



Analiza și Prognoza Seriilor de Timp

# Capitolul 9: Prophet și TBATS

Seminar



# Cuprins Seminar

- 1 Test de Recapitulare
- 2 Întrebări Adevărat/Fals
- 3 Probleme Practice
- 4 Exemple Rezolvate
- 5 Subiecte de Discuție
- 6 Exerciții pentru Studiu Individual

## Test 1: Problema Sezonality Multiple

### Întrebare

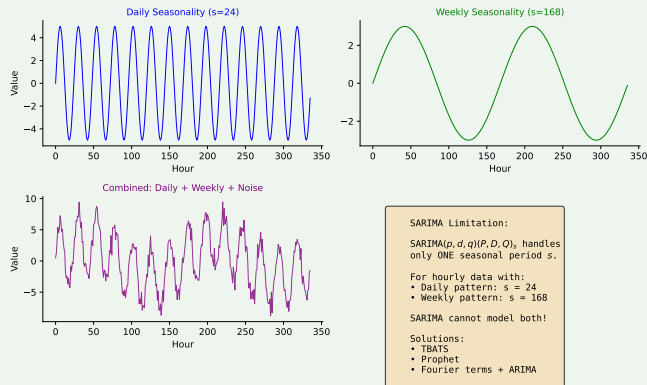
De ce modelele SARIMA standard nu pot gestiona datele orare de consum electric?

- ☐ A) SARIMA poate gestiona doar date lunare
- ☐ B) SARIMA permite o singură perioadă sezonieră (parametrul  $m$ )
- ☐ C) SARIMA nu suportă componente de trend
- ☐ D) SARIMA necesită date normal distribuite

*Răspunsul pe slide-ul următor...*

## Test 1: Răspuns

Răspuns: B – SARIMA permite o singură perioadă sezonieră



**Esențial:** Datele orare au pattern-uri zilnice (24h), săptămânale (168h) și anuale (8760h). Parametrul unic  $m$  din SARIMA nu poate captura simultan toate acestea.

## Test 2: Acronimul TBATS

### Întrebare

Ce reprezintă TBATS?

- A) Trend, Baseline, ARMA, Transform, Seasonal
- B) Trigonometric, Box-Cox, ARMA, Trend, Seasonal
- C) Time-Based Automatic Time Series
- D) Temporal Bayesian Adaptive Trend System

*Răspunsul pe slide-ul următor...*

### Răspuns: B – Trigonometric, Box-Cox, ARMA, Trend, Seasonal

#### TBATS: What Does It Stand For?

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

#### Componentele TBATS:

- Trigonometric: Termeni Fourier pentru sezonalitate
- Box-Cox: Stabilizarea varianței
- ARMA: Autocorelația erorilor

### Întrebare

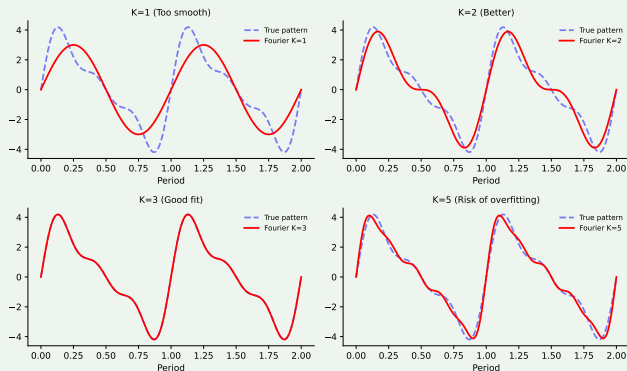
În TBATS, creșterea numărului de armonice Fourier ( $K$ ) pentru un pattern sezonier:

- A) Îmbunătățește întotdeauna acuratețea prognozei
- B) Permite forme sezoniere mai flexibile (complexe)
- C) Reduce complexitatea modelului
- D) Elimină necesitatea transformării Box-Cox

*Răspunsul pe slide-ul următor...*

## Test 3: Răspuns

### Răspuns: B – Permite forme sezoniere mai flexibile



**Compromis:** Mai multe armonice = mai multă flexibilitate dar și mai mulți parametri.

$$s_t^{(i)} = \sum_{j=1}^{K_i} \left[ a_j^{(i)} \cos\left(\frac{2\pi j t}{m_i}\right) + b_j^{(i)} \sin\left(\frac{2\pi j t}{m_i}\right) \right]$$



## Test 4: Descompunerea Prophet

### Întrebare

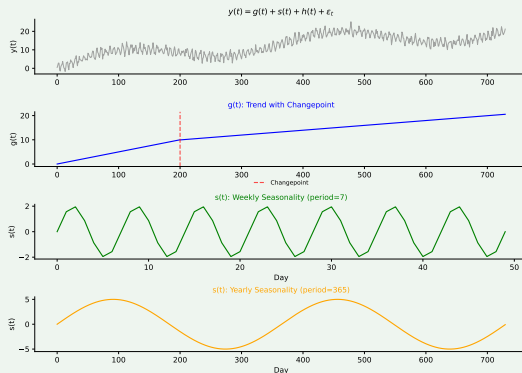
Prophet descompune o serie de timp în care componente?

- ☐ A) Componente AR, MA și sezoniere
- ☐ B) Trend, sezonalitate, sărbători și eroare
- ☐ C) Medie, varianță și autocorelație
- ☐ D) Nivel, pantă și curbura

*Răspunsul pe slide-ul următor...*

## Test 4: Răspuns

### Răspuns: B – Trend, sezonalitate, sărbători și eroare



**Modelul Prophet:**  $y(t) = g(t) + s(t) + h(t) + \epsilon_t$

- $g(t)$ : Trend (liniar pe porțiuni sau creștere logistică)
- $s(t)$ : Sezonalitate (serii Fourier)
- $h(t)$ : Efecte de sărbători

## Test 5: Prophet vs TBATS

### Întrebare

Când ați alege Prophet în locul TBATS?

- ☐ A) Când aveți nevoie de selecție automată a modelului
- ☐ B) Când aveți sărbători și puncte de schimbare cunoscute de încorporat
- ☐ C) Când aveți nevoie de cel mai simplu model
- ☐ D) Când datele nu au trend

*Răspunsul pe slide-ul următor...*

### Răspuns: B – Sărbători și puncte de schimbare cunoscute

**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

**Avantaje Prophet:** Integrare ușoară a sărbătorilor, analyst-in-the-loop, gestionează date lipsă, componente interpretabile.

**Avantaje TBATS:** Selecție automată a modelului, gestionează sezonalitate complexă fără expertiză de domeniu.

### Întrebare

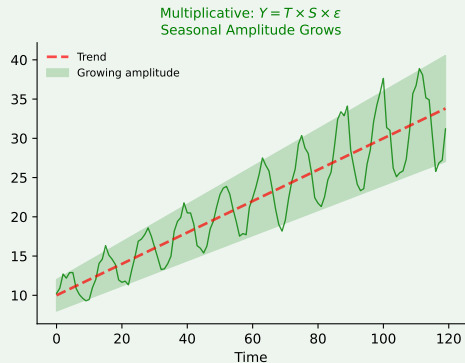
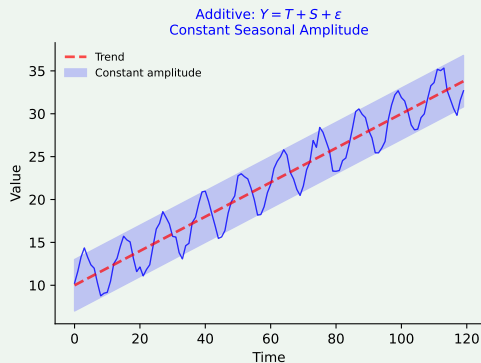
Pentru date de vânzări retail unde vânzările din decembrie sunt de 3 ori media lunară, ce mod de sezonality este mai potrivit în Prophet?

- ☐ A) Sezonality aditivă
- ☐ B) Sezonality multiplicativă
- ☐ C) Ambele funcționează la fel de bine
- ☐ D) Niciunul—folosiți ARIMA în schimb

*Răspunsul pe slide-ul următor...*

## Test 6: Răspuns

### Răspuns: B – Sezonalitate multiplicativă



**Esențial:** Când amplitudinea sezonieră se scalează cu nivelul, folosiți multiplicativ.

**Aditiv:**  $y = g(t) + s(t)$  (efect sezonier constant)

**Multiplicativ:**  $y = g(t) \cdot (1 + s(t))$  (efect sezonier proporțional)

## Test 7: Punctele de Schimbare în Prophet

### Întrebare

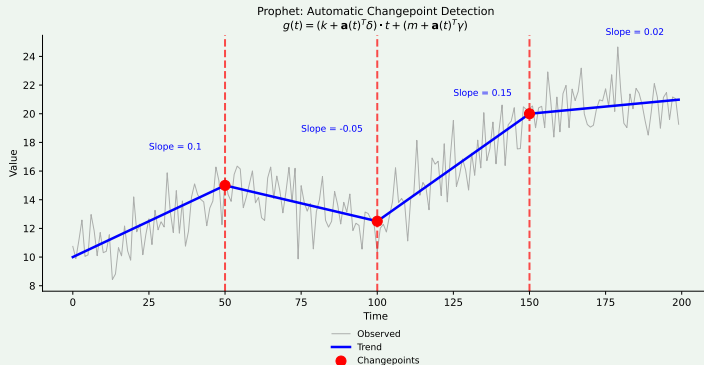
În Prophet, punctele de schimbare (changepoints) permit modelului să:

- ☐ A) Schimbe automat perioada sezonieră
- ☐ B) Ajusteze panta trendului în puncte specifice în timp
- ☐ C) Comute între modurile aditiv și multiplicativ
- ☐ D) Detecteze și elimine valorile aberante

*Răspunsul pe slide-ul următor...*

## Test 7: Răspuns

### Răspuns: B – Ajustează panta trendului în puncte specifice



**Puncte de schimbare:** Permit trend liniar pe porțiuni cu pante diferite.

$$g(t) = (k + \mathbf{a}(t)^T \delta) \cdot t + (m + \mathbf{a}(t)^T \gamma)$$

Prophet detectează automat punctele de schimbare sau le puteți specifica manual.



## Test 8: Selecția Modelului

### Întrebare

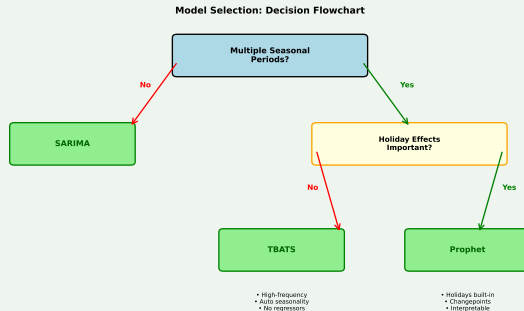
Aveți date zilnice de call center cu doar sezonality săptămânală. Ce model este cel mai potrivit?

- ☐ A) TBATS (proiectat pentru sezonality multiplă)
- ☐ B) Prophet (gestionează bine orice sezonality)
- ☐ C) SARIMA standard (mai simplu și suficient)
- ☐ D) Rețea neuronală LSTM (cea mai flexibilă)

*Răspunsul pe slide-ul următor...*

## Test 8: Răspuns

Răspuns: C – SARIMA standard este suficient



**Principiul parsimoniei:** Folosiți cel mai simplu model care se potrivește datelor.  
Cu doar sezonaltate săptămânală ( $m = 7$ ), SARIMA funcționează bine.

Folosiți TBATS/Prophet când *aveți nevoie* de sezonaltăți multiple sau funcționalități speciale.

## Test 9: Incertitudinea în Prophet

### Întrebare

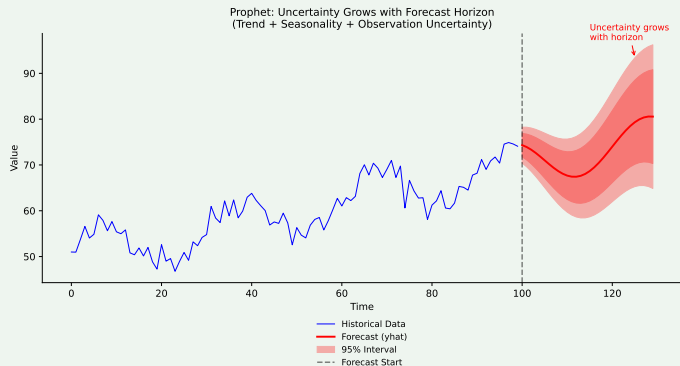
Prophet generează intervale de predicție prin:

- ☐ A) Presupunând reziduri normal distribuite
- ☐ B) Eșantionând din distribuția posterioară a parametrilor
- ☐ C) Folosind reeșantionare bootstrap a erorilor istorice
- ☐ D) Aplicând un multiplicator fix la prognozele punctuale

*Răspunsul pe slide-ul următor...*

## Test 9: Răspuns

### Răspuns: B – Eșantionează din distribuția posterioară



Prophet folosește estimare Bayesiană:

- Estimare MAP pentru prognoze punctuale
- MCMC sau simulare pentru intervale de incertitudine
- Incertitudine atât din trend (changepoints) cât și din zgomotul observațiilor

### Întrebare

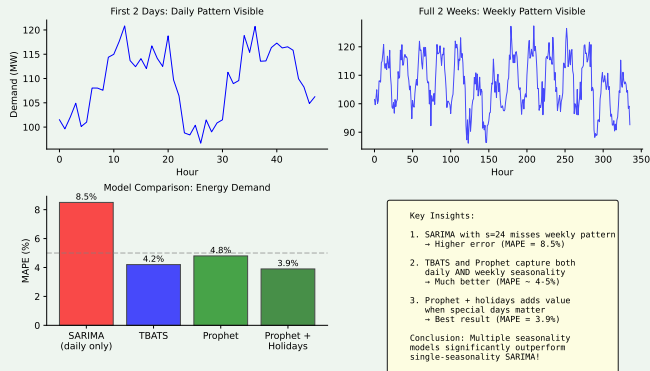
Pentru prognoza consumului orar de energie cu pattern-uri zilnice, săptămânale și anuale plus efecte de sărbători, care abordare este cea mai bună?

- ☐ A) SARIMA cu  $m = 24$
- ☐ B) TBATS cu trei perioade sezoniere
- ☐ C) Prophet cu sărbători personalizate
- ☐ D) Fie TBATS fie Prophet, în funcție de importanța sărbătorilor

*Răspunsul pe slide-ul următor...*

## Test 10: Răspuns

### Răspuns: D – TBATS sau Prophet în funcție de necesități



Ambele pot gestiona sezonabilitate multiplă:

- Dacă efectele sărbătorilor sunt cruciale ⇒ **Prophet** (modelare explicită)
- Dacă preferați selecție automată ⇒ **TBATS**
- Adesea: Încercați ambele și comparați prin validare încrucișată

Determinați dacă fiecare afirmație este Adevărată sau Falsă:

- ❶ Prophet a fost dezvoltat de Facebook (Meta) pentru prognoza de business.
- ❷ TBATS poate gestiona cel mult două perioade sezoniere.
- ❸ În Prophet, trendul implicit este creștere logistică.
- ❹ Termenii Fourier aproximează sezonabilitatea folosind funcții sinus și cosinus.
- ❺ Prophet necesită date de serii de timp echidistante.
- ❻ Transformarea Box-Cox din TBATS stabilizează varianța.

*Răspunsurile pe slide-ul următor...*

## Adevărat/Fals: Soluții

- ❶ Prophet a fost dezvoltat de Facebook (Meta) pentru prognoza de business.

Lansat în 2017, proiectat pentru prognoză „analyst in the loop” la scară largă.

ADEVĂRAT

- ❷ TBATS poate gestiona cel mult două perioade sezoniere.

TBATS poate gestiona orice număr de perioade sezoniere (ex. zilnic, săptămânal, anual).

FALS

- ❸ În Prophet, trendul implicit este creștere logistică.

Implicit este liniar pe porțiuni. Creșterea logistică trebuie specificată explicit.

FALS

- ❹ Termenii Fourier aproximează sezonabilitatea folosind funcții sinus și cosinus.

$$s(t) = \sum_{k=1}^K [a_k \cos(2\pi kt/m) + b_k \sin(2\pi kt/m)]$$

ADEVĂRAT

- ❺ Prophet necesită date de serii de timp echidistante.

Prophet gestionează elegant datele lipsă și timestamp-urile neregulate.

FALS

- ❻ Transformarea Box-Cox din TBATS stabilizează varianța.

$$y^{(\lambda)} = (y^\lambda - 1)/\lambda \text{ pentru } \lambda \neq 0; \log(y) \text{ pentru } \lambda = 0.$$

ADEVĂRAT



## Problemă 1: Calculul Termenilor Fourier

### Enunț

Pentru date zilnice cu sezonalitate săptămânală ( $m = 7$ ), doriți să folosiți termeni Fourier cu  $K = 3$  armonice. Câți parametri adaugă aceasta la model?

*Răspunsul pe slide-ul următor...*

## Problemă 1: Soluție

### Soluție: 6 parametri

Fiecare armonică necesită 2 parametri (coeficienți sinus și cosinus):

$$s(t) = \sum_{k=1}^K \left[ a_k \cos \left( \frac{2\pi kt}{m} \right) + b_k \sin \left( \frac{2\pi kt}{m} \right) \right]$$

Cu  $K = 3$  armonice:

- $k = 1$ :  $a_1, b_1$  (frecvența fundamentală)
- $k = 2$ :  $a_2, b_2$  (prima armonică)
- $k = 3$ :  $a_3, b_3$  (a doua armonică)

**Total:**  $2 \times K = 2 \times 3 = 6$  parametri

**Notă:** Maximum util  $K = \lfloor m/2 \rfloor = 3$  pentru  $m = 7$ .

## Problemă 2: Alegerea Modului de Sezonalitate

### Enunț

Proгноzați rezervările lunare la hotel. Datele arată:

- Iulie 2020: 1000 rezervări (sezon de vârf)
- Ianuarie 2020: 400 rezervări (extrasezon)
- Iulie 2023: 2000 rezervări (sezon de vârf)
- Ianuarie 2023: 800 rezervări (extrasezon)

Ar trebui să folosiți sezonalitate aditivă sau multiplicativă? De ce?

*Răspunsul pe slide-ul următor...*

## Problemă 2: Soluție

### Soluție: Sezonalitate multiplicativă

**Analiză:** Verificați dacă amplitudinea sezonieră este proporțională cu nivelul.

An	Iulie	Ianuarie	Raport (Iul/Ian)
2020	1000	400	2.5
2023	2000	800	2.5

**Observație cheie:** *Raportul* rămâne constant (2.5), nu diferența!

- Aditiv ar însemna: Iulie întotdeauna +600 față de Ianuarie
- Dar 2020:  $1000 - 400 = 600$ ; 2023:  $2000 - 800 = 1200$

**Concluzie:** Folosiți multiplicativ: `seasonality_mode='multiplicative'`

## Problemă 3: Interpretarea Modelului TBATS

### Enunț

Un model TBATS ajustat pe date orare de electricitate raportează:

- Box-Cox  $\lambda = 0.5$
- Perioade sezoniere:  $m_1 = 24$ ,  $m_2 = 168$
- Termeni Fourier:  $K_1 = 5$ ,  $K_2 = 3$

Ce ne spune fiecare componentă despre date?

*Răspunsul pe slide-ul următor...*

## Problemă 3: Soluție

### Soluție

**Box-Cox**  $\lambda = 0.5$ :

- Transformare rădăcină pătrată aplicată
- Datele aveau varianță crescătoare cu nivelul
- Transformare:  $y^{(0.5)} = \sqrt{y}$

**Perioade sezoniere:**

- $m_1 = 24$ : Pattern zilnic (24 ore)
- $m_2 = 168$ : Pattern săptămânal ( $7 \times 24 = 168$  ore)

**Termeni Fourier:**

- $K_1 = 5$  pentru zilnic: Pattern intrazilnic complex (5 armonice captează vârfuri, văi)
- $K_2 = 3$  pentru săptămânal: Pattern săptămânal mai simplu (zi lucrătoare vs weekend)

**Total parametri sezonieri:**  $2(K_1 + K_2) = 2(5 + 3) = 16$

## Problemă 4: Efectele Sărbătorilor în Prophet

### Enunț

Prognozați veniturile zilnice ale unui restaurant. Doriți să adăugați aceste efecte de sărbători în Prophet:

- Ziua Îndrăgostiților (14 Feb) – creștere majoră
- Paște (dată variabilă) – restaurant închis
- Crăciun (25 Dec) – restaurant închis

Scrieți codul Python pentru a crea dataframe-ul de sărbători pentru 2024-2025.

*Răspunsul pe slide-ul următor...*

## Problemă 4: Soluție

### Soluție

```
import pandas as pd
from prophet import Prophet
holidays = pd.DataFrame({
    'holiday':  ['valentines', 'paste', 'craciun'],
    'ds':       pd.to_datetime([...]),
    'lower_window': [0, 0, 0],
    'upper_window': [0, 0, 0]
})
model = Prophet(holidays=holidays)
model.fit(df)
```

**Notă:** Folosiți `lower_window=-1`, `upper_window=1` pentru a captura efecte pe zile adiacente.



# Exemplu: Prognoza Vânzărilor Retail cu Prophet

## Scenariu

Date lunare de vânzări retail (2018-2023) cu:

- Vârfuri puternice în decembrie (cumpărături de Crăciun)
- Impact COVID-19 în 2020 (ruptură structurală)
- Trend crescător în timp

## Configurare Prophet

```
model = Prophet(  
    seasonality_mode='multiplicative',  
    changepoint_prior_scale=0.5,  
    yearly_seasonality=True)  
model.add_country_holidays(country_name='RO')
```

## Decizie Cheie

Sezonalitate multiplicativă: efectul decembrie este proporțional cu nivelul de bază.

# Exemplu: Cererea de Energie cu TBATS

## Scenariu

Cerere orară de electricitate cu:

- Pattern intrazilnic (24 ore)
- Pattern săptămânal (168 ore)
- Pattern anual (8760 ore)

## TBATS în R

```
library(forecast)
energy_msts <- msts(energy_data,
  seasonal.periods = c(24, 168, 8760))
fit <- tbats(energy_msts)
fc <- forecast(fit, h = 168)
```

## Notă

TBATS selectează automat  $K$  pentru fiecare perioadă sezonieră prin AIC.

# Exemplu: Comparație prin Validare Încrucișată

## Obiectiv

Comparați Prophet, TBATS și SARIMA pe 2 ani de date zilnice de vânzări.

## Validare Încrucișată Prophet

```
from prophet.diagnostics import cross_validation
df_cv = cross_validation(model, initial='365 days',
    period='90 days', horizon='30 days')
metrics = performance_metrics(df_cv)
```

## Rezultate Tipice

Model	MAPE	Timp de Calcul
SARIMA (doar săptămânal)	8.5%	Rapid
TBATS (săptămânal + anual)	6.2%	Moderat
Prophet (săptămânal + anual + sărbători)	5.8%	Rapid

# Discuție: Când să Folosim Care Model?

## Întrebare Cheie

Aveți o nouă sarcină de prognoză. Cum alegeți între SARIMA, TBATS și Prophet?

## Cadru de Decizie

### 1 Câte perioade sezoniere?

- Una  $\Rightarrow$  SARIMA poate fi suficient
- Multiple  $\Rightarrow$  TBATS sau Prophet

### 2 Aveți cunoștințe de domeniu de încorporat?

- Sărbători, evenimente, puncte de schimbare  $\Rightarrow$  Prophet
- Lăsați datele să vorbească  $\Rightarrow$  TBATS

### 3 Cerințe de interpretabilitate?

- Trebuie să explicați componentele  $\Rightarrow$  Prophet
- Doar prognoze  $\Rightarrow$  Oricare

### Întrebare Cheie

Puteți avea prea mulți termeni Fourier? Care sunt simptomele?

Răspuns: Da!

#### Simptome de supraajustare:

- Ajustare in-sample excelentă, dar out-of-sample slabă
- Sezonabilitatea arată „zimțată” sau nerealistă
- Prognozele oscilează puternic

### Ghiduri

- Maximum  $K \leq m/2$  (limita Nyquist)
- Începeți cu  $K = 3-5$  pentru majoritatea aplicațiilor
- Folosiți validare încrucișată pentru a selecta  $K$
- Implicit Prophet:  $K = 10$  pentru anual,  $K = 3$  pentru săptămânal

## Scenariu

Datele istorice includ perioada COVID-19 (2020-2021). Cum gestionați acest lucru când prognozați 2024?

## Opțiuni

- ❶ **Exclueți perioada COVID:** Antrenați doar pe date pre-COVID și post-COVID
- ❷ **Folosiți puncte de schimbare:** Lăsați Prophet să detecteze/specificați rupturi
- ❸ **Adăugați regresori:** Includeți variabilă indicator COVID
- ❹ **Ajustare:** Ajustați manual valorile 2020-2021 la „normal”

## Abordare Prophet

```
model = Prophet(changepoints=[...])  
df['covid'] = (df['ds'] >= '2020-03-15') & ...  
model.add_regressor('covid')
```

# Exerciții de Lucru Acasă

- 1 **Teoretic:** Demonstrați că  $K = m/2$  termeni Fourier pot reprezenta orice funcție periodică cu perioada  $m$  (pentru  $m$  par).
- 2 **Calcul:** Pentru pattern-ul sezonier de mai jos (date zilnice, ciclu săptămânal), determinați numărul minim de armonice Fourier necesare:

Lun: 100, Mar: 110, Mie: 115, Joi: 110, Vin: 120, Sâm: 80, Dum: 65

- 3 **Aplicat:** Descărcați date orare de cerere de electricitate dintr-o sursă publică:
  - Ajustați atât TBATS (în R) cât și Prophet (în Python)
  - Comparați acuratețea prognozei folosind RMSE și MAPE
  - Vizualizați descompunerile componentelor
- 4 **Gândire Critică:** De ce ar putea Prophet să performeze slab pe date financiare de înaltă frecvență (ex. prețuri de acțiuni minut cu minut)?

## Indicii

- ❶ Prin teorema Fourier, orice funcție periodică poate fi reprezentată ca sumă de sinusuri și cosinusuri. Cu perioada  $m$ , frecvențele sunt  $k/m$  pentru  $k = 1, \dots, m/2$ .
- ❷ Pattern-ul are:
  - Un vârf (Vineri) și o vale (Duminică)
  - Tranziții destul de netede
  - $K = 2$  sau  $K = 3$  probabil suficient (încercați și comparați)
- ❸ Pentru date de electricitate:
  - Includeți pattern-uri zilnice (24h) și săptămânale (168h)
  - Adăugați sărbătorile pentru regiunea dumneavoastră în Prophet
  - Așteptați-vă la MAPE în jur de 3-5% pentru prognoze orare
- ❹ Probleme cu datele financiare:
  - Fără sezonalitate clară (eficiența pieței)
  - Raport zgomot-semnal ridicat
  - Prophet proiectat pentru date „de business” cu trenduri și sezoane



## Modele pentru Sezonalitate Multiplă

- 1 **TBATS**: Automat, bazat pe Fourier, gestionează orice număr de perioade sezoniere
- 2 **Prophet**: Prietenos cu analistul, gestionare explicită a sărbătorilor/evenimentelor, interpretabil
- 3 **Folosiți SARIMA** când există doar o perioadă sezonieră

## Decizii Cheie

- **Modul de sezonalitate**: Aditiv (amplitudine constantă) vs Multiplicativ (proporțional)
- **Termeni Fourier**: Mai mulți = flexibil dar risc de supraajustare; folosiți CV pentru selecție
- **Puncte de schimbare**: Permit trendului să se adapteze la rupturi structurale

## Rețineți

**Prophet**: Excelent când aveți cunoștințe de domeniu de încorporat

**TBATS**: Excelent pentru modelare automată a sezonality complexe

# Vă Mulțumesc!

Întrebări?

`danpele@ase.ro`