

Pronóstico

Pronóstico



El pronóstico es una etapa fundamental en el análisis de series temporales, donde se utilizan los modelos desarrollados previamente para prever los valores futuros de la serie de tiempo. Esta técnica es crucial en una variedad de contextos y aplicaciones, ya que proporciona información valiosa para la planificación, la toma de decisiones y la gestión de recursos en diversas áreas.

Una de las principales razones para realizar pronósticos es anticipar posibles tendencias y patrones en los datos, permitiendo a las organizaciones y empresas prepararse y adaptarse de manera proactiva a los cambios. Veamos su uso en algunos sectores:



Sector financiero

El pronóstico de precios de acciones ayuda a los inversionistas a tomar decisiones informadas sobre dónde invertir su capital.



Ámbito empresarial

El pronóstico de la demanda de productos ayuda a planificar la producción y el inventario de manera eficiente.



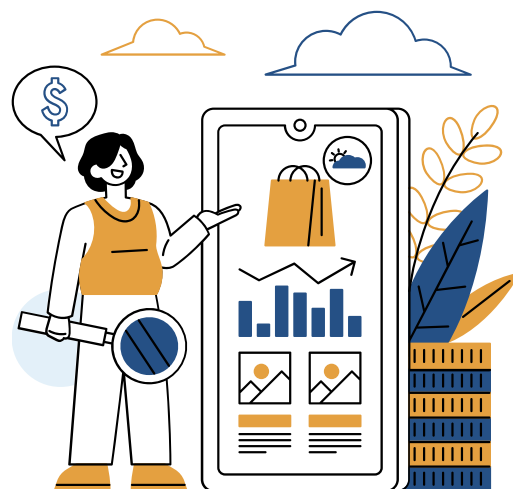
Meteorología

El pronóstico del clima es esencial para la planificación de actividades al aire libre y la gestión de riesgos asociados con fenómenos meteorológicos extremos.

Ejemplo de pronóstico:

El siguiente ejemplo realiza el pronóstico de una serie de tiempo utilizando el modelo ARIMA (AutoRegressive Integrated Moving Average).

Se importan las bibliotecas necesarias: pandas para el manejo de datos, numpy para operaciones numéricas, matplotlib para la visualización de gráficos y ARIMA de statsmodels para ajustar el modelo ARIMA.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
```

Se generan datos de ejemplo en forma de una serie de tiempo utilizando numpy y pandas. La serie de tiempo se compone de fechas y valores generados aleatoriamente.

```
# Generar datos de ejemplo
np.random.seed(0)
dates = pd.date_range(start='2022-01-01', end='2022-12-31')
values = np.arange(len(dates)) * 0.2 + np.random.rand(len(dates)) * 50
ts = pd.Series(values, index=dates)
```

Se dividen los datos en un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento contiene datos hasta una fecha específica ('2022-10-01'), mientras que el conjunto de prueba contiene datos desde esa fecha hasta el final de la serie.

```
# Dividir los datos en entrenamiento y prueba
train_data = ts[:'2022-10-01']
test_data = ts['2022-10-02':]
```

Se ajusta el modelo ARIMA al conjunto de entrenamiento utilizando el orden especificado como (10, 1, 10). Este orden indica que se están considerando 10 términos autoregresivos (AR), 1 diferencia de primer orden (I) y 10 términos de media móvil (MA).

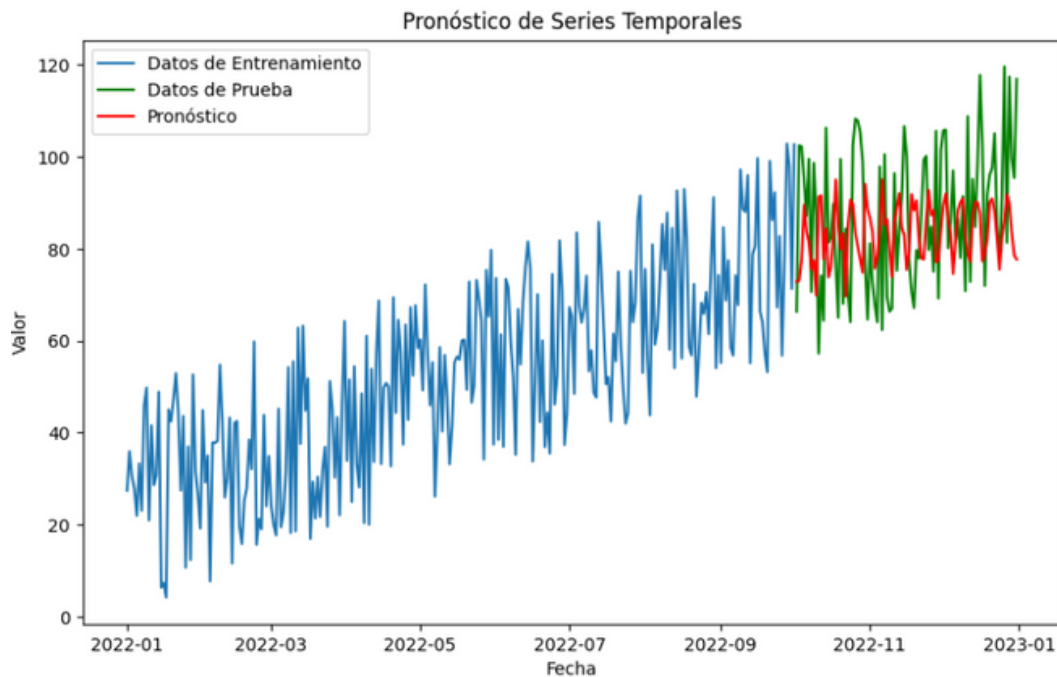
```
# Ajustar el modelo ARIMA
model = ARIMA(train_data, order=(10, 1, 10))
model_fit = model.fit()
```

Se realiza el pronóstico utilizando el modelo ajustado para prever los valores de la serie de tiempo en el conjunto de prueba.

```
# Realizar el pronóstico
forecast = model_fit.forecast(steps=len(test_data))
```

Finalmente, se visualizan los datos originales de entrenamiento, los datos de prueba y el pronóstico generado por el modelo ARIMA en un gráfico utilizando matplotlib.

```
# Gráfica de los datos originales y el pronóstico
plt.figure(figsize=(10, 6))
plt.plot(train_data, label='Datos de Entrenamiento')
plt.plot(test_data, label='Datos de Prueba', color='green')
plt.plot(test_data.index, forecast, label='Pronóstico', color='red')
plt.legend()
plt.title('Pronóstico de Series Temporales')
plt.xlabel('Fecha')
plt.ylabel('Valor')
plt.show()
```



En el modelo ARIMA, los parámetros de orden (p, d, q) especifican el número de términos autorregresivos (AR), la diferenciación (d) y el número de términos de medias móviles (MA) que se utilizarán en el modelo. Aquí hay algunos valores que podrías probar para el orden del modelo ARIMA:



Orden $(1, 0, 0)$: Un modelo ARIMA(1, 0, 0) implica un modelo autorregresivo simple de primer orden sin diferenciación y sin términos de medias móviles. Es útil cuando los datos son estacionarios y no hay componente de tendencia o estacionalidad.

Orden $(0, 1, 0)$: Un modelo ARIMA(0, 1, 0) implica solo diferenciación sin términos autorregresivos ni de medias móviles. Es útil cuando los datos tienen una tendencia lineal.



Orden $(1, 1, 0)$: Un modelo ARIMA(1, 1, 0) implica un modelo autorregresivo de primer orden con un término de diferenciación y sin términos de medias móviles. Es útil cuando los datos tienen una tendencia lineal y necesitan ser diferenciados una vez para hacerlos estacionarios.

Orden (0, 1, 1): Un modelo ARIMA(0, 1, 1) implica un modelo de medias móviles simple de primer orden con un término de diferenciación y sin términos autorregresivos. Es útil cuando los datos tienen una tendencia lineal y necesitan ser diferenciados una vez para hacerlos estacionarios, y también exhiben una componente de ruido que puede ser modelada por una media móvil.



Otros órdenes superiores: Dependiendo de la complejidad de tus datos y el grado de estacionalidad, tendencia y ruido, puedes probar con órdenes superiores como (2, 1, 2) o (5, 1, 5), etc., para capturar mejor las características de los datos.