



Analiza și Prognoza Seriilor de Timp

Capitolul 9: Prophet și TBATS



Daniel Traian PELE

Academia de Studii Economice din București

IDA Institute Digital Assets

Blockchain Research Center

AI4EFin Artificial Intelligence for Energy Finance

Academia Română, Institutul de Prognoză Economică

MSCA Digital Finance

Obiective de învățare

La finalul acestui capitol, veți fi capabili să:

1. Gestionati serii de timp cu **sezonalități multiple**
2. Utilizați **Prophet (Meta)** pentru prognoză flexibilă cu sărbători
3. Aplicați modele **TBATS** (Trigonometric Box-Cox ARMA Trend Seasonal) pentru sezonabilitate complexă
4. Comparați și selectați între metodele moderne de prognoză

Cuprins

Fundamente

- ▣ Sezonaliități Multiple
- ▣ Modelul TBATS
- ▣ Prophet (Meta)

Aplicații

- ▣ Comparatie și Ghid de Selecție
- ▣ Studiu de Caz
- ▣ NeuralProphet și Predicție Conformală
- ▣ Rezumat și Quiz

Problema: tipare sezoniere complexe

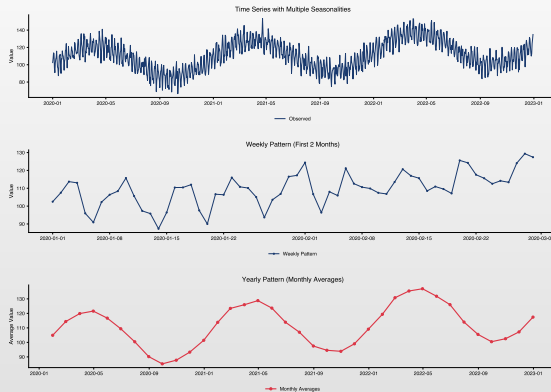
Exemple din lumea reală

- ▣ **Cerere de electricitate pe oră:** Tipare zilnice + săptămânale + anuale
- ▣ **Trafic web:** Zilnic + săptămânal + efecte de sărbători
- ▣ **Vânzări retail:** Săptămânal + lunar + anual + sărbători
- ▣ **Volum call center:** Pe oră + zilnic + săptămânal

Limitarea SARIMA (Seasonal ARIMA)

- ▣ $SARIMA(p, d, q)(P, D, Q)_s$ standard gestionează doar **o singură** perioadă sezonieră s
- ▣ Pentru date orare cu tipare zilnice și săptămânale, avem nevoie de $s_1 = 24$ și $s_2 = 168$

Exemplu: date orare cu sezonalități multiple



Soluții pentru sezonalități multiple

Abordări tradiționale

- ▣ **Termeni Fourier:** Adăugare regresori sin/cos
- ▣ **Variable indicatoare:** Mulți parametri
- ▣ **Modele imbricate:** Specificare complexă

Abordări moderne

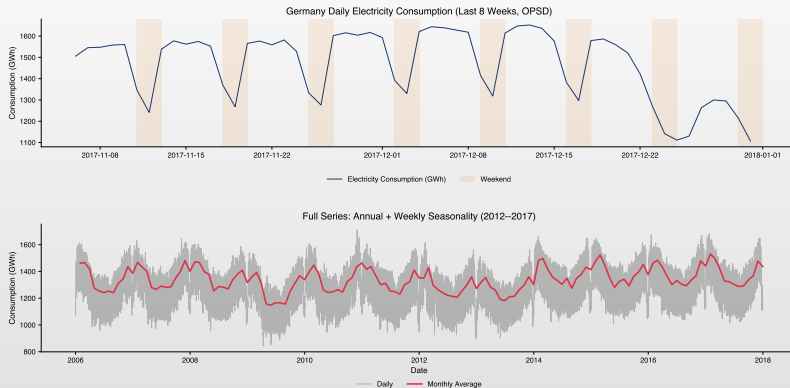
- ▣ **TBATS:** Automat, gestionează multe perioade
- ▣ **Prophet:** Flexibil, interpretabil
- ▣ **Metode neurale:** Deep learning

Comparație

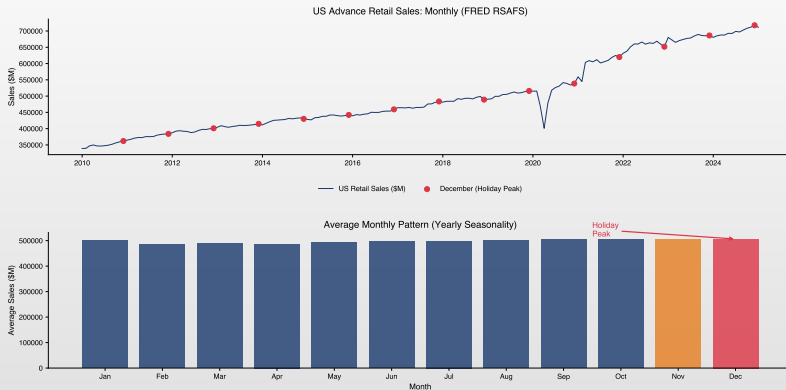
- ▣ Rezumat comparativ:

Metodă	Nr. Max Sezonalități	Interpretabil
SARIMA	1	Da
Fourier + ARIMA (AutoRegressive Integrated Moving Average)	Multiple	Moderat
TBATS	Multiple	Moderat
Prophet	Multiple	Da

Exemplu real: cerere de electricitate



Exemplu real: vânzări cu amănuntul — influența sărbătorilor



Portret de cercetător: Rob J. Hyndman



*1967

 [Wikipedia \(en\)](#)

Biografie

- ▣ Statistician australian, profesor la Monash University
- ▣ Unul dintre cei mai influenți cercetători în prognoza seriilor de timp
- ▣ Creatorul pachetului `forecast` pentru R, utilizat pe scară largă
- ▣ Redactor-șef al *International Journal of Forecasting* (2005–2018)

Contribuții principale

- ▣ **Modelul TBATS** (2011) — Box-Cox trigonometric ARMA cu perioade sezoniere multiple
- ▣ **Cadrul ETS** (Error, Trend, Seasonality) — modele spațiu-stare de netezire exponențială cu selecție automată
- ▣ **Pachetul forecast** pentru R — setul standard de instrumente pentru prognoza seriilor de timp
- ▣ **Prognoza ierarhică** și metode de reconciliere a prognozelor

TBATS: ce înseamnă?

Componentele TBATS

- ▣ **T** \Rightarrow Sezonalitate **Trigonometrică** folosind termeni Fourier
- ▣ **B** \Rightarrow Transformare **Box-Cox** pentru stabilizarea varianței
- ▣ **A** \Rightarrow Erori **ARMA** (AutoRegressive Moving Average) pentru autocorelația reziduală
- ▣ **T** \Rightarrow Componentă de **Trend** (posibil amortizat)
- ▣ **S** \Rightarrow Componente **Sezoniere** (multiple permise)

Inovația principală

- ▣ TBATS folosește **reprezentare trigonometrică** pentru sezonalitate:

$$s_t^{(i)} = \sum_{j=1}^{k_i} \left[s_j^{(i)} \cos \left(\frac{2\pi jt}{m_i} \right) + s_j^{*(i)} \sin \left(\frac{2\pi jt}{m_i} \right) \right]$$

- ▣ m_i este perioada sezonieră de index i , iar k_i este numărul de armonici

Transformarea Box-Cox

Definiție 1 (Transformarea Box-Cox)

Transformarea Box-Cox cu parametrul λ este definită astfel:

$$y_t^{(\lambda)} = \begin{cases} \frac{y_t^\lambda - 1}{\lambda} & \text{dacă } \lambda \neq 0 \\ \ln(y_t) & \text{dacă } \lambda = 0 \end{cases}$$

Scop

- ▣ **Stabilizarea varianței:** Face varianța constantă în timp
- ▣ **Normalizare:** Reduce asimetria în date
- ▣ Valori uzuale: $\lambda = 0$ (log), $\lambda = 0.5$ (rădăcină pătrată), $\lambda = 1$ (fără transformare)

Structura modelului TBATS

Specificația completă a modelului

□ Ecuațiile modelului TBATS:

$$y_t^{(\lambda)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^M s_{t-1}^{(i)} + d_t$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t$$

$$b_t = \phi b_{t-1} + \beta d_t$$

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t$$

Notații

- $y_t^{(\lambda)} \Rightarrow$ seria transformată Box-Cox (dacă $\lambda \neq 1$)
- $\ell_t \Rightarrow$ nivelul local, $b_t \Rightarrow$ trendul cu amortizare ϕ
- $s_t^{(i)} \Rightarrow M$ componente sezoniere cu perioade m_1, \dots, m_M
- $d_t \Rightarrow$ procesul de eroare ARMA(p, q)

TBATS: evoluția stărilor sezonaliității trigonometrice

Definiție 2 (Recurсія trigonometrică în spațiul stărilor)

Pentru fiecare componentă sezonieră cu perioada m_i și k_i armonici, definim stările:

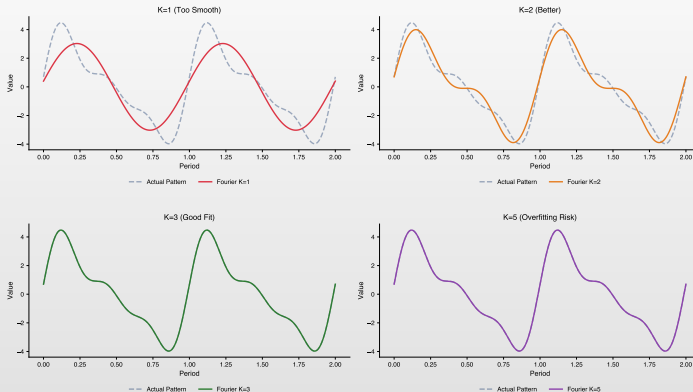
$$\begin{pmatrix} s_{j,t}^{(i)} \\ s_{j,t}^{*(i)} \end{pmatrix} = \begin{pmatrix} \cos(\omega_j) & \sin(\omega_j) \\ -\sin(\omega_j) & \cos(\omega_j) \end{pmatrix} \begin{pmatrix} s_{j,t-1}^{(i)} \\ s_{j,t-1}^{*(i)} \end{pmatrix} + \begin{pmatrix} \gamma_1^{(i)} \\ \gamma_2^{(i)} \end{pmatrix} d_t$$

unde $\omega_j = \frac{2\pi j}{m_i}$ este frecvența armoniciei j .

Interpretare

- Matricea de rotație păstrează structura periodică
- Sezonalitatea totală: $s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$
- Parametri: $2k_i$ stări per perioadă sezonieră

Aproximarea Fourier a sezonaliității



TBATS: sezonabilitate trigonometrică

De ce termeni Fourier/trigonometrici?

- ▣ **Simplu:** Mai puțini parametri decât variabilele indicatoare
- ▣ **Neted:** Captează natural tiparele sezoniere netede
- ▣ **Flexibil:** Numărul de armonici k controlează complexitatea
- ▣ **Perioade non-întregi:** Poate gestiona $s = 365.25$ pentru date zilnice

k mic (puține armonici)

- ▣ Tipar neted
- ▣ Mai puțini parametri
- ▣ Poate rata vârfuri abrupte

k mare (multe armonici)

- ▣ Poate capta orice tipar
- ▣ Mai mulți parametri
- ▣ Risc de supraajustare

TBATS în practică

Implementare Python

- ▣ **Pachet** `tbats`: Oferă selecție automată a modelului — selectează automat Box-Cox λ , numărul de armonici k_i , ordinele ARMA (p, q) , trend amortizat vs neamortizat

Exemplu de cod

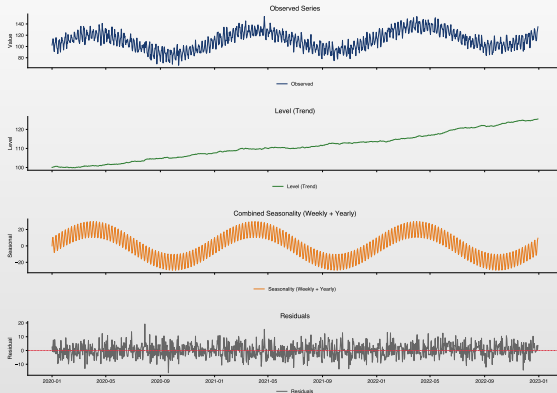
- ▣ **Cod Python**:

```
from tbats import TBATS
estimator = TBATS(seasonal_periods=[7, 365.25])
model = estimator.fit(y)
forecast = model.forecast(steps=30)
```

Notă

- ▣ BATS (Box-Cox ARMA Trend Seasonal) este versiunea mai simplă fără termeni trigonometrice
- ▣ BATS folosește stări sezoniere tradiționale în loc de reprezentare Fourier

Exemplu descompunere TBATS



TBATS: avantaje și limitări

Avantaje

- ▣ Gestionează **multiple** perioade sezoniere
- ▣ Selecție **automată** a modelului
- ▣ Gestionează perioade **non-întregi** (365.25)
- ▣ **Box-Cox** pentru heteroscedasticitate
- ▣ Bun pentru date de **frecvență înaltă**

Limitări

- ▣ **Intensiv computațional**
- ▣ Fără **regresori externi**
- ▣ Mai puțin **interpretabil** decât Prophet
- ▣ Poate fi **lent** pentru serii foarte lungi
- ▣ Necesită **suficiente date** per sezon

Prophet: prezentare generală

Ce este Prophet?

- ▣ **Origine:** Procedură de prognoză dezvoltată de Facebook (Meta) în 2017
- ▣ **Proiectat pentru serii de timp de business** cu:
 - ▶ Efecte sezoniere puternice (zilnice, săptămânale, anuale)
 - ▶ Efecte de sărbători
 - ▶ Puncte de schimbare în trend (*change points*)
 - ▶ Date lipsă și outlieri

Principiul de bază

- ▣ *Prognoză cu analistul implicat activ în proces (analyst-in-the-loop)*
- ▣ Prophet este proiectat pentru a fi ajustat de analiști cu cunoștințe de domeniu
- ▣ Nu necesită experiență avansată în modelarea seriilor de timp

Structura modelului Prophet

Abordare prin descompunere

- Prophet folosește o **descompunere aditivă**:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

$g(t)$: Trend

- Liniar sau logistic
- Changepoints automate
- Saturație de creștere

$s(t)$: Sezonalitate

- Serii Fourier
- Perioade multiple
- Sezonalitate personalizată

$h(t)$: Sărbători

- Sărbători pe țară
- Evenimente custom
- Efecte de fereastră

Prophet: componenta de trend

Trend liniar cu Changepoints

- **Ecuatia:** $g(t) = (k + \mathbf{a}(t)^T \boldsymbol{\delta}) \cdot t + (m + \mathbf{a}(t)^T \boldsymbol{\gamma})$
- **Parametri:**
 - ▶ $k \Rightarrow$ rata de creștere de bază
 - ▶ $\boldsymbol{\delta} \Rightarrow$ vector de ajustări de rată la changepoints
 - ▶ $\mathbf{a}(t) \Rightarrow$ indică ce changepoints sunt active la momentul t
 - ▶ $m \Rightarrow$ offset-ul, $\boldsymbol{\gamma}$ asigură continuitatea

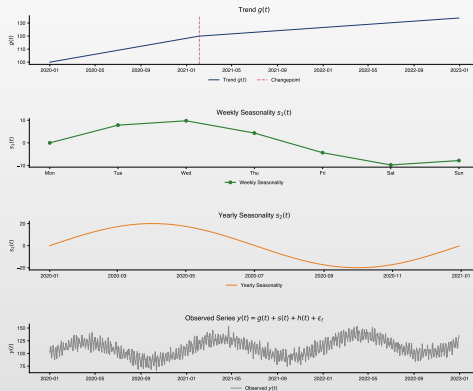
Creștere logistică (pentru trenduri cu saturație)

- Ecuatia de creștere logistică:

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^T \boldsymbol{\delta})(t - (m + \mathbf{a}(t)^T \boldsymbol{\gamma})))}$$

- $C(t)$ este capacitatea maximă (posibil variabilă în timp)

Descompunerea componentelor Prophet



Prophet: componenta de sezonabilitate

Reprezentare prin serii Fourier

- Pentru o perioadă sezonieră P , Prophet folosește:

$$s(t) = \sum_{n=1}^N \left[a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right]$$

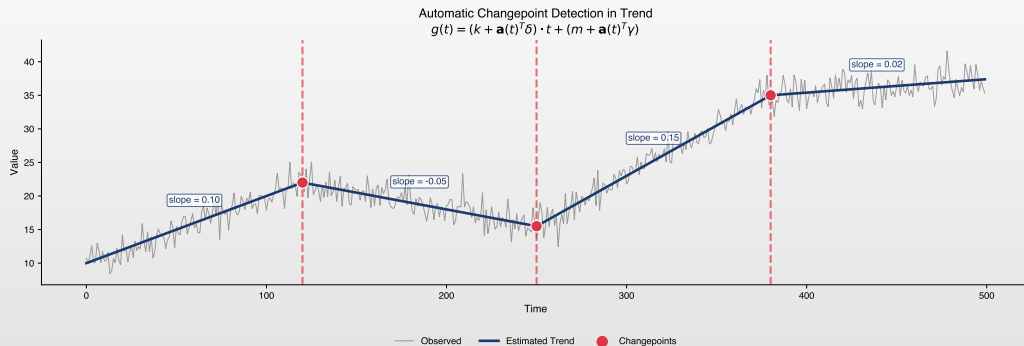
Setări implicite

- **Anuală:** perioadă 365.25 zile, ordin Fourier 10
- **Săptămânală:** perioadă 7 zile, ordin Fourier 3
- **Zilnică:** perioadă 1 zi, ordin Fourier 4

Atenție

- Ordin Fourier N mai mare \Rightarrow mai multă flexibilitate (tipare mai complexe) dar risc mai mare de supraajustare

Detectarea changepoints în trend



Prophet: efecte de sărbători

Modelul de sărbători

- Ecuația efectelor de sărbătoare:

$$h(t) = Z(t) \cdot \kappa$$

- $Z(t)$ este o matrice indicator pentru sărbători și κ sunt efectele sărbătorilor

Caracteristici

- **Sărbători integrate:** 60+ țări suportate
- **Sărbători custom:** Adăugați propriile evenimente (Black Friday, evenimente companie)
- **Efecte de fereastră:** Sărbătorile pot afecta zilele înainte/după
- **Scala distribuției *a priori*:** Controlează regularizarea efectelor de sărbătoare

Exemplu de cod

- Cod Python:

```
holidays = pd.DataFrame({'holiday': 'black_friday', ...})  
model = Prophet(holidays=holidays)
```

Prophet în practică

Utilizare de bază

```
□ Cod Python: from prophet import Prophet
import pandas as pd

# Datele trebuie să aibă coloane 'ds' (dată) și 'y' (valoare)
df = pd.DataFrame({'ds': dates, 'y': values})

model = Prophet()
model.fit(df)

future = model.make_future_dataframe(periods=365)
forecast = model.predict(future)
```

Adăugare sezonalitate personalizată

```
□ Cod Python: model = Prophet(weekly_seasonality=False)
model.add_seasonality(name='monthly', period=30.5, fourier_order=5)
model.add_seasonality(name='quarterly', period=91.25, fourier_order=3)
```

Prophet: cuantificarea incertitudinii

Trei surse de incertitudine

- ▣ **Incertitudine de trend:** Changepoints viitoare sunt incerte
- ▣ **Incertitudine de sezonaliitate:** Incertitudine în estimarea parametrilor
- ▣ **Zgomot de observație:** Aleatorietate inerentă

Intervale de predicție

- ▣ **Prophet oferă:**
 - ▶ Prognoză punctuală: `yhat`
 - ▶ Limita inferioară: `yhat_lower`
 - ▶ Limita superioară: `yhat_upper`
- ▣ **Implicit:** interval de 80%, schimbați cu `interval_width=0.95`

Notă

- ▣ Incertitudinea crește cu orizontul de prognoză
- ▣ Efectul este cel mai pronunțat pentru componenta de trend

Prophet: parametri de ajustare

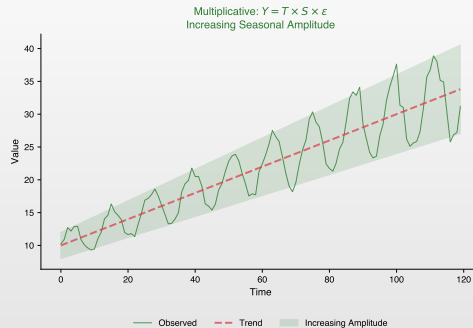
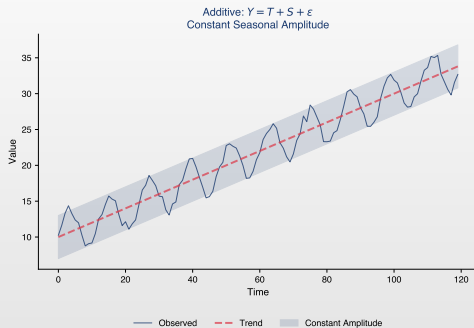
Parametri principali

- ▣ `changepoint_prior_scale`: Flexibilitate trend (implicit: 0.05)
- ▣ `seasonality_prior_scale`: Flexibilitate sezonabilitate (implicit: 10)
- ▣ `holidays_prior_scale`: Mărime efect sărbători (implicit: 10)
- ▣ `seasonality_mode`: 'additive' sau 'multiplicative'
- ▣ `changepoint_range`: Porțiuni din istoric pentru changepoints

Sfaturi practice

- ▣ **Supraajustare pe trend?** Micșorați `changepoint_prior_scale`
- ▣ **Subajustare pe sezonabilitate?** Măriți `seasonality_prior_scale`
- ▣ **Amplitudinea sezonieră variază?** Folosiți `seasonality_mode='multiplicative'`

Sezonalitate aditivă vs multiplicativă



 TSA_ch9_additive_vs_multiplicative

Prophet: metoda de estimare

Estimare Bayesiană prin Stan

- ▣ Prophet folosește **Stan** (limbaj probabilistic) pentru inferență
- ▣ **Implicit**: MAP (Maximum A Posteriori) — rapid, punct unic
- ▣ **Opțional**: MCMC complet — intervale de credibilitate complete
 - ▶ `model = Prophet(mcmc_samples=300)`

Priori și regularizare

- ▣ $\delta \sim \text{Laplace}(0, \tau)$ — favorizează changepoints rare (sparsitate)
- ▣ τ mic \Rightarrow trend rigid; τ mare \Rightarrow trend flexibil (implicit: $\tau = 0.05$)
- ▣ $\kappa \sim N(0, \nu^2)$ — regularizare L2 pe efectele de sărbători
- ▣ Coeficienții Fourier: $a_n, b_n \sim N(0, \sigma_s^2)$ — controlează amplitudinea sezonality

Prophet: avantaje și limitări

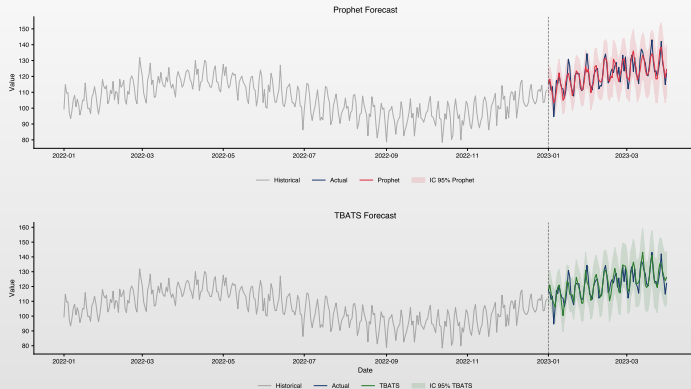
Avantaje

- ▣ **Accesibil:** Ajustare minimă necesară
- ▣ **Interpretabil:** Descompunere clară
- ▣ **Gestionează date lipsă** bine
- ▣ **Efecte sărbători** integrate
- ▣ **Sezonalități multiple**
- ▣ **Regresori externi** suportați
- ▣ **Ajustare rapidă**

Limitări

- ▣ **Nu bazat pe ARIMA:** Fără modelare autocorelație
- ▣ **Focus pe date zilnice:** Mai puțin potrivit pentru frecvență foarte înaltă
- ▣ **Ipoteze de trend:** Liniar/logistic poate să nu se potrivească
- ▣ **CV (Cross-Validation) integrat:** `cross_validation()` disponibil, dar necesită configurare atentă
- ▣ **Risc supraajustare** cu multe sezonalități

Comparație Prophet vs TBATS: prognoze



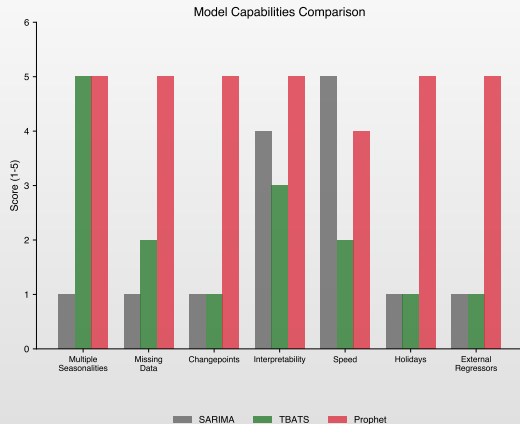
TBATS vs Prophet: comparație directă

Comparație detaliată

☐ Rezumat al diferențelor principale:

Caracteristică	TBATS	Prophet
Sezonalități multiple	Da (automat)	Da (manual sau auto)
Efecte sărbători	Nu	Da (integrat)
Regresori externi	Nu	Da
Changepoints trend	Nu (neted)	Da (automat)
Date lipsă	Necesită interpolare	Gestionează nativ
Interpretabilitate	Moderată	Înaltă
Viteză calcul	Lent	Rapid
Date frecvență înaltă	Bun	Moderat
Perioade non-întregi	Da (ex: 365.25)	Da
Intervale incertitudine	Da	Da

Ghid selecție model



When to Use Each Model

SARIMA

- Single seasonality
- Regular data
- Statistical inference
- Short-term forecast

TBATS

- High frequency (hourly)
- Non-integer periods
- Automatic selection
- No external regressors

Prophet

- Business forecasting
- Holiday effects
- Missing data
- Changepoints trend
- External regressors

Când să folosim fiecare model

Folosiți TBATS când:

- Date de frecvență înaltă (orare, sub-zilnice)
- Multiple perioade sezoniere complexe
- Nu sunt necesari regresori externi
- Se preferă selecție automată a modelului
- Se dorește cadrul tradițional de tip spațiu al stărilor (*state-space*)

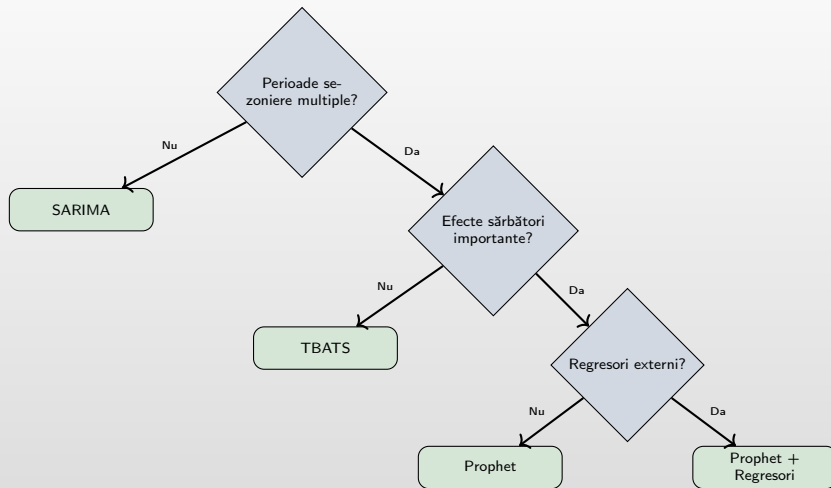
Folosiți Prophet când:

- Prognoză de business (zilnic/săptămânal)
- Efectele sărbătorilor sunt importante
- Trendul are rupturi structurale
- Sunt prezente date lipsă
- Interpretabilitatea este esențială
- Sunt disponibili regresori externi

Ghid general

- **Prophet:** pentru aplicații de business cu date zilnice
- **TBATS:** pentru aplicații tehnice cu date de frecvență înaltă

Diagramă de decizie



Metrici de evaluare

Metrici de acuratețe a prognozei

- ▣ *(Metricile RMSE, MAE, MAPE — definite în Capitolul 0)*

Acoperire

- ▣ Pentru intervalele de predicție $[\hat{y}_t^L, \hat{y}_t^U]$, rata de acoperire este proporția valorilor reale care se încadrează în interval
- ▣ Țintă: rata de acoperire să corespundă nivelului nominal (ex: 80%)

Studiu de caz: prognoza cererii de energie

Problema

- ▣ **Obiectiv:** Prognozați cererea de electricitate pe oră
- ▣ **Provocări:** tipar zilnic (vârf la prânz/seara), săptămânal (mai scăzut în weekend), anual (vara/iarna), efecte sărbători (cerere mai mică)

Abordare

- ▣ **Pas 1:** Încercați TBATS cu perioade [24, 168, 8766]
- ▣ **Pas 2:** Încercați Prophet cu sezonality zilnică, săptămânală, anuală + sărbători
- ▣ **Pas 3:** Comparați folosind cross-validation

Studiu de caz: interpretarea rezultatelor

Rezultate tipice (metrici definite anterior)

□ Comparație performanță:

Model	MAPE	RMSE	Acoperire
SARIMA (doar zilnic)	8.5%	450 MW	75%
TBATS	4.2%	220 MW	82%
Prophet	4.8%	250 MW	85%
Prophet + sărbători	3.9%	200 MW	88%

Concluzie

- Modelele cu sezonaliități multiple depășesc semnificativ SARIMA
- Gestionarea unei singure perioade sezoniere este insuficientă pentru date complexe

Exercițiu AI: Gândire critică

Prompt de testat în ChatGPT / Claude / Copilot

“Descarcă setul de date Peyton Manning Wikipedia pageviews din Prophet (sau folosește date zilnice de consum de electricitate din 2020-01-01 până în 2024-12-31, aprox. 1.800 observații). Folosește Prophet (Meta) pentru a prognoza următoarele 30 de zile. Include sărbătorile din SUA și componentele de sezonality săptămânală/anuală. Compară cu TBATS. Vreau cod Python complet.”

Exercițiu

1. Rulați prompt-ul într-un LLM (Large Language Model) la alegere și analizați critic răspunsul.
2. Prophet detectează automat sezonality multiple (zilnice, săptămânale)?
3. Cum sunt specificate sărbătorile? Specifice țării sau evenimente personalizate?
4. Folosește cross-validation cu puncte de referință (performance_metrics)?
5. TBATS ar fi mai potrivit pentru această frecvență? De ce sau de ce nu?

Atenție: Codul generat de AI poate produce cod funcțional cu aparență de corectitudine. *Asta nu înseamnă că e corect.*

NeuralProphet (Triebe et al., 2021)

TSA_ch9_neural_prophet

Arhitectură: descompunere Prophet + AR-Net

Modelul extinde descompunerea Prophet cu componente neuronale:

$$y(t) = T(t) + S(t) + E(t) + F(t) + A(t) + L(t)$$

- ▣ $T(t)$: trend liniar sau logistic pe segmente (*piecewise*) — ca în Prophet
- ▣ $S(t)$: sezonalitate Fourier — ca în Prophet
- ▣ $E(t)$: evenimente speciale / sărbători
- ▣ $F(t)$: regresori viitori (*future regressors* — covariabile cunoscute)
- ▣ $A(t)$: **AR-Net** — rețea neuronală *feedforward* pe lag-uri y_{t-1}, \dots, y_{t-p}
- ▣ $L(t)$: **regresori cu lag** — covariabile externe procesate prin NN separat

Training și avantaje

- ▣ Backend **PyTorch**: mini-batch SGD, selecție automată a ratei de învățare
- ▣ **Avantaj cheie** față de Prophet: captează tipare autoregresive pe care Prophet le ignoră
- ▣ Sezonalitate locală: coeficienți Fourier variabili în timp

NeuralProphet: Comparație și Rezultate

 TSA_ch9_neural_prophet

Prophet vs. NeuralProphet

Caracteristică	Prophet	NeuralProphet
Backend	Stan (MCMC/MAP)	PyTorch (SGD)
Autoregresie	Nu	Da (AR-Net)
Covariabile lag	Nu	Da (<i>Lagged Regressors</i>)
Viteză training	Lentă (MCMC)	Rapidă (SGD)
Interpretabilitate	Ridicată	Ridicată (componente)
Intervale de predicție	Bayesiene	Bootstrap / Conformal

Rezultate empirice

- Depășește Prophet pe seturi de date cu **structură AR puternică**
- Performanță comparabilă pe date cu trend dominant, sezonabilitate stabilă
- NeuralProphet mai potrivit pentru volume mari de date și covariabile multiple

Software

`pip install neuralprophet` — API similar cu Prophet, tranziție ușoară

`from neuralprophet import NeuralProphet; m = NeuralProphet()`

Predicție conformală (*Conformal Prediction*)

Q TSA_ch9_conformal_prediction

Problema

Intervalele tradiționale de predicție presupun o distribuție parametrică (de ex. gaussiană). În practică, această ipoteză este adesea încălcată.

Predicție conformală (Vovk, Gammerman & Shafer, 2005)

Metodă **distribution-free** cu garanție de acoperire pe eșantioane finite:

1. **Scor de neconformitate:** $\alpha_t = |y_t - \hat{y}_t|$ (sau orice scor bazat pe reziduuri)
2. **Set de calibrare:** se calculează scorurile pe date rezervate $\{\alpha_1, \dots, \alpha_n\}$
3. **Interval de predicție:**

$$\hat{C}_{T+1} = \hat{y}_{T+1} \pm Q_{1-\alpha}(\{\alpha_1, \dots, \alpha_n\})$$

unde $Q_{1-\alpha}$ este cuantila empirică de ordin $\lceil (1 - \alpha)(1 + 1/n) \rceil / n$

Garanție de acoperire

Sub ipoteza de **interschimbabilitate** (*exchangeability*): $\mathbb{P}(y_{T+1} \in \hat{C}_{T+1}) \geq 1 - \alpha$ — **exact**, nu asimptotic.

Predicție conformală adaptivă pentru serii de timp

TSA ch9 aci

Problema

Seriile de timp **încalcă** ipoteza de interschimbabilitate \Rightarrow garanția clasică nu se aplică direct.

Adaptive Conformal Inference — ACI (Gibbs & Candès, 2021)

- ▣ Nivel de semnificație **variabil în timp**: $\alpha_t = \alpha + \gamma(\text{err}_{t-1} - \alpha)$, $\text{err}_{t-1} = 1\{y_{t-1} \notin \hat{C}_{t-1}\}$
- ▣ Ajustează intervalele pe baza erorilor recente de acoperire
- ▣ Menține acoperire **asimptotică** chiar și sub deplasare de distribuție (*distribution shift*)

EnbPI — Ensemble Batch Prediction Intervals (Xu & Xie, 2022)

- ▣ Folosește **ansamblu bootstrap** de modele de bază
- ▣ Actualizează secvențial distribuția reziduurilor
- ▣ Acoperire validă fără ipoteză de interschimbabilitate

Software

MAPIE (Python) — intervale de predicție model-agnostice: `pip install mapie`

Aplicații: Intervale de predicție model-agnostice

TSA_ch9_enbpi

Flux de lucru

1. Antrenați **orice** model de prognoză (ARIMA, GARCH, XGBoost, LSTM, Prophet...)
2. Calculați reziduurile pe setul de calibrare
3. Aplicați calibrare conformală \Rightarrow intervale cu **garanție de acoperire**

Comparație cu abordările tradiționale

Metodă	Distribuție	Acoperire	Model-agnostic
Parametrică (ARIMA)	Gaussiană	Asimptotică	Nu
Bootstrap	Empirică	Asimptotică	Parțial
Conformală	Fără ipoteze	Finită (exactă)	Da

Aplicații practice și software

- ▣ **Domenii:** risc financiar (VaR), prognoză energetică, planificarea cererii
- ▣ **Biblioteci Python:** mapie, conformal-tights, crepes

Concluzii

Sezonalități multiple

- ▣ Datele din lumea reală au adesea tipare sezoniere multiple
- ▣ SARIMA standard gestionează doar o perioadă sezonieră
- ▣ TBATS și Prophet sunt proiectate pentru această provocare

Selecția modelului

- ▣ **TBATS**: Automat, gestionează frecvență înaltă, fără regresori externi
- ▣ **Prophet**: Interpretabil, efecte sărbători, regresori externi
- ▣ **NeuralProphet**: Extinde Prophet cu AR-Net și covariabile lag (PyTorch)

De reținut

- ▣ **Predicție conformală**: intervale distribution-free, model-agnostice, cu garanție finită de acoperire — completează orice model de prognoză.

Întrebări?

Întrebări?

Pași următori

- ▣ Exersați cu notebook-ul Jupyter
- ▣ Încercați Prophet și NeuralProphet pe propriile date
- ▣ Aplicați predicție conformală (MAPIE) pentru intervale model-agnostice

Întrebarea 1

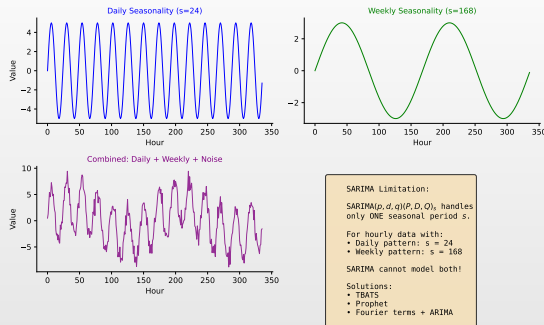
Întrebare

- De ce nu poate $SARIMA(p, d, q)(P, D, Q)_s$ standard să modeleze date orare de electricitate cu tipare zilnice și săptămânale?

Variante de răspuns

- (A) SARIMA poate gestiona doar o singură perioadă sezonieră s la un moment dat
- (B) SARIMA necesită erori normal distribuite pentru sezonaliități multiple
- (C) SARIMA poate gestiona sezonaliități multiple, dar necesită mai multe date
- (D) SARIMA funcționează doar cu date lunare sau trimestriale

Întrebarea 1: Răspuns



Răspuns: (A)

- ☐ SARIMA gestionează doar **o singură** perioadă sezonieră s
- ☐ Nu se pot seta $s = 24$ (zilnic) și $s = 168$ (săptămânal) simultan într-un singur model SARIMA

Întrebarea 2

Întrebare

□ Ce reprezintă fiecare literă din TBATS?

Variante de răspuns

- (A) Trend, Bayes, Autoregresiv, Timp, Staționaritate
- (B) Sezonaliitate Trigonometrică, Box-Cox, erori ARMA, Trend, componente Sezoniere
- (C) Taylor, Box-Cox, ARIMA, Transformare, Smoothing
- (D) Trigonometric, Bayesian, ARMA, Trend, Analiză spectrală

Întrebarea 2: Răspuns

TBATS: What Does It Stand For?

T	Trigonometric	Fourier terms for seasonality $\sum [a_n \cos(\frac{2\pi n t}{m}) + b_n \sin(\frac{2\pi n t}{m})]$
B	Box-Cox	Variance stabilization $y^{(\omega)} = (y^\omega - 1)/\omega$
A	ARMA	Error autocorrelation $\phi(L)d_t = \theta(L)\varepsilon_t$
T	Trend	Level + slope (possibly damped) $\ell_t = \ell_{t-1} + \phi b_{t-1}$
S	Seasonal	Multiple seasonal periods m_1, m_2, \dots, m_T

Răspuns: (B)

- ☐ Sezonalitate Trigonometrică, transformare Box-Cox, erori ARMA, Trend, componente Sezoniere.

Întrebarea 3

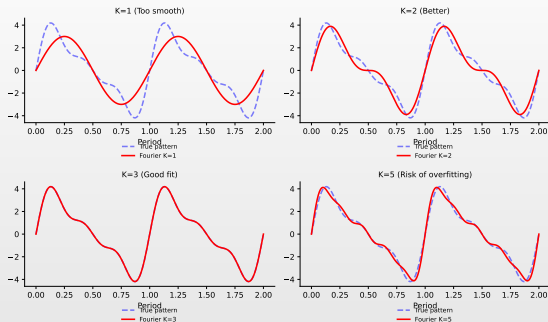
Întrebare

□ Ce se întâmplă când creștem numărul de armonici Fourier K ?

Variante de răspuns

- (A) Modelul devine mai simplu și mai robust
- (B) Modelul captează tipare sezoniere mai complexe, dar riscă supraajustarea
- (C) Orizontul de prognoză crește proporțional
- (D) Perioada sezonieră s se schimbă automat

Întrebarea 3: Răspuns



Răspuns: (B)

- Un K mai mare captează tipare sezoniere mai complexe, dar crește riscul de supraajustare. Maximul este $K \leq s/2$.

Întrebarea 4

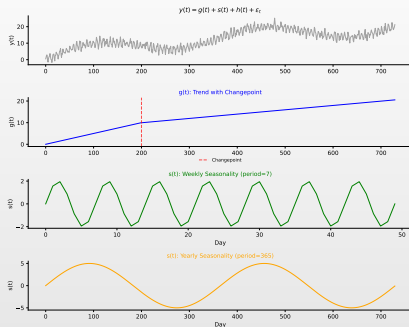
Întrebare

□ Care sunt componentele principale în modelul Prophet $y(t) = g(t) + s(t) + h(t) + \varepsilon_t$?

Variante de răspuns

- (A) $g(t)$ = volatilitate GARCH, $s(t)$ = test de staționaritate, $h(t)$ = heteroscedasticitate
- (B) $g(t)$ = creștere (trend cu changepoints), $s(t)$ = sezonaliitate, $h(t)$ = efecte de sărbători
- (C) $g(t)$ = zgomot gaussian, $s(t)$ = netezire, $h(t)$ = termeni armonici
- (D) $g(t)$ = gradient, $s(t)$ = densitate spectrală, $h(t)$ = exponent Hurst

Întrebarea 4: Răspuns



Răspuns: (B)

- $g(t)$ = trend cu changepoints, $s(t)$ = sezonaliitate (termeni Fourier), $h(t)$ = efecte de sărbători, ε_t = termen de eroare.

Întrebarea 5

Întrebare

☐ Ce caracteristici distinctive are Prophet pe care TBATS nu le are?

Variante de răspuns

- (A) Sezonabilitate trigonometrică și transformare Box-Cox
- (B) Selecție automată a parametrilor și netezire exponențială
- (C) Efecte de sărbători, regresori externi, changepoints în trend și gestionare nativă a datelor lipsă
- (D) Formulare spațiu-stare și modelare erori ARMA

Întrebarea 5: Răspuns

TBATS vs Prophet: Head-to-Head Comparison

Feature	TBATS	Prophet
Multiple seasonalities	Yes (automatic)	Yes (manual/auto)
Holiday effects	No	Yes (built-in)
External regressors	No	Yes
Trend changepoints	No (smooth)	Yes (automatic)
Missing data	Needs interpolation	Handles natively
Interpretability	Moderate	High
Computation speed	Slow	Fast
High-frequency data	Good	Moderate
Non-integer periods	Yes (e.g., 365.25)	Yes
Best for	Technical/high-freq	Business/daily

Răspuns: (C)

- ☐ Prophet oferă efecte de sărbători, regresori externi, changepoints în trend și gestionare nativă a datelor lipsă
- ☐ Aceste caracteristici sunt indisponibile în TBATS

Bibliografie I

Prophet

- Taylor, S.J., & Letham, B. (2018). Forecasting at Scale, *The American Statistician*, 72(1), 37–45.
- Harvey, A.C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press.

TBATS și netezire exponențială

- De Livera, A.M., Hyndman, R.J., & Snyder, R.D. (2011). Forecasting Time Series with Complex Seasonal Patterns Using Exponential Smoothing, *JASA*, 106(496), 1513–1527.
- Hyndman, R.J., Koehler, A.B., Ord, J.K., & Snyder, R.D. (2008). *Forecasting with Exponential Smoothing: The State Space Approach*, Springer.
- Taylor, J.W. (2003). Short-term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing, *Journal of the Operational Research Society*, 54(8), 799–805.

Bibliografie II

Comparații și competiții de prognoză

- ▣ Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition, *International Journal of Forecasting*, 36(1), 54–74.
- ▣ Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*, 3rd ed., OTexts.
- ▣ Petropoulos, F., et al. (2022). Forecasting: Theory and Practice, *International Journal of Forecasting*, 38(3), 845–1054.

Resurse online și cod


- ▣ **Quantlet:** <https://quantlet.com> – Platformă de cod pentru metode cantitative
- ▣ **Quantinar:** <https://quantinar.com> – Platformă de învățare pentru metode cantitative
- ▣ **GitHub TSA:** https://github.com/QuantLet/TSA/tree/main/TSA_ch9 – Cod Python pentru acest capitol

Vă Mulțumim!

Întrebări?

Materialele cursului sunt disponibile la: <https://danpele.github.io/Time-Series-Analysis/>

 Quantlet

 Quantinar