

IBV
100

escuela
británica de
artes creativas
y tecnología

Buenas prácticas



Clase 01

Buenas prácticas

A continuación te compartimos algunas buenas prácticas sobre sobre sobre modelos autorregresivos $AR(p)$ incluyen:

1. Entender la teoría detrás de los modelos: Antes de aplicar modelos autorregresivos, es crucial comprender cómo funcionan y cómo los parámetros afectan el comportamiento de la serie temporal. Esto incluye conocer la influencia del parámetro ϕ en la autocorrelación y el momento de la serie.
2. Uso de gráficas para visualizar el comportamiento de la serie: Visualizar diferentes valores de ϕ y cómo afectan a la serie temporal es una buena práctica para obtener una comprensión intuitiva de la dinámica de la serie.
3. Determinar el orden óptimo del modelo: Utilizar técnicas como la función de autocorrelación parcial y criterios de información como AIC o BIC para seleccionar el orden adecuado del modelo $AR(p)$ es esencial para evitar el sobreajuste o el subajuste.

Clase 01

Buenas prácticas

4. Dividir los datos en conjuntos de entrenamiento y prueba: Esta práctica permite validar la capacidad predictiva del modelo y asegurar que no se ajuste excesivamente a los datos de entrenamiento.

5. Interpretar correctamente los resultados: Comprender y comunicar lo que significan los parámetros estimados, las predicciones y los intervalos de confianza es fundamental para la toma de decisiones basada en el modelo.

6. Evaluar el rendimiento del modelo: Utilizar métricas como el error cuadrático medio (MSE) y el error medio absoluto porcentual (MAPE) para evaluar la precisión de las predicciones del modelo.

Ejemplo de utilización en el mercado laboral:

Los modelos $AR(p)$ se utilizan en el análisis financiero para pronosticar precios de acciones, como se demostró con los datos históricos de precios de acciones de Apple. Los analistas pueden utilizar estos modelos para tomar decisiones de inversión informadas o para estrategias de trading algorítmico.

Clase 01

Buenas prácticas

Código ejecutable ejemplificando el tema:

```
```python
import yfinance as yf
import pandas as pd
import warnings
from statsmodels.tsa.ar_model import AutoReg
from sklearn.metrics import mean_squared_error,
mean_absolute_percentage_error

Descargar datos históricos de precios de acciones de Apple
data = yf.download('AAPL', start='2010-01-01', end='2020-01-01')

Dividir los datos en conjuntos de entrenamiento y prueba
train_data = data['Close'][:'2018-12-31']
test_data = data['Close']['2019-01-01':]

Ajustar un modelo AR(1)
warnings.filterwarnings("ignore")
model = AutoReg(train_data, lags=1).fit()
```

# Clase 01

Buenas prácticas

```
Imprimir la estimación del parámetro phi
print(f'phi: {model.params}')
```

```
Hacer predicciones
predictions = model.predict(start=len(train_data),
end=len(train_data) + len(test_data) - 1, dynamic=False)
```

```
Asignar fechas a las predicciones y comparar con los
valores reales
predictions.index = test_data.index
comparison = pd.DataFrame({'Actual': test_data, 'Predicted':
predictions})
```

```
Calcular MSE y MAPE
mse = mean_squared_error(test_data, predictions)
mape = mean_absolute_percentage_error(test_data,
predictions)
print(f'MSE: {mse}, MAPE: {mape}')
```

# Clase 01

Buenas prácticas

Explicación del código:

- Se importan las librerías necesarias.
- Se descargan los datos históricos de precios de acciones de Apple utilizando ``yfinance``.
- Se dividen los datos en conjuntos de entrenamiento y prueba.
- Se ajusta un modelo autorregresivo de primer orden ( $AR(1)$ ) a los datos de entrenamiento utilizando ``statsmodels``.
- Se imprimen los parámetros estimados del modelo, incluyendo  $\phi$ .
- Se realizan predicciones para el conjunto de prueba.
- Se asignan fechas a las predicciones y se comparan con los valores reales.
- Se calculan y se imprimen el MSE y el MAPE para evaluar el rendimiento del modelo.

# Clase 02

## Buenas prácticas

A continuación te compartimos algunas buenas prácticas sobre sobre pronósticos en series de tiempo utilizando modelos autorregresivos, específicamente el modelo  $AR(1)$ , incluyen:

1. Realizar Pronósticos con Metodología Adecuada: Utilizar el método Forecast para realizar pronósticos a corto plazo, como cinco días hábiles futuros, y visualizar estos pronósticos con herramientas como el comando Tail en Python.
2. Generar Pronósticos de Confianza: Crear un pronóstico de confianza con un intervalo del 95% para dar una estimación de la incertidumbre asociada con las predicciones.
3. Ajustar la Base de Prueba: Asegurarse de que las fechas funcionen como índice en la base de prueba para facilitar la manipulación y visualización de los datos.
4. Manipulación de Datos para Estimación de Fechas Futuras: Utilizar ciclos para agregar días hábiles a la última fecha conocida y convertir los resultados en formato de fecha, lo cual es crucial para la concatenación con las predicciones.

# Clase 02

Buenas prácticas

5. Preparación de Datos para Concatenación y Visualización: Agregar las fechas estimadas a la base de datos existente y preparar los datos para la concatenación con las predicciones y su posterior visualización en gráficos.

6. Validación de la Adecuación del Modelo: Utilizar herramientas como la gráfica de autocorrelación parcial y los criterios de información de Akaike (AIC) y Bayesiano (BIC) para validar la adecuación del modelo autorregresivo elegido.

7. Aplicación Práctica en el Mercado Laboral: Analizar series de tiempo con datos históricos reales, como los precios de acciones de empresas conocidas, para ajustar modelos autorregresivos y pronosticar valores futuros, justificando la exactitud de las predicciones con indicadores de bondad de ajuste.

8. Documentación y Presentación de Resultados: Entregar un Jupyter Notebook y un archivo PDF con capturas de pantalla que muestren el código desarrollado y la solución al problema planteado, incluyendo comentarios que respondan a las preguntas formuladas.



# Clase 02

Buenas prácticas

Ejemplo de código ejecutable en Python para un modelo AR(1):

```

python
import pandas as pd
import numpy as np
from statsmodels.tsa.ar_model import AutoReg
import matplotlib.pyplot as plt

Cargar datos de series de tiempo
data = pd.read_csv('precios_acciones.csv', index_col='fecha',
parse_dates=True)

Ajustar un modelo AR(1)
modelo_ar1 = AutoReg(data['precio'], lags=1).fit()

Realizar pronóstico para los próximos 5 días hábiles
pronostico = modelo_ar1.forecast(steps=5)

Crear intervalo de confianza al 95%
intervalo_confianza =
modelo_ar1.get_prediction(start=-5).conf_int(alpha=0.05)

```

# Clase 02

Buenas prácticas

```
Visualizar los resultados
plt.figure(figsize=(10, 5))
plt.plot(data['precio'], label='Datos Históricos')
plt.plot(pronostico.index, pronostico, label='Pronóstico AR(1)')
plt.fill_between(intervalo_confianza.index,
intervalo_confianza.iloc[:, 0], intervalo_confianza.iloc[:, 1],
color='pink', label='Intervalo de Confianza 95%')
plt.legend()
plt.show()
'''
```

Explicación del código:

- Se importan las librerías necesarias para el manejo de datos, modelado y visualización.
- Se cargan los datos de las acciones utilizando `pandas`, estableciendo la columna de fecha como índice y asegurándose de que se interpreten como fechas.
- Se ajusta un modelo autorregresivo de primer orden (AR(1)) a los precios de las acciones.

# Clase 03

Buenas prácticas

- Se realiza un pronóstico para los próximos 5 días hábiles utilizando el modelo ajustado.
- Se calcula el intervalo de confianza al 95% para el pronóstico.
- Se visualizan los datos históricos, el pronóstico y el intervalo de confianza en un gráfico para una interpretación clara de los resultados.

**¡Mucho éxito en tus estudios!**