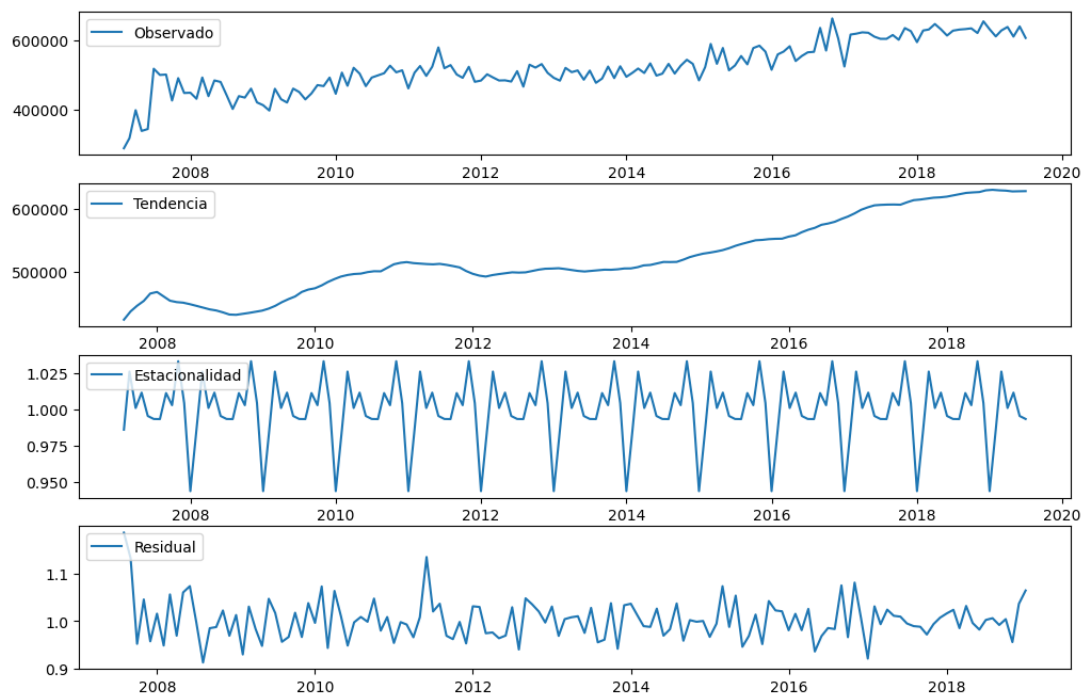


Figura 2: Descomposición estacional multiplicativa del precio medio mensual de viviendas tipo *house* con tres habitaciones (2007–2019).

Como puede observarse en la Figura 2, la serie presenta una tendencia claramente creciente y un patrón estacional anual bien definido. La componente estacional oscila alrededor de la unidad, lo que confirma la idoneidad del esquema multiplicativo.

En consecuencia, se concluye que la serie presenta tendencia creciente y estacionalidad anual, por lo que no es estacionaria en media y requerirá transformaciones posteriores para su modelización mediante técnicas ARIMA.

3. Transformación logarítmica y diferenciación

Con el objetivo de preparar la serie para su posterior modelización mediante técnicas ARIMA, se aplicó una transformación compuesta consistente en:

1. Transformación logarítmica, con el fin de estabilizar la varianza.

Listing 4: Transformación Logarítmica

```
1 # 1) Transformacion Logaritmica
2 serie_log = np.log(serie)
```

2. Diferenciación de primer orden, para eliminar la tendencia y aproximar la estacionariedad en media.

Listing 5: Diferenciación de (1)

```
1 # 2) Diferencia orden 1
2 serie_log_diff = serie_log.diff(1).dropna()
```

3. Desplazamiento aditivo (shift) posterior a la diferenciación, con el propósito de evitar valores negativos en la nueva serie.

Listing 6: Shift para evitar valores negativos

```
1 # 3) Shift para no-negatividad
2 min_diff = float(serie_log_diff.min())
3 shift = (-min_diff) + eps if min_diff <= 0 else 0.0
4 serie_ready = serie_log_diff + shift
```

La Figura 3 muestra la descomposición estacional de la serie transformada.

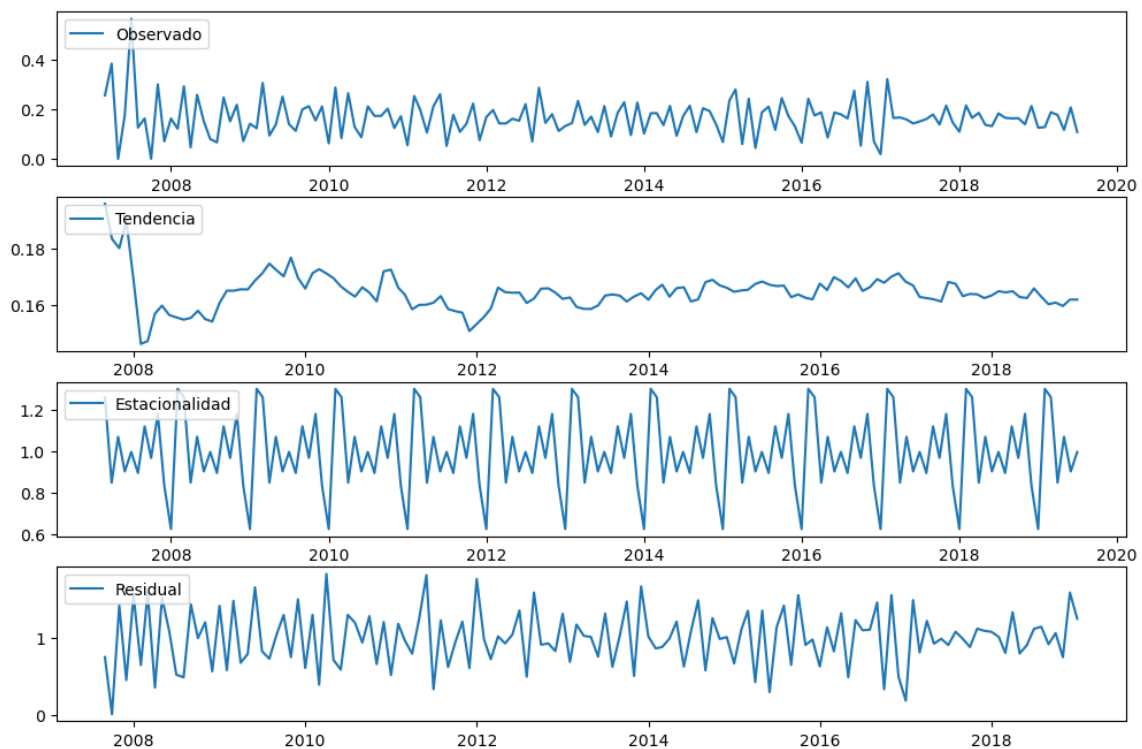


Figura 3: Descomposición estacional de la serie transformada (log + diferencia de primer orden + ajuste aditivo).

La transformación logarítmica aplicada a la serie reduce significativamente la tendencia creciente observada en los precios originales de las viviendas de 3 habitaciones, generando un comportamiento más estable en media.

Si bien la diferenciación permite reforzar la estacionariedad, en este caso su impacto sobre la estructura de la serie no resulta sustancial, ya que la transformación logarítmica consigue estabilizar adecuadamente la varianza y suavizar la tendencia sin alterar de forma significativa la dinámica subyacente.

Asimismo, la componente estacional mantiene un patrón anual identificable, lo que sugiere que la serie conserva su estructura temporal relevante sin necesidad de aplicar transformaciones adicionales para el análisis comparativo.

Por tanto, para la estimación y comparación de los modelos predictivos, se utilizará exclusivamente la serie transformada en logaritmos, al proporcionar una representación suficientemente estable y adecuada para el modelado sin introducir complejidad innecesaria mediante diferenciaciones adicionales.

4. Reserva de datos para validación

Se reservaron las últimas 12 observaciones de la serie (equivalentes a un ciclo estacional completo, dado que la periodicidad es mensual con $s = 12$).

Listing 7: Extracción de los últimos 12 meses

```
1 s = 12
2 train = serie_log.iloc[:-s]
3 test = serie_log.iloc[-s:]
```

El objetivo de esta partición es:

- Ajustar los modelos únicamente sobre el conjunto de entrenamiento.
- Evaluar la capacidad predictiva sobre datos no utilizados en la estimación.
- Comparar el rendimiento de distintos métodos (Holt-Winters y SARIMA) de manera objetiva.

Este procedimiento evita el sobreajuste y permite medir el error de predicción real del modelo.

5. Modelo de Suavizado Exponencial

En esta sección se evalúa el comportamiento de los modelos de suavizado exponencial tanto sobre la serie original como sobre la serie transformada. El objetivo es analizar si los

modelos deterministas basados en nivel, tendencia y estacionalidad son adecuados para capturar la dinámica observada.

5.1. Aplicación sobre la serie original

La serie original presenta una tendencia creciente clara y un patrón estacional anual (periodicidad $s = 12$). Por tanto, se considera apropiado aplicar modelos de suavizado exponencial.

5.1.1. Suavizado Exponencial Simple (SES)

El modelo de suavizado exponencial simple se define como:

$$\hat{X}_{t+1} = \alpha X_t + (1 - \alpha) \hat{X}_t$$

donde $\alpha \in (0, 1)$ es el parámetro de suavizado.

Este modelo es adecuado para series sin tendencia ni estacionalidad.

Listing 8: Suavizado Simple Exponencial

```
1 modelo_ses = sm.tsa.SimpleExpSmoothing(  
2     train,  
3     initialization_method="estimated"  
4 ).fit()
```

En el caso de la serie original, se utiliza como modelo base de comparación. Notando que la serie en la predicción se vuelve estacionaria y no se adapta a la tendencia.

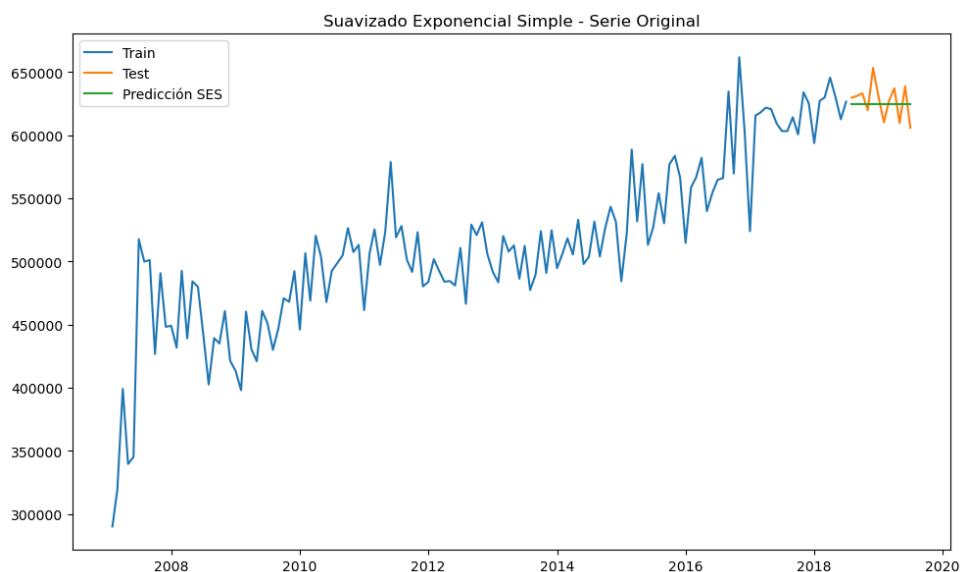


Figura 4: Predicción mediante Suavizado Exponencial Simple sobre la serie original.

5.1.2. Modelo Holt-Winters

Dado que la serie original presenta tendencia y estacionalidad, se considera el modelo de Holt-Winters (suavizado exponencial triple).

Listing 9: Modelo Holt-Winters

```
1 modelo_hw = sm.tsa.ExponentialSmoothing(  
2     train ,  
3     trend=None ,  
4     seasonal='add' ,  
5     seasonal_periods=12 ,  
6     initialization_method="estimated" ,  
7     freq='MS' ,  
8 ).fit()
```

En el caso aditivo:

$$X_t = L_t + b_t + S_t + e_t$$

donde:

- L_t es el nivel en el instante t ,
- S_t es la componente estacional aditiva,
- e_t es el término de error aleatorio.

Se estiman los parámetros α , β y γ mediante optimización automática y se generan predicciones para un horizonte de 12 meses.

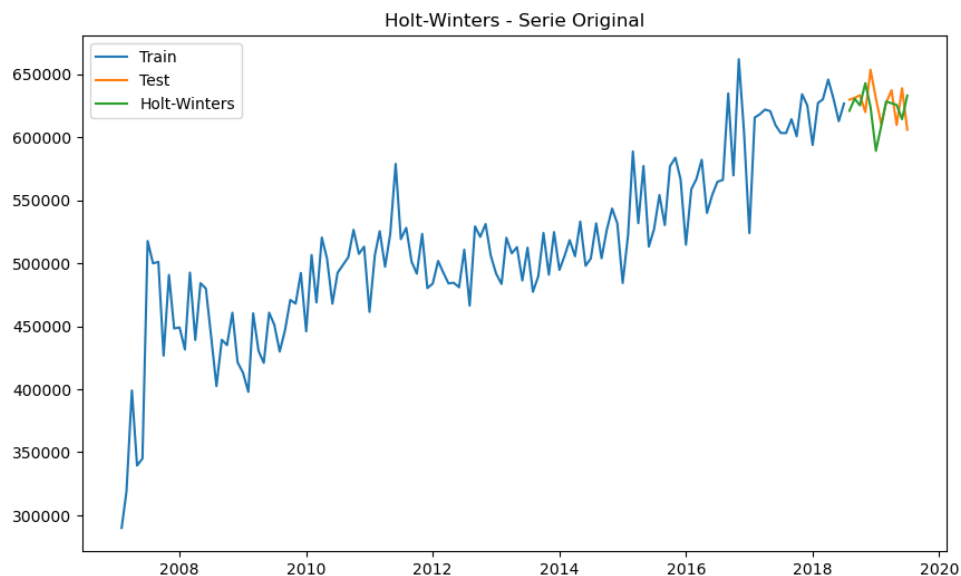


Figura 5: Predicción mediante Holt-Winters sobre la serie original.

Los resultados obtenidos muestran que el modelo Holt-Winters no mejora el rendimiento frente al suavizado simple, lo que sugiere que la estructura estacional no es lo suficientemente estable como para ser capturada eficazmente por un esquema determinista.

5.2. Aplicación sobre la serie transformada

Posteriormente, se analiza la serie transformada en la sección anterior, presentando la serie un comportamiento más cercano a un proceso estacionario.

5.2.1. Suavizado Exponencial Simple

Se aplica nuevamente el modelo SES sobre la serie transformada, dado que ahora no se observa tendencia determinista clara. A partir de los resultados que observamos en la siguiente imagen sigue teniendo un componente estacionario pero no se ajusta a la variabilidad de la transformación.

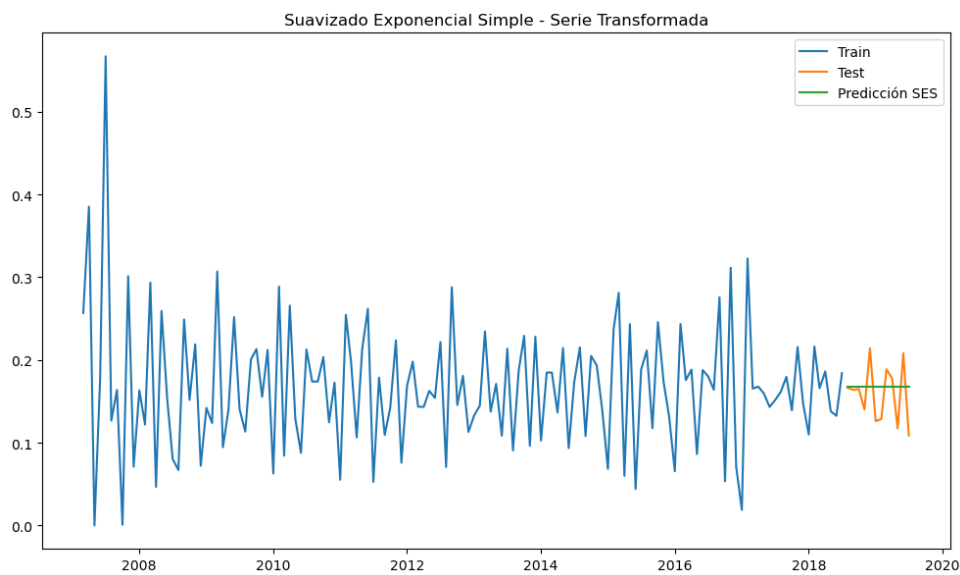


Figura 6: Predicción mediante Suavizado Exponencial Simple sobre la serie transformada.

5.2.2. Holt-Winters sobre la serie transformada

También se prueba el modelo Holt-Winters con componente estacional aditiva y periodicidad anual ($s = 12$). En este apartado al igual que con la serie original no mejora los resultados.

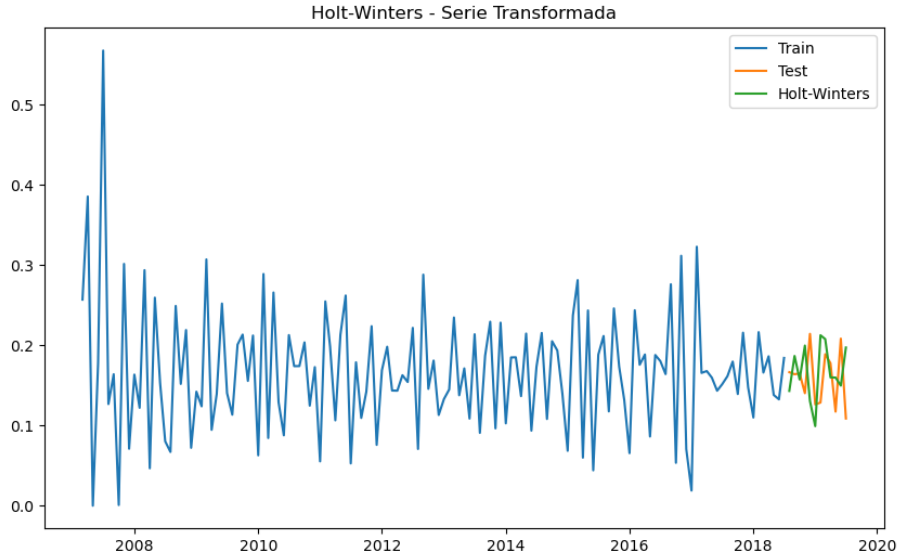


Figura 7: Predicción mediante Holt-Winters sobre la serie transformada.

Los resultados muestran que, tras la transformación, la estructura determinista (tendencia + estacionalidad) se ha reducido considerablemente, por lo que el modelo Holt-Winters no aporta mejoras frente al suavizado simple.

Esto sugiere que la dependencia restante en la serie es de naturaleza autoregresiva, lo que motiva la aplicación de modelos ARIMA en la siguiente sección.

6. Análisis ACF, PACF y ajuste ARIMA

Siguiendo la metodología Box-Jenkins [3]:

1. Se analiza la estacionariedad de la serie.
2. Se aplican diferenciaciones si procede.
3. Se estudian las funciones de autocorrelación (ACF) y autocorrelación parcial (PACF).
4. Se determinan los órdenes (p, d, q) y $(P, D, Q)_s$.

El modelo general considerado es:

$$ARIMA(p, d, q)(P, D, Q)_{12}$$

6.1. Análisis de la serie original

Como primer punto a considerar la serie original el único tratamiento anterior que se le va a realizar es el de transformarlo logaritmicamente para reducir la varianza.

Listing 10: Análisis ACF y PACF

```

1 from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
2 fig, ax = plt.subplots(2, 1, figsize=(10, 6), sharex=True)
3 plot_acf(train, lags=48, alpha=0.05, ax=ax[0])
4 ax[0].set_title("ACF - Serie Logaritmica (train)")
5 plot_pacf(train, lags=48, alpha=0.05, ax=ax[1], method="ywm")
6 ax[1].set_title("PACF - Serie Logaritmica (train)")
7 plt.tight_layout()
8 plt.show()

```

El análisis inicial de la ACF de la serie original transformada logarítmicamente demostró un decrecimiento lento y persistente de las autocorrelaciones, lo cual es indicativo de no estacionariedad en media. La PACF presentó un pico significativo en el primer retardo, pero la ausencia de corte claro en la ACF confirmó la presencia de raíz unitaria.

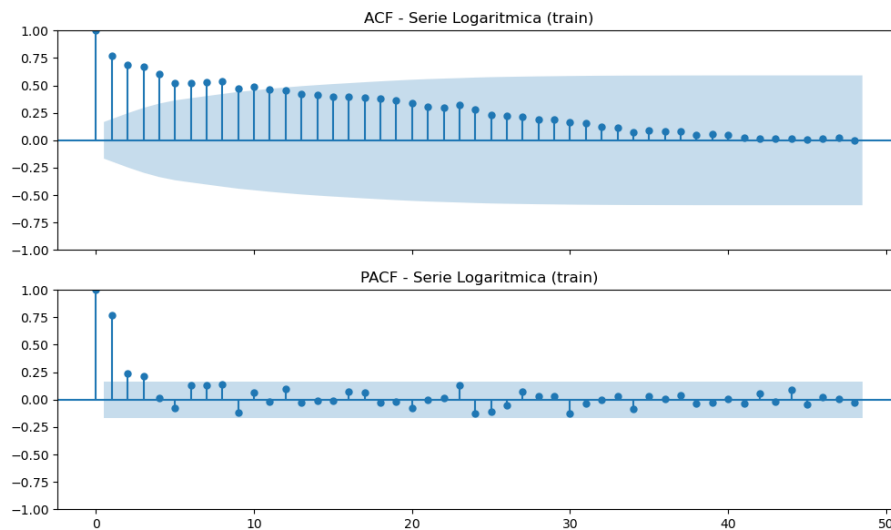


Figura 8: ACF-PACF sobre la serie transformada.

Por tanto, se estableció la necesidad de aplicar una diferenciación de primer orden, fijando:

$$d = 1$$

6.2. Análisis tras diferenciación ($d = 1$)

Una vez diferenciada la serie, la ACF mostró un corte claro en el primer retardo con autocorrelación negativa significativa, mientras que la PACF no presentó estructura persistente.

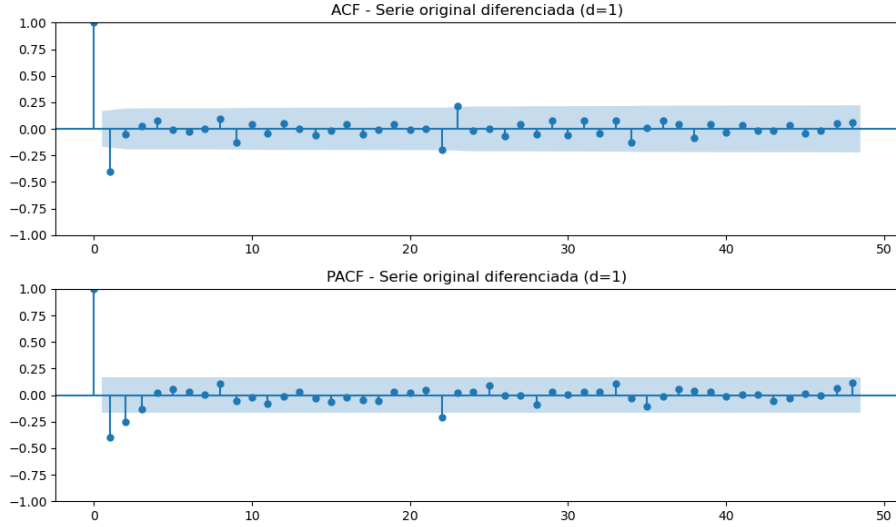


Figura 9: ACF-PACF sobre la serie transformada y Diferenciada de primer orden.

Este patrón es característico de un modelo MA(1), por lo que se propuso inicialmente:

$$ARIMA(0, 1, 1)$$

6.3. Modelos candidatos evaluados

Durante el proceso de identificación se estimaron distintos modelos ARIMA/SARIMA con el objetivo de evaluar su ajuste, significancia de parámetros, diagnóstico de residuos y capacidad predictiva.

Modelo	AIC	BIC	LogLik	Par. Sig.	Ljung-Box p
ARIMA(0,0,1)	-122.59	-116.78	63.30	1/1	0.00
ARIMA(0,0,2)	-161.42	-152.73	83.71	2/2	0.00
ARIMA(1,0,1)	-293.62	-284.91	149.81	1/2	0.00
ARIMA(1,0,2)	-342.20	-330.62	175.10	1/3	0.09
ARIMA(0,1,1)	-349.56	-343.75	176.78	1/1	0.89

Cuadro 1: Comparación inicial de modelos ARIMA candidatos

Se utilizaron como criterios de selección el AIC y BIC (parsimonia), el Log-Likelihood (calidad de ajuste), la significancia de los parámetros, el contraste de Ljung-Box (incorrección residual) y las métricas MAE y RMSE para evaluar capacidad predictiva.

Los modelos con $d = 0$ presentaron residuos autocorrelacionados, mientras que algunas especificaciones mostraron problemas de identificación. Dentro del conjunto evaluado, el modelo **ARIMA(0,1,1)** obtuvo el menor AIC y superó el contraste de Ljung-Box ($p = 0,89$), confirmando la ausencia de autocorrelación residual.

Adicionalmente, se realizó un *grid search* manual sobre modelos SARIMA (probando múltiples combinaciones de (p, d, q) y $(P, D, Q)_{12}$ y distintas opciones de tendencia), almacenando los resultados en un archivo externo (`SARIMA_GRID.csv`). En el cuadro 2 se muestran los **5 mejores modelos** (según AIC) obtenidos en dicha búsqueda.

Modelo	Trend	AIC	BIC	LogLik	Ljung-Box p(12)	Conv.
SARIMA(0,1,1)(1,0,1,12)	c	-383.80	-369.74	196.90	0.9999	Sí
SARIMA(0,1,1)(1,0,0,12)	c	-382.63	-371.32	195.32	1.0000	No
SARIMA(1,1,1)(1,0,1,12)	c	-382.51	-365.64	197.25	0.9996	Sí
SARIMA(1,0,1)(0,0,1,12)	n	-381.41	-370.13	194.71	1.0000	Sí
SARIMA(0,1,1)(0,0,1,12)	c	-381.20	-369.95	194.60	1.0000	Sí

Cuadro 2: Top-5 modelos SARIMA del grid search (extraídos de `SARIMA_GRID.csv`)

Nota: el segundo modelo en AIC presenta *no convergencia* (Conv. = No), por lo que, aunque aparece en el ranking, **no se considera candidato final** sin resolver primero el problema de ajuste.

6.4. Modelo seleccionado

Listing 11: Modelo SARIMA con mejor rendimiento

```

1 model = SARIMAX(
2     train,
3     order=(0, 1, 1),
4     seasonal_order=(1, 0, 1, 12),
5     trend="c",
6     enforce_stationarity=False,
7     enforce_invertibility=False,
8     freq= 'MS'
9 )
10 res = model.fit(dispatch=False, maxiter=500)

```

Tras la exploración sistemática mediante *grid search* y el análisis comparativo de criterios de información y diagnóstico residual, el modelo finalmente seleccionado es:

SARIMA(0, 1, 1)(1, 0, 1)₁₂ con *trend* = *c*

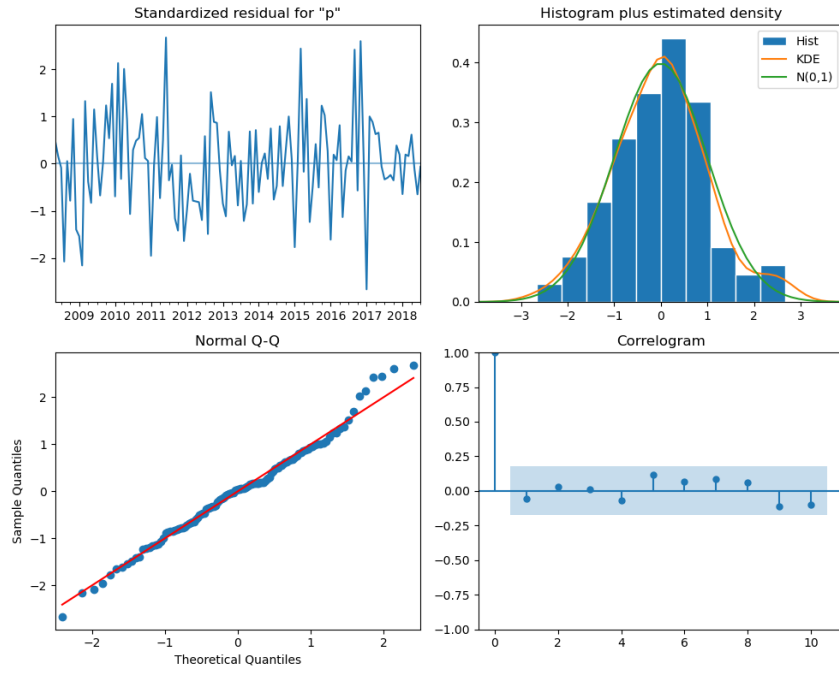


Figura 10: Diagnostico estadístico de serie transformada.

Listing 12: Suavizado Simple Exponencial

```

1 fig = res.plot_diagnostics(figsize=(10, 8))
2 fig.tight_layout()
3 fig.show()

```

Este modelo incorpora:

- Un componente de media móvil no estacional $MA(1)$.
- Una diferenciación regular de orden $d = 1$ para capturar la tendencia.
- Componentes autorregresivo y de media móvil estacionales ($P = 1, Q = 1$) con periodicidad mensual $s = 12$.
- Un término constante (*drift*) mediante $trend = c$.

7. Expresión algebraica del modelo

El modelo seleccionado es:

$$SARIMA(0, 1, 1)(1, 0, 1)_{12}$$

con término constante (intercepto).

En notación mediante el operador de retardo B , el modelo puede expresarse como:

$$(1 - \Phi_1 B^{12})(1 - B)X_t = c + (1 + \theta_1 B)(1 + \Theta_1 B^{12}) \varepsilon_t$$

donde:

- B es el operador de retardo,
- c es el término constante,
- ε_t es un proceso de ruido blanco,
- Φ_1 es el parámetro autorregresivo estacional,
- θ_1 es el parámetro de medias móviles no estacional,
- Θ_1 es el parámetro de medias móviles estacional.

Sustituyendo los valores estimados:

$$(1 + 0,1826B^{12})(1 - B)X_t = 0,0032 + (1 - 0,7908B)(1 + 0,3102B^{12}) \varepsilon_t$$

8. Predicciones e intervalos de confianza

Una vez validado el modelo SARIMA(0, 1, 1)(1, 0, 1)₁₂ con término constante, se generó predicciones para los 12 meses siguientes obteniendo mediante el estimador de máxima verosimilitud del modelo ajustado, incorporando intervalos de confianza al 95 %, calculados como:

$$\hat{y}_{t+h} \pm 1,96 \cdot \sqrt{\text{Var}(\hat{y}_{t+h})}$$

donde la varianza de predicción incrementa con respecto el horizonte temporal h .

Listing 13: Graficando intervalos de confianza

```

1 # Grafico
2 plt.figure(figsize=(10,6))
3 plt.plot(train, label="Train")
4 plt.plot(test, label="Test")
5 plt.plot(pred, label="SARIMAX")
6 plt.fill_between(conf_int.index, conf_int.iloc[:, 0], conf_int.iloc
   [:, 1], color="blue", alpha=0.2, label="IC 95%")
7 plt.legend()
8 plt.title("SARIMAX - Serie Logaritmica")
9 plt.show()

```


Representación gráfica

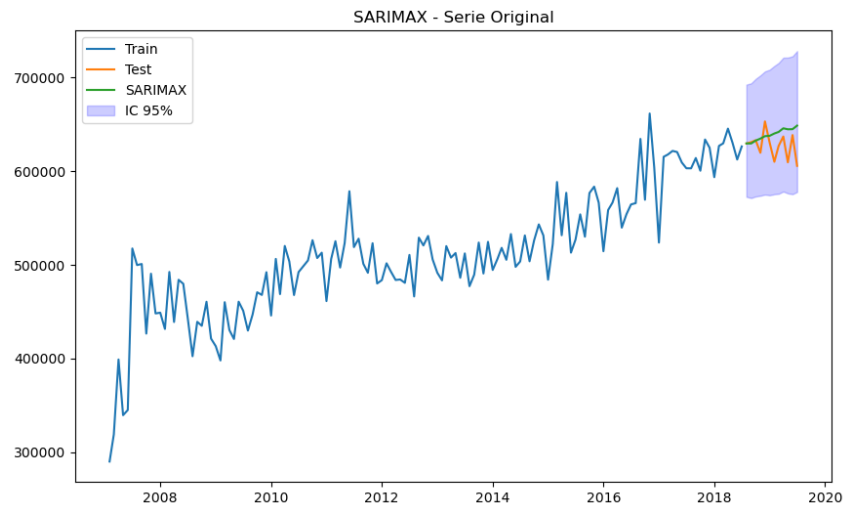


Figura 11: Predicción de serie original con intervalos de confianza.

En la Figura 11 se muestran la serie de entrenamiento, los valores reales fuera de muestra, la predicción puntual y los intervalos de confianza al 95 %.

La trayectoria estimada mantiene la tendencia creciente capturada por el término de *drift* y la estructura estacional $(1, 0, 1)_{12}$, observándose que los valores del conjunto de test se sitúan dentro de los intervalos estimados, lo que indica una adecuada calibración probabilística del modelo.

Las predicciones iniciales se sitúan en torno a 630 000–650 000 unidades, con intervalos de confianza de aproximadamente 110 000–120 000 unidades, reflejando la variabilidad inherente al proceso.

9. Comparación de modelos y conclusiones

Con el objetivo de determinar el modelo con mejor desempeño predictivo, se comparan los modelos seleccionados, aplicados al dataset original:

- Modelo ARIMA (ganador según AIC/BIC inicial)
- Modelo de Suavizado Exponencial Simple (SES)
- Modelo Holt-Winters (ganador entre los exponenciales)

9.1. Criterios de comparación

La evaluación se realiza mediante:

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)
- MAPE (Mean Absolute Percentage Error)
- AIC (Akaike Information Criterion)
- BIC (Bayesian Information Criterion)

El MAE y RMSE miden la magnitud del error de predicción. El RMSE penaliza más fuertemente los errores grandes. El MAPE permite interpretar el error en términos porcentuales. Los criterios AIC y BIC penalizan la complejidad del modelo, favoreciendo modelos parsimoniosos.

9.2. Resultados

Modelo	MAE	RMSE	MAPE (%)	AIC	BIC
ARIMA	14894.47	20170.72	2.41	-383.80	-369.74
SES	11397.41	13450.69	1.81	-733.62	-727.77
Holt-Winters	17617.33	22135.17	2.80	-736.39	-695.41

Cuadro 3: Comparación de métricas de desempeño en escala original

9.3. Análisis comparativo

Se observa que el modelo SES presenta los menores valores de MAE, RMSE y MAPE, lo que indica una mayor capacidad predictiva fuera de muestra en términos absolutos para el precio de venta de viviendas de 3 habitaciones.

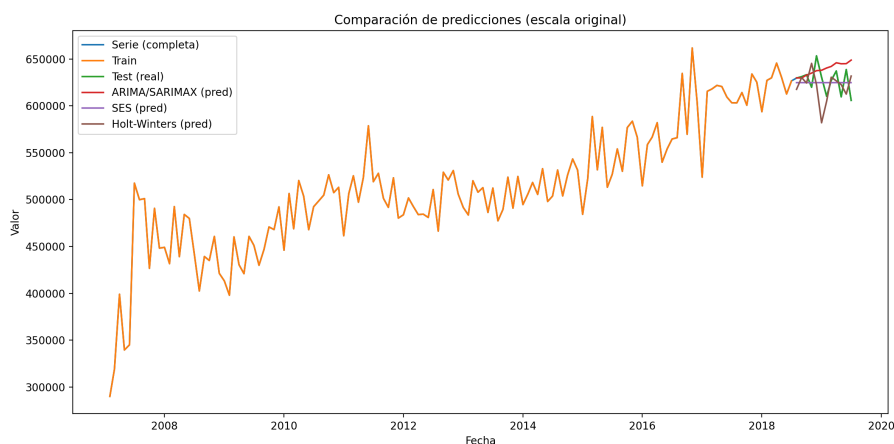


Figura 12: Comparación predictiva de los modelos.

En términos prácticos, esto implica que, en el horizonte evaluado, SES es el modelo que comete menor error al estimar el valor futuro de este tipo de inmueble, lo cual puede resultar útil para estimaciones de corto plazo en el mercado inmobiliario.

Por su parte, el modelo Holt-Winters presenta el menor valor de AIC, mientras que SES obtiene el menor BIC, lo que sugiere un buen equilibrio entre ajuste y complejidad en ambos modelos exponenciales.

No obstante, es importante destacar que el modelo SES genera predicciones constantes para horizontes mayores a un periodo, lo que implica un comportamiento similar a un *random walk*. En el contexto de los precios de viviendas de 3 habitaciones, esto supone asumir que el valor futuro se mantendrá cercano al último nivel observado, sin incorporar de manera explícita una tendencia de crecimiento o desaceleración del mercado.

Aunque este modelo minimiza el error en el horizonte evaluado, no captura explícitamente posibles dinámicas estructurales del mercado, como tendencias alcistas sostenidas, cambios de ciclo o efectos estacionales relacionados con la actividad inmobiliaria.

9.4. Conclusión

Tras la comparación de métricas, el modelo **SES** presenta el menor error de predicción en el horizonte evaluado, por lo que puede considerarse el mejor modelo desde un enfoque estrictamente predictivo a corto plazo para el precio de venta de viviendas de 3 habitaciones.

En términos prácticos, esto implica que, para estimaciones inmediatas de valor de mercado por ejemplo, en procesos de tasación rápida, fijación de precios de salida o estimaciones preliminares para agencias inmobiliarias, el modelo SES ofrece la mayor precisión dentro del periodo analizado.

No obstante, dado que SES no incorpora una tendencia explícita y genera pronósticos constantes para horizontes mayores a un periodo, su capacidad para capturar dinámicas estructurales del mercado inmobiliario es limitada. En contextos donde puedan producirse cambios de ciclo, aceleraciones en los precios o ajustes derivados de condiciones macroeconómicas, este comportamiento tipo *random walk* puede resultar insuficiente.

Por su parte, el modelo ARIMA, aunque presenta un mayor error predictivo en el horizonte evaluado, modela explícitamente la estructura temporal de la serie, permitiendo capturar relaciones dinámicas propias del mercado de la vivienda, como persistencia en los precios o ajustes graduales.

En consecuencia:

- Si el objetivo es minimizar el error de predicción a corto plazo en el precio de venta de viviendas de 3 habitaciones, se selecciona el modelo SES.
- Si el objetivo es analizar la dinámica estructural del mercado inmobiliario y comprender la evolución temporal de los precios, el modelo ARIMA resulta más consistente

metodológicamente.

Finalmente, es importante señalar que los modelos utilizados se basan exclusivamente en la información histórica de la propia serie temporal. Para mejorar el rendimiento predictivo en futuras investigaciones, podría incorporarse un modelo con variables exógenas (ARIMAX o SARIMAX)

10. Repositorio y reproducibilidad del análisis

Con el objetivo de garantizar la reproducibilidad del estudio y permitir la realización de pruebas adicionales, todo el código utilizado en este análisis se encuentra disponible en el siguiente repositorio de GitHub:

https://github.com/danielrawlins21/series_temporales_analisis_prediccion

El repositorio incluye:

- Scripts completos de entrenamiento y comparación de modelos.
- Implementación de modelos ARIMA, SES y Holt-Winters.
- Generación automática de métricas (MAE, RMSE, MAPE, AIC, BIC).
- Visualización gráfica de predicciones.
- Estructura modular que permite incorporar nuevas variables o modelos.

11. Bibliografía

Referencias

- [1] Ejercicio de Evaluación: Análisis y Predicción de Series Temporales. Máster en Big Data y Business Analytics [1].
- [2] HTAG Holdings, *Property Sales Time Series Dataset*, Kaggle. Disponible en: https://www.kaggle.com/datasets/htagholdings/property-sales?select=raw_sales.csv
- [3] Metodología para aplicar un modelo ARIMA estacional [3].
- [4] Guevara Gil, J.A., *Series Temporales – Clase 1 y 2*. Máster en Big Data e Inteligencia Artificial [4].