

ANÁLISIS DE SERIES TEMPORALES Y OPTIMIZACIÓN DE MODELOS MEDIANTE ALGORITMOS DE OPTIMIZACIÓN

INTRODUCCIÓN A LA IA

ENTREGA FINAL

Prof. Yomin Jaramillo Múnera

Universidad EAFIT

AUTORES:

Yeniffer Andrea Córdoba

Pablo Gómez Mutis

Luz Adriana Yepes

INDICE

1. RESUMEN	4
2. INTRODUCCIÓN Y ANTECEDENTES	5
2.1. Presentación del problema	5
2.2. Importancia del problema	5
3. DEFINICIÓN DEL PROBLEMA	7
3.1. Contexto	7
3.2. Restricciones	7
3.3. Objetivos	8
4. METODOLOGÍA Y RESULTADOS	9
4.1. DESCRIPCIÓN Y JUSTIFICACIÓN DE LOS ALGORITMOS	9
4.1.1. Modelos de series de tiempo	9
4.1.2. Algoritmos de optimización	10
4.2. ARIMA	11
4.2.1. Modelo original	11
4.2.2. Hill Climbing	13
4.2.3. Algoritmo Genético	14
4.3. VAR	15
4.3.1. Modelo original	15
4.3.2. Hill Climbing	17
4.3.3. Algoritmo Genético	18
4.4. Random Forest	19
4.4.1. Modelo original	19
4.4.2. Hill Climbing	20
4.4.3. Algoritmo Genético	20
4. CONCLUSIONES Y POSIBLES MEJORAS	21
4.1. Modelo ARIMA	21
4.2. Modelo VAR	22
4.3. Modelo Random Forest	23
4.4. Comparación General de los Algoritmos de Optimización	24

1. RESUMEN

Este proyecto se enfocará en la implementación de **algoritmos de optimización aplicados a series de tiempo**, específicamente a una serie de tiempo que refleja la variación del precio de la vivienda a lo largo de doce años (2007-2019). Este análisis tomará en cuenta diversas variables, como el número de habitaciones y el tipo de vivienda (casa o apartamento).

Para el modelamiento de las series de tiempo, utilizaremos dos modelos estadísticos tradicionales: **ARIMA** (univariado) y **VAR** (multivariado). Además, con fines comparativos, incluiremos un modelo de *machine learning*: **Random Forest**. Cada uno de estos modelos será inicialmente ejecutado con sus parámetros estándar.

Posteriormente, aplicaremos dos algoritmos de optimización estudiados en clase: **Hill Climbing** (búsqueda local) y **Algoritmo Genético** (optimización evolutiva). El mejoramiento en el rendimiento de cada modelo se presentará gráficamente, y se proporcionarán los valores originales y optimizados de MSE (Error Cuadrático Medio) y RMSE (Raíz del Error Cuadrático Medio).

Finalmente, se realizará una comparación del rendimiento de todos los modelos de series de tiempo optimizados para identificar la opción más adecuada.

2. INTRODUCCIÓN Y ANTECEDENTES

2.1. Presentación del problema

El problema principal consiste en aplicar algoritmos de optimización a series de tiempo para desarrollar un sistema más preciso de predicción del precio de la vivienda.

El dataset utilizado refleja la variación del precio medio de la vivienda en una región específica durante el período comprendido entre 2007 y 2019. Incluye las siguientes variables:

- Fecha.
- Precio medio: Esta será la variable objetivo en los modelos.
- Número de habitaciones: 1, 2, 3, 4, 5.
- Tipo de vivienda: Casa o apartamento.

El análisis se centrará en métodos estadísticos tradicionales para series de tiempo: ARIMA (univariado) y VAR (multivariado). Sin embargo, también se incluirá un modelo de Machine Learning: Random Forest, con el propósito de realizar una comparación más completa. Aunque la intención inicial era no utilizar este tipo de modelos debido a que ya fueron practicados en clase, su inclusión permitirá evaluar cómo se comportan frente a los algoritmos de optimización en comparación con los modelos estadísticos tradicionales.

Los algoritmos de optimización seleccionados son Hill Climbing y el Algoritmo Genético, ya que en ejercicios previos demostraron ser los más efectivos. Se calcularán las métricas RMSE (Raíz del Error Cuadrático Medio) y MSE (Error Cuadrático Medio) después de aplicar cada algoritmo, con el objetivo de cuantificar la mejora obtenida.

2.2. Importancia del problema

Decidimos enfocarnos en este problema debido a la dificultad recurrente que hemos encontrado al aplicar modelos y algoritmos en distintas clases, especialmente cuando se trata de mejorar su rendimiento. La optimización es fundamental para garantizar que modelos estadísticos y de machine learning proporcionen predicciones precisas, y la falta de este proceso puede llevar a resultados inadecuados o incompletos.

Este proyecto nos permite profundizar en el uso de algoritmos de optimización como Hill Climbing y Algoritmos Genéticos, aplicándolos tanto a modelos estadísticos tradicionales (ARIMA y VAR) como a un modelo de machine learning (Random Forest).

Estos métodos ya demostraron ser efectivos en ejercicios previos, pero queremos evaluar su desempeño de manera más sistemática en el contexto de series de tiempo. Elegimos trabajar con series de tiempo porque nuestros futuros proyectos de grado estarán relacionados con este tema, y consideramos que desarrollar un enfoque sólido para optimizar modelos predictivos será de gran valor para investigaciones posteriores. Además, comparar métodos estadísticos y de machine learning optimizados proporcionará un marco claro para identificar las técnicas más adecuadas según sus características y rendimiento.

3. DEFINICIÓN DEL PROBLEMA

3.1. Contexto

El problema seleccionado consiste en la predicción de la variable 'MA' (Media Anual), un indicador numérico que se pretende estimar con precisión a partir de un conjunto de datos históricos. Este conjunto incluye variables relevantes como:

- 'saledate': Fecha de registro, esencial para capturar la estacionalidad y las tendencias temporales.
- 'type': Tipo de propiedad, categorizado y transformado mediante variables dummies para ser procesado por modelos de aprendizaje automático.
- 'bedrooms': Número de habitaciones, que influye en la predicción de la variable objetivo.

Este problema se aborda mediante un enfoque integral que combina varios modelos de predicción y técnicas de optimización, permitiendo una comparación de sus desempeños para seleccionar el mejor método. Los principales métodos implementados son:

- **Modelos clásicos de series de tiempo:** ARIMA y VAR, que aprovechan la estructura temporal de los datos para realizar predicciones.
- **Modelos basados en aprendizaje automático:** Random Forest, optimizado mediante algoritmos de búsqueda metaheurística como Hill Climbing y Algoritmos Genéticos para mejorar su precisión.

3.2. Restricciones

Preservación de la columna 'saledate': Dado que algunos modelos requieren la variable temporal para capturar patrones y tendencias (y que otros modelos a veces la convierten en el índice) esta columna debe mantenerse durante la preparación de datos. Para esto en ciertos modelos creamos copias del dataframe para que así no eliminaran la columna “saledate” de los datos.

Compatibilidad de algoritmos: Los métodos de optimización deben aplicarse sobre versiones del dataset que mantengan intacta la estructura relevante para cada modelo. Esto es porque, después de aplicar cada modelo, el dataframe original podría verse afectado (el típico caso apenas mencionado que eliminan la columna de las

fechas por haberla convertido en índice). Por lo tanto, es muy importante revisar el estado del dataframe antes de aplicar cualquier algoritmo o método nuevo.

Consumo de recursos: La implementación de técnicas como Random Forest y optimización metaheurística debe ser eficiente en términos de memoria y tiempo de ejecución.

Consistencia en la comparación: Los resultados obtenidos por diferentes modelos deben ser comparables, utilizando métricas comunes como MSE (Mean Squared Error), RMSE (Root Mean Squared Error) y MAE (Mean Absolute Error).

3.3. Objetivos

Construcción de Modelos Base:

- Implementar modelos clásicos como ARIMA y VAR, aprovechando la estructura temporal y las relaciones entre variables para predecir la serie 'MA'.
- Implementar un modelo basado en Random Forest Regressor que considere tanto variables numéricas como categóricas.

Optimización del Modelo Random Forest:

- Aplicar Hill Climbing con múltiples iteraciones para mejorar la precisión del modelo al ajustar hiperparámetros importantes.
- Implementar Algoritmos Genéticos para explorar múltiples configuraciones de hiperparámetros y seleccionar la mejor combinación.

Comparación y Evaluación:

- Comparar los modelos en términos de MSE, RMSE y MAE sobre un conjunto de prueba común.
- Evaluar la efectividad de las técnicas de optimización aplicadas al modelo Random Forest en relación con los modelos clásicos ARIMA y VAR.

Seleccionar el Mejor Modelo:

- Identificar el modelo con menor error y mejor capacidad de generalización para futuras predicciones.
- Generar un sistema robusto que permita realizar predicciones confiables de la variable 'MA' considerando tanto relaciones temporales como multivariantes.

4. METODOLOGÍA Y RESULTADOS

4.1. DESCRIPCIÓN Y JUSTIFICACIÓN DE LOS ALGORITMOS

4.1.1. Modelos de series de tiempo

ARIMA (Autoregressive Integrated Moving Average):

El modelo ARIMA es uno de los enfoques más populares y efectivos para la predicción de series de tiempo univariadas. Este modelo se basa en tres componentes: autoregresión (AR), diferenciación (I) y media móvil (MA). ARIMA es ideal para modelar series de tiempo con dependencia temporal y patrones de tendencia o estacionalidad. En este proyecto, ARIMA es una excelente opción porque nos permite modelar de manera eficiente la variación del precio de la vivienda a lo largo de los años, utilizando solo los datos históricos de precios, lo cual es clave cuando se trabaja con series de tiempo univariadas.

VAR (Vector Autoregressive):

El modelo VAR es un enfoque multivariado que permite modelar y predecir series de tiempo en las que múltiples variables influyen entre sí. Este modelo es particularmente útil cuando se tienen varias variables que interactúan entre sí, como el número de habitaciones o el tipo de vivienda en relación con el precio de la vivienda. Al aplicar VAR, podemos capturar la dinámica entre estas variables, lo que puede mejorar la precisión de las predicciones al considerar las interacciones entre las variables relevantes en el precio de la vivienda.

Random Forest:

Random Forest es un modelo de machine learning basado en un conjunto de árboles de decisión, lo que permite capturar relaciones no lineales y complejas en los datos. Aunque originalmente no planeábamos usar este tipo de modelo debido a su complejidad y que es *Machine Learning*, decidimos incluir Random Forest para realizar una comparación con los modelos estadísticos tradicionales, como ARIMA y VAR. Su capacidad para manejar grandes volúmenes de datos y su robustez frente a datos ruidosos lo convierten en una excelente opción para abordar la predicción del precio de la vivienda, especialmente cuando se consideran múltiples variables.

4.1.2. Algoritmos de optimización

Hill Climbing:

El algoritmo Hill Climbing es ideal para optimizar modelos en espacios de búsqueda complejos, realizando ajustes iterativos en los parámetros para mejorar la precisión de las predicciones. Su simplicidad y eficiencia lo hacen adecuado para optimizar modelos de series de tiempo de manera local y sin una carga computacional excesiva.

Algoritmo Genético:

El Algoritmo Genético es útil para explorar grandes espacios de parámetros de manera global, evitando óptimos locales y mejorando la precisión de los modelos. Su capacidad para encontrar soluciones robustas lo convierte en una excelente opción para optimizar modelos de predicción, como Random Forest, en escenarios complejos como la predicción del precio de la vivienda.

4.2. ARIMA

4.2.1. Modelo original

El primer modelo que implementaremos en el proyecto es el ARIMA (Autoregressive Integrated Moving Average), un modelo estadístico tradicional ampliamente utilizado para el análisis y predicción de series de tiempo. Es considerado el más sencillo de los modelos que emplearemos debido a su estructura **univariada**.

Características principales:

- Es un modelo **univariado**, lo cual significa que solo considera dos variables: la fecha y la variable objetivo (precio). Las demás variables presentes en el dataset no se incluyen en este análisis.
- Es especialmente efectivo cuando se busca capturar patrones temporales sin involucrar múltiples variables independientes.
- Se utiliza principalmente para series que son estacionarias o que pueden convertirse en estacionarias mediante procesos de diferenciación.

Componentes del modelo ARIMA:

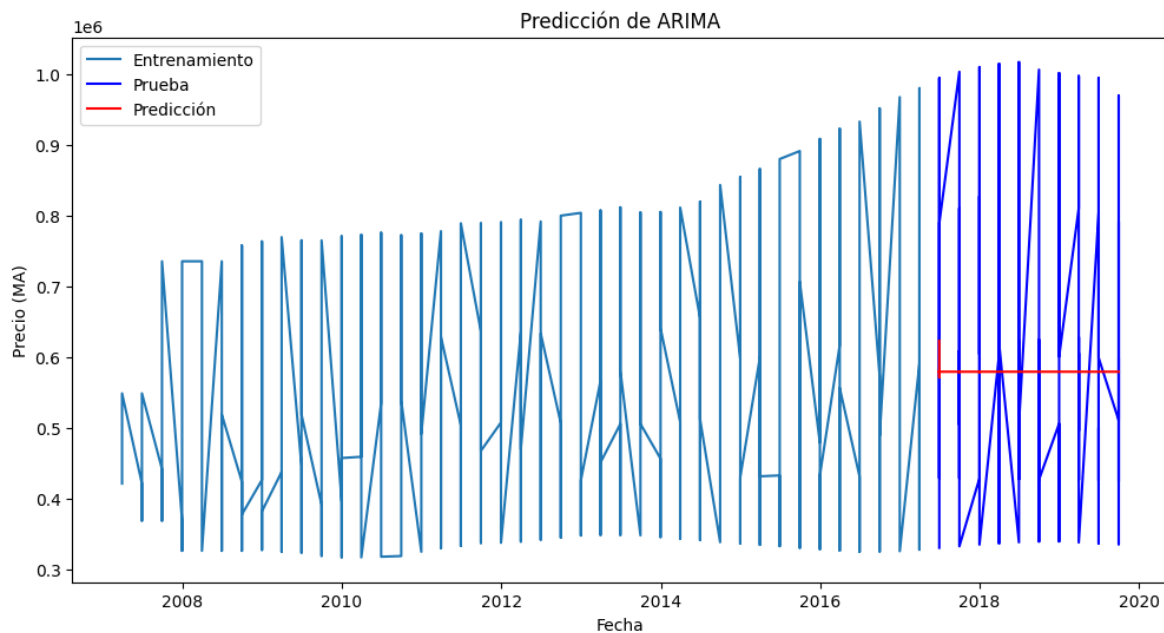
1. **Autoregresivo (AR):** Utiliza valores pasados de la serie para predecir el valor actual, basándose en la relación lineal entre observaciones anteriores y la observación actual.
2. **Integrado (I):** Aplica diferenciación para hacer que la serie sea estacionaria, eliminando tendencias o estacionalidades que puedan distorsionar las predicciones.
3. **Media Móvil (MA):** Modela la relación entre el valor actual y los errores de predicción previos, ajustando las predicciones con base en errores pasados.

Este modelo servirá como punto de partida para evaluar el desempeño de nuestros algoritmos de optimización y compararlos con otros enfoques más complejos.

El modelo ARIMA fue implementado utilizando los parámetros estándar ($p=1$, $d=1$, $q=1$). Estos valores representan:

- **p (Autoregresivo):** El número de observaciones pasadas utilizadas para predecir el valor actual. En este caso, se seleccionó **1**, lo que significa que solo se considera el valor de la serie temporal inmediatamente anterior.

- **d (Diferenciación):** El número de veces que se debe aplicar la diferenciación para hacer que la serie temporal sea estacionaria. En este modelo, **d=1**, lo que implica una diferenciación simple.
- **q (Media Móvil):** El número de términos de error pasados utilizados en el modelo. Un valor de **q=1** indica que se considera el error de predicción de la observación anterior.



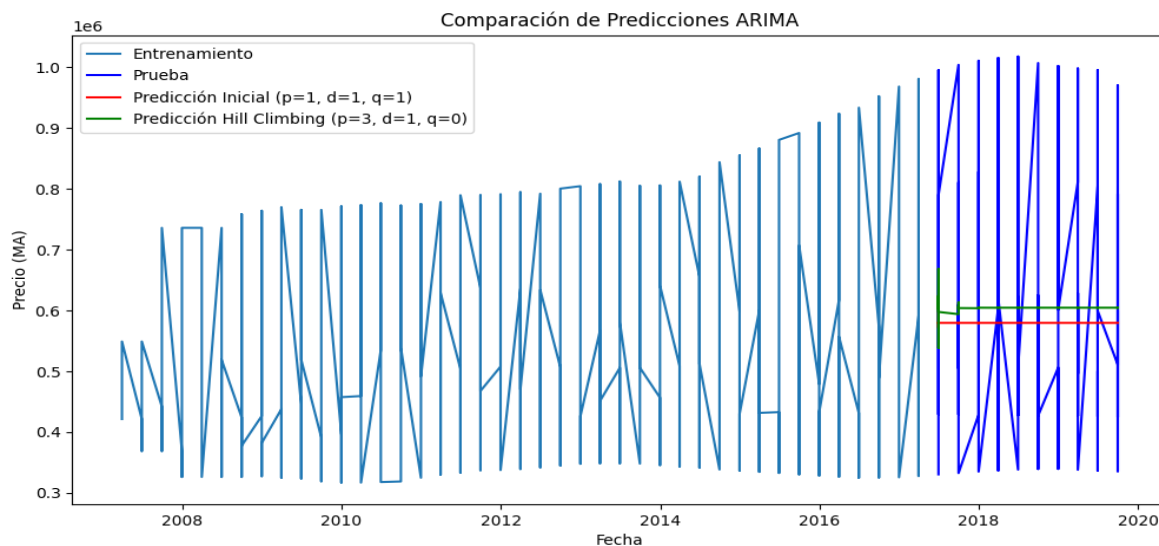
El modelo ARIMA presentó las siguientes métricas de error:

- **MSE (Error Cuadrático Medio):** \$ 45.260.747.979,3
- **RMSE (Raíz del Error Cuadrático Medio):** \$212.745,7

La media de los precios en la serie temporal es de 548.132. Al comparar el RMSE (\$212.745,7) con la media de los precios, vemos que representa aproximadamente el **38.8%** de la media, lo que sugiere que el modelo tiene un error notable en sus predicciones.

Este resultado podría explicarse por el hecho de que, al analizar la correlación entre la variable de tiempo y el precio de la vivienda, encontramos que la relación no es tan fuerte. Esto indica que el comportamiento del precio no está tan determinado por el paso del tiempo en sí mismo, lo que podría haber dificultado que el modelo ARIMA, que se basa en patrones temporales, lograra hacer predicciones precisas. Es probable que para obtener mejores resultados sea necesario incorporar variables adicionales con mayor correlación.

4.2.2. Hill Climbing



El algoritmo **Hill Climbing** fue aplicado al modelo ARIMA para optimizar las predicciones y mejorar su rendimiento. Los resultados fueron:

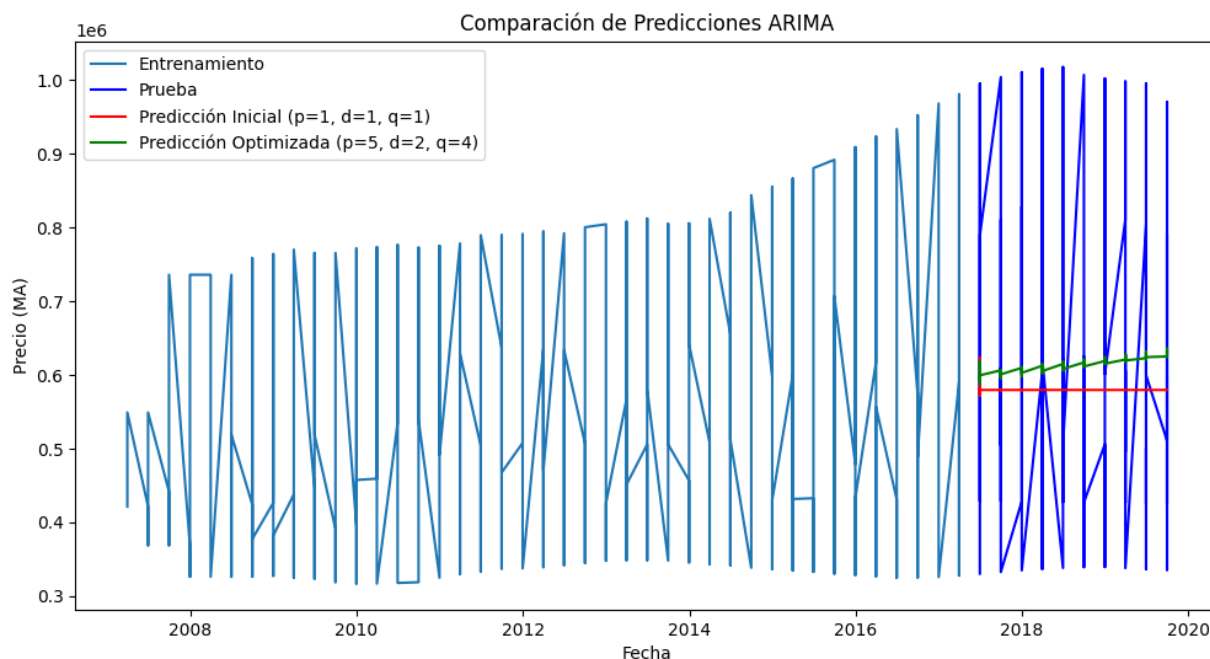
- **MSE:** \$ 35.217.983.265,4
- **RMSE:** \$ 187.669,5

Los parámetros óptimos dados por el modelo son **$p=3, d=1, q=0$** .

Al comparar los resultados, se observa una reducción significativa en el MSE (de \$ 45.260.747.979,3 a \$ 35.217.983.265,4) y una disminución en el RMSE (de 212.745,7 a 187.669,5). Esto indica que el algoritmo **Hill Climbing** logró mejorar el rendimiento del modelo ARIMA, reduciendo el error y mejorando la precisión de las predicciones. En términos de mejora en el **RMSE**, el porcentaje de reducción fue aproximadamente del **11.8%**.

Sin embargo, el **RMSE** sigue representando aproximadamente el **34.2%** de la media de los precios (548.132), lo que sugiere que, aunque se ha logrado una mejora significativa, la relación débil entre la variable de tiempo y el precio de la vivienda sigue limitando la precisión del modelo.

4.2.3. Algoritmo Genético



Los resultados del algoritmo genético fueron:

- **MSE:** \$ 45.134.299.511,8
- **RMSE:** \$ 212.448,3

Los parámetros óptimos dados por el modelo son **p=5, d=2, q=4**.

Al comparar los resultados, se observa una reducción en el MSE (de \$46.749.952.031 a \$45.134.299.511,8) y una disminución en el RMSE (de 212.745,7,4 a 212.448,3). Esto indica que el Algoritmo Genético también logró optimizar el modelo y reducir su error. En términos de mejora en el RMSE, el porcentaje de reducción fue aproximadamente del 0,14%.

Aunque el RMSE ha mejorado, sigue siendo relativamente alto (aproximadamente el 38,9% de la media de los precios, que es 548.132).

4.3. VAR

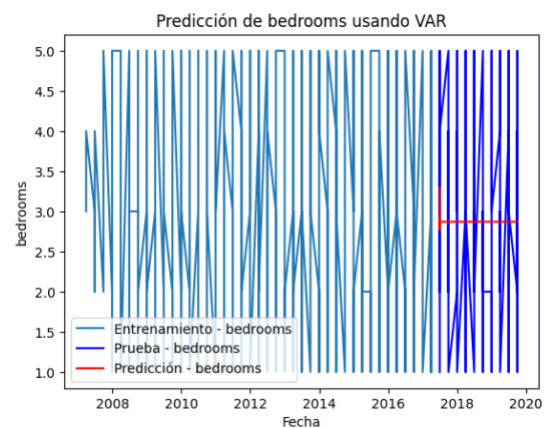
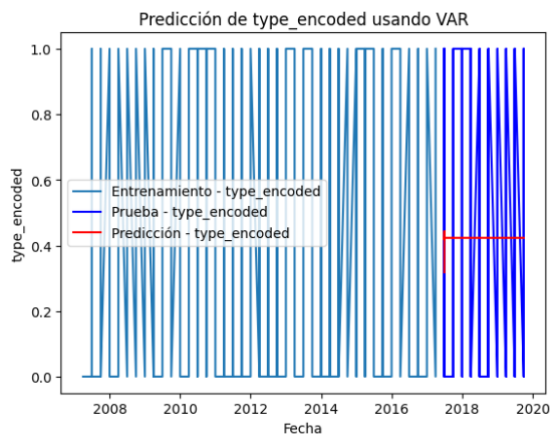
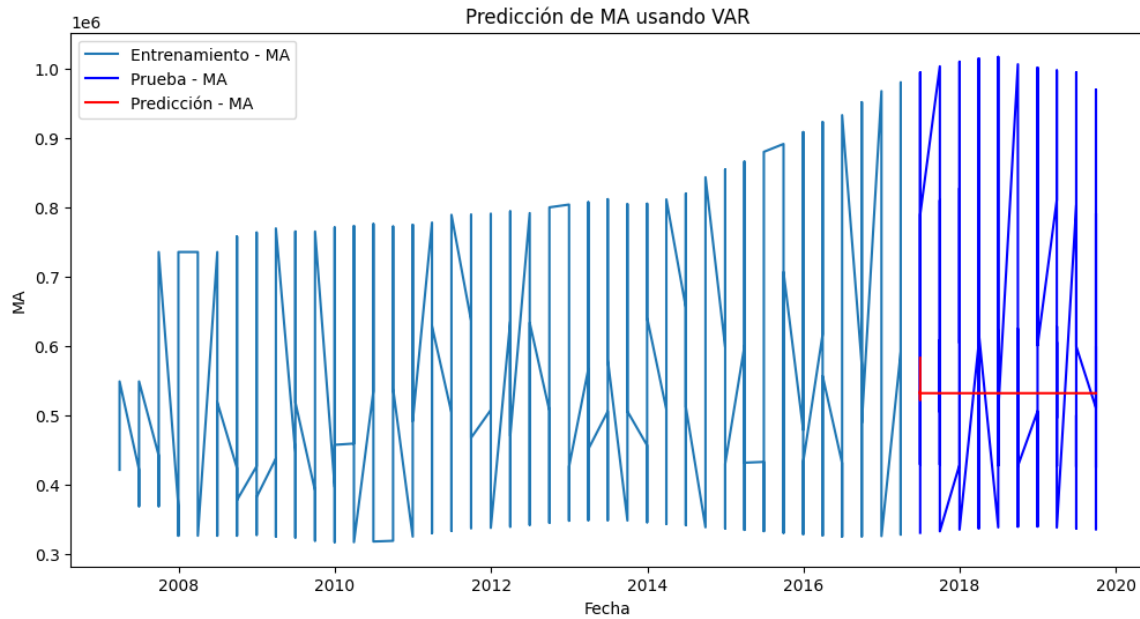
4.3.1. Modelo original

El **VAR (Vector Autoregressive)** es el segundo modelo que implementaremos en este proyecto. Se trata de un modelo estadístico tradicional utilizado en series de tiempo **multivariadas**, lo que significa que puede capturar las relaciones entre múltiples variables dentro de un mismo modelo. A diferencia del modelo ARIMA, que es univariado, el VAR permite considerar varias variables simultáneamente, lo que lo hace ideal para este proyecto, donde el precio de la vivienda está influenciado por diversas variables, como el número de habitaciones y el tipo de vivienda.

Una característica distintiva del VAR es que no requiere distinguir entre variables dependientes e independientes, ya que todas las variables son tratadas de manera simétrica. Esto permite modelar de forma más flexible la dinámica de las series de tiempo.

El modelo VAR captura las siguientes características clave:

- **Relaciones cruzadas entre variables:** Cada variable se modela no solo en función de sus propios rezagos, sino también de los rezagos de las demás variables incluidas en el modelo.
- **Estacionariedad:** Al igual que el modelo ARIMA, las series de tiempo deben ser estacionarias antes de aplicar el modelo, lo que implica que las propiedades estadísticas de la serie deben ser constantes en el tiempo.
- **Orden del modelo (lags):** Este parámetro determina cuántos períodos pasados de cada variable se deben considerar al predecir el valor actual, lo cual es crucial para captar las interacciones temporales de manera precisa.



Precio medio (MA): RMSE: 228.954,78

Muestra el mayor error de predicción, lo que refleja la complejidad de predecir el precio de la vivienda.

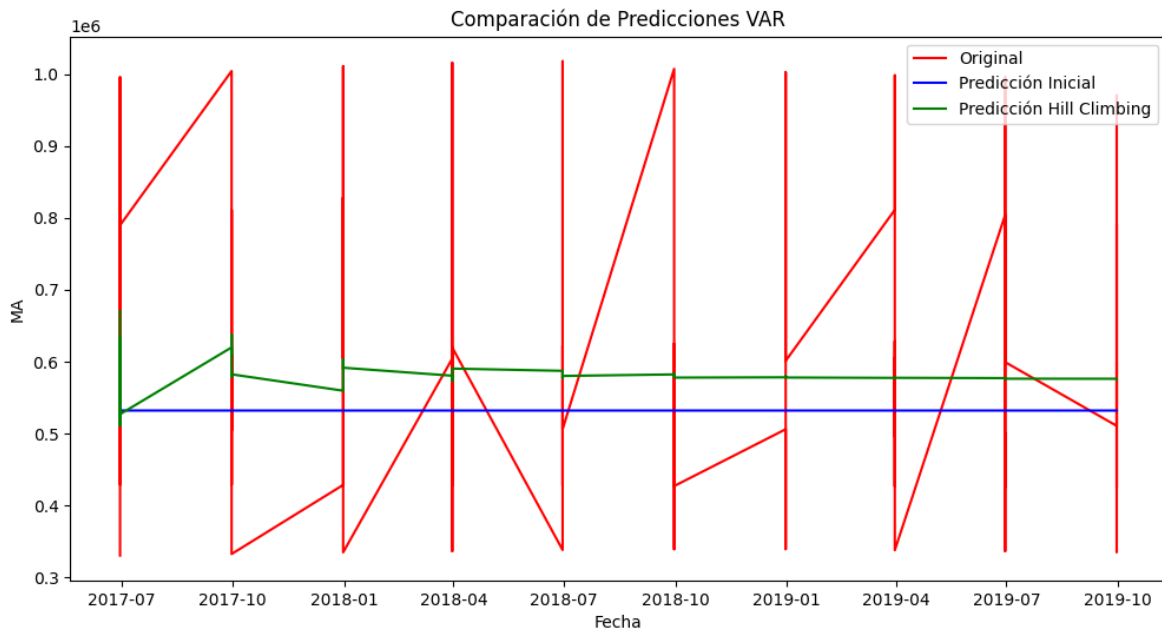
Tipo de vivienda (type_encoded): RMSE: 0.50

Presenta un RMSE bajo, lo que indica una predicción precisa debido a su naturaleza más simple.

Número de habitaciones (bedrooms): RMSE: 1.25

Muestra una relación moderada con el precio, con errores de predicción razonables.

4.3.2. Hill Climbing

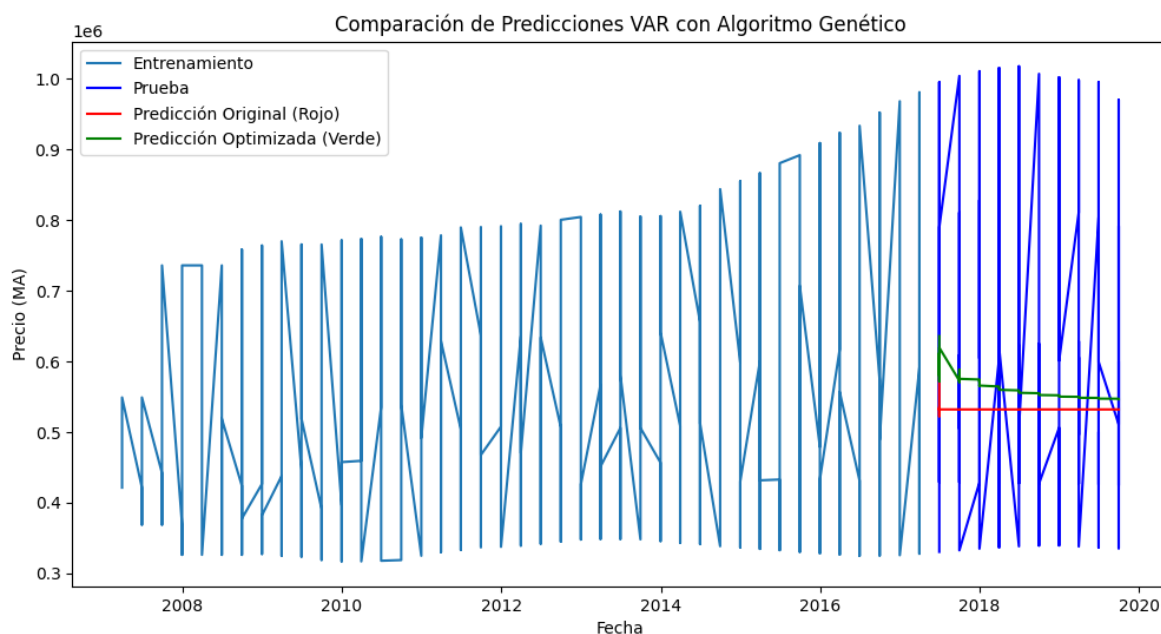


El algoritmo **Hill Climbing** fue aplicado al modelo VAR para optimizar las predicciones y mejorar su rendimiento. Los resultados fueron:

- **MSE:** 23,353,468,550.93
- **RMSE:** 152,818.42

Al comparar los resultados, se observa una reducción significativa en el MSE (de 26,218,629,879.27 a 23,353,468,550.93) y una disminución en el RMSE (de 228.954,78 a 152,818.42). Esto indica que el algoritmo Hill Climbing logró mejorar el rendimiento del modelo VAR, reduciendo el error y mejorando la precisión de las predicciones. En términos de mejora en el RMSE, el porcentaje de reducción fue aproximadamente del **33.23%**.

4.3.3. Algoritmo Genético



El **Algoritmo Genético** fue aplicado al modelo VAR para optimizar las predicciones y mejorar su rendimiento. Los resultados fueron:

- **MSE:** 48,245,819,042.46
- **RMSE:** 219,649.31

Al comparar los resultados, se observa una reducción en el MSE (de 52,420,292,804.45 a 48,245,819,042.46) y una disminución en el RMSE (de 228,954.78 a 219,649.31). Esto indica que el algoritmo Genético logró mejorar el rendimiento del modelo VAR, reduciendo el error y mejorando la precisión de las predicciones. En términos de mejora en el RMSE, el porcentaje de reducción fue aproximadamente del **4.1%**.

4.4. Random Forest

4.4.1. Modelo original

Random Forest es un algoritmo de Machine Learning que se basa en la construcción de múltiples árboles de decisión para realizar predicciones. Este método mejora la precisión de los modelos individuales al combinar los resultados de varios árboles, lo que reduce el riesgo de sobreajuste y mejora la generalización del modelo.

El algoritmo utiliza una técnica llamada **bagging** (bootstrap aggregating), donde cada árbol se entrena utilizando un subconjunto aleatorio de los datos. Una vez entrenados los árboles, las predicciones de cada uno se combinan para obtener el resultado final. En el caso de problemas de regresión, como la predicción de precios de vivienda, el modelo toma el promedio de las predicciones de todos los árboles, lo que aumenta la precisión general del modelo.

Random Forest es particularmente efectivo para manejar datos complejos y relaciones no lineales, lo que lo convierte en una opción ideal para este proyecto. En comparación con los modelos estadísticos tradicionales como ARIMA y VAR, Random Forest permitirá explorar la interacción entre múltiples variables y proporcionará una predicción robusta y precisa.

Las métricas de error de Random Forest son las siguientes:

- MSE: 19880036499.78398
- RMSE: 140996.5832911705
- MAE: 121783.7686698424

Al comparar estos resultados con los obtenidos en los modelos VAR y ARIMA, observamos lo siguiente:

- **MSE:** 19,880,036,499.78 en Random Forest es considerablemente más bajo que el MSE de ARIMA (45,260,747,979.3) y VAR (26,218,629,879.27), lo que sugiere que Random Forest ha logrado un mejor desempeño en términos de error cuadrático medio.
- **RMSE:** El RMSE de Random Forest (140,996.58) también es inferior al de ARIMA (212,745.7) y VAR (228.954,78), lo que indica una menor desviación promedio entre las predicciones y los valores reales.

- **MAE:** El MAE de Random Forest (121,783.77) es más bajo que los errores de predicción de ARIMA y VAR, lo que refuerza la idea de que Random Forest ha proporcionado predicciones más cercanas a los valores reales.

4.4.2. Hill Climbing

El algoritmo **Hill Climbing** optimizó el modelo de Random Forest con los siguientes parámetros:

- **n_estimators** = 65
- **max_depth** = None

El resultado obtenido es un MSE de 19,777,496,933.43, lo que produce un RMSE de 140,632.5.

Esto significa que hubo una mejora del error de 0.26%.

4.4.3. Algoritmo Genético

El algoritmo Genético optimizó el modelo de Random Forest en la **Generación 50** con el siguiente resultado:

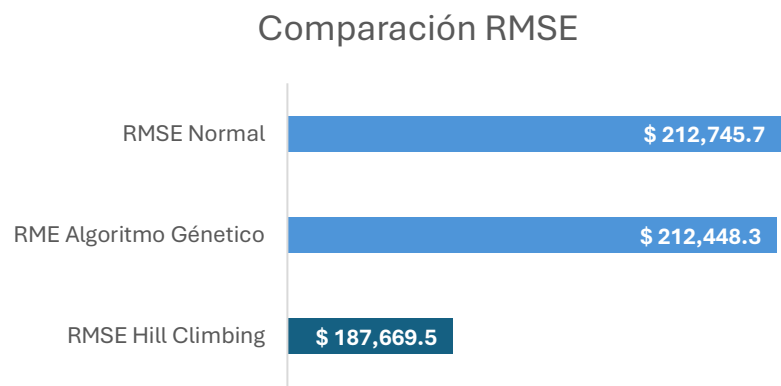
- **Mejor MSE** = 19,690,182,510.28
- **RMSE** = 140,235.35

Comparando con el modelo sin optimizar, cuyo MSE era 19880036499.78 y RMSE 140996.58, observamos que el algoritmo Genético logró una mejora del **0.54%** en la reducción del MSE y RMSE.

4. CONCLUSIONES Y POSIBLES MEJORAS

4.1. Modelo ARIMA

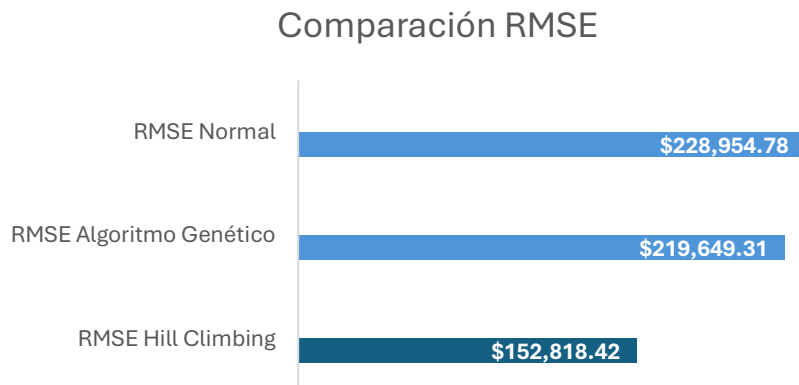
Al analizar la optimización del modelo ARIMA, encontramos que reflejó diferencias claras en la efectividad de los algoritmos aplicados. Hill Climbing logró reducir el RMSE en un 11.8%, mostrando una mejora considerable en la precisión del modelo, mientras que el Algoritmo Genético solo alcanzó una mejora del 0.14%, sin generar un ajuste significativo. Estos resultados sugieren que Hill Climbing fue la mejor estrategia de optimización para ARIMA, evidenciando su capacidad para encontrar configuraciones más eficientes en comparación con el método evolutivo.



A pesar de esta optimización, el RMSE aún representa el 39% de la media, lo que indica la necesidad de continuar mejorando el modelo. Para ello, se pueden explorar ajustes adicionales en los hiperparámetros, la inclusión de nuevas variables, un mejor tratamiento de la estacionalidad, modelos más sofisticados y la evaluación de métricas alternativas para una optimización más precisa.

4.2. Modelo VAR

El modelo VAR, por su parte, mostró un **RMSE** inicial de **\$228.954,78**, reflejando un alto error en la predicción del precio de vivienda. La optimización con **Hill Climbing** logró una reducción significativa del **33.2%**, mientras que el **Algoritmo Genético** solo mejoró en un **4.1%**, sugiriendo que la búsqueda local fue más efectiva en este caso. Nuevamente, la mejor optimización aplicada fue la de Hill Climbing, lo que puede explicarse por la naturaleza de las relaciones entre variables en el modelo VAR.

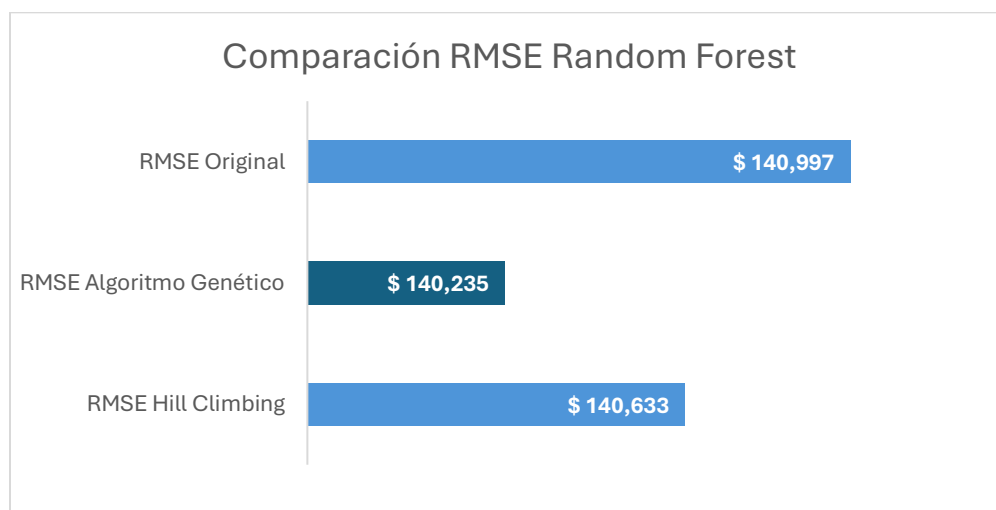


A pesar de la optimización, el RMSE del modelo VAR representa aproximadamente el 39.07% de la media del precio de la vivienda, lo que indica que el error sigue siendo significativo en relación con el valor promedio. Esto sugiere la necesidad de mejorar aún más el modelo mediante una mejor selección de rezagos, la inclusión de más variables explicativas o el uso de modelos más avanzados como VARMAX o redes neuronales recurrentes.

4.3. Modelo Random Forest

El modelo Random Forest presentó inicialmente un RMSE de \$140,996.58, logrando desde su configuración base un menor error en comparación con ARIMA y VAR. Tras la optimización con Hill Climbing, no se observó una mejora significativa, lo que sugiere que los hiperparámetros seleccionados inicialmente ya eran eficientes.

Por otro lado, el Algoritmo Genético logró una ligera mejora, reduciendo el RMSE a \$140,235.35, lo que representa un ajuste del 0.54%. Este resultado indica que, a diferencia de los modelos estadísticos ARIMA y VAR, donde Hill Climbing mostró mayor efectividad, en Random Forest la optimización mediante búsqueda evolutiva tuvo un impacto levemente superior.



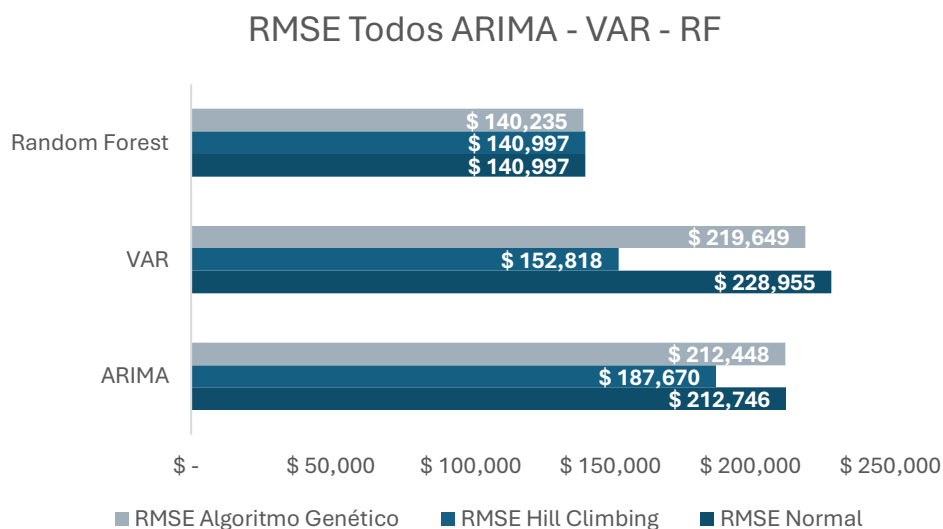
Dado que la mejora con optimización fue marginal, se podrían explorar estrategias más avanzadas, como Grid Search o Bayesian Optimization, para encontrar configuraciones de hiperparámetros más eficientes.

4.4. Comparación General de los Algoritmos de Optimización

En la optimización de los tres modelos (ARIMA, VAR y Random Forest), se observó que la efectividad de los algoritmos dependía del tipo de modelo. Hill Climbing fue más eficiente en los modelos estadísticos, logrando mejoras considerables en ARIMA (11.8%) y VAR (33.2%), lo que confirma su idoneidad para ajustar hiperparámetros en metodologías basadas en series de tiempo.

Cambio (%)	ARIMA	VAR	Random Forest
RMSE Algoritmo Genético	0,14%	4%	0,54%
RMSE Hill Climbing	11,8%	33%	0,26%

En contraste, en Random Forest la optimización tuvo un impacto marginal. Hill Climbing no logró reducir el error, y aunque el Algoritmo Genético mejoró el RMSE en un 0.54%, la diferencia fue mínima. Esto sugiere que la naturaleza del modelo influye directamente en la eficacia de los métodos de optimización, y que en modelos de árboles de decisión podrían explorarse estrategias más avanzadas como Grid Search o Bayesian Optimization.



Además, es importante considerar que los resultados obtenidos podrían haber mejorado con mayor tiempo de ejecución y mejor capacidad computacional. Dado que los algoritmos de optimización dependen del número de iteraciones y la profundidad de exploración del espacio de búsqueda, un mayor poder de cómputo habría permitido evaluar más configuraciones de hiperparámetros y posiblemente encontrar combinaciones óptimas, especialmente en el Algoritmo Genético, cuya exploración evolutiva suele requerir tiempos de ejecución más prolongados para converger a una solución óptima.

Estos resultados refuerzan la importancia de seleccionar la técnica de optimización adecuada según la naturaleza del modelo, asegurando que el ajuste de hiperparámetros realmente impacte en la precisión y reducción del error