

Tema 1

1. Boxploturi

Fig 1. Elemente principale boxplot

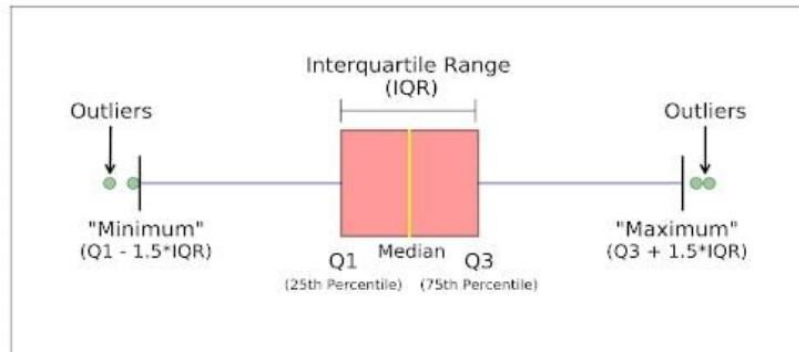
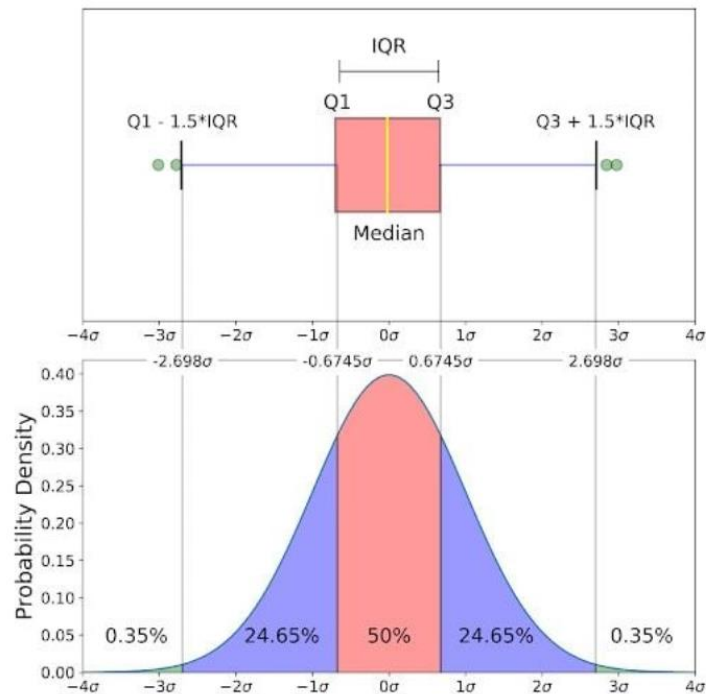


Fig 2. Interpretare boxplot dacă datele urmează o distribuție normală



2. Corelații liniare între variabile

Corelația liniară dintre două variabile poate fi:

- 1, în cazul unei corelații perfect pozitive
- -1, în cazul unei corelații perfect negative
- 0 sau aproape de 0 pentru nicio corelație

Fig 4. Tipuri de corelații liniare

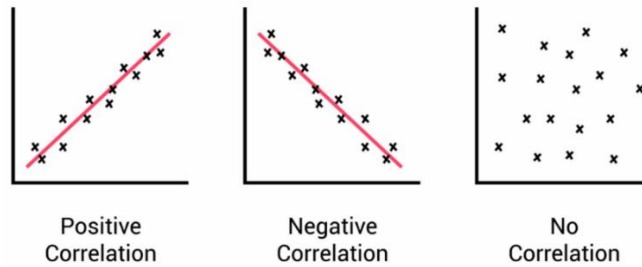


Fig 4. Un exemplu de matrice de corelație

	Connectivity	Digital Public Services	Human Capital	Integration of Digital Technology	Use of Internet
Connectivity	1.00	0.64	0.71	0.65	0.77
Digital Public Services	0.64	1.00	0.58	0.64	0.62
Human Capital	0.71	0.58	1.00	0.66	0.72
Integration of Digital Technology	0.65	0.64	0.66	1.00	0.60
Use of Internet	0.77	0.62	0.72	0.60	1.00

3. Modelul Prophet

Observații:

- Componentele unei serii de timp:
 - **Trend** – indică direcția generală a seriei de timp pe termen lung. Exemple: trend crescător, descrescător sau orizontal (staționar).
 - **Sezonalitate** - indică un pattern care se repetă relativ la timp, direcție și magnitudine în fiecare zi/săptămână/lună/semestru/an. Exemplu: creșterea temperaturii în timpul verii.
 - **Componenta ciclică** - indică un pattern care se repetă; diferă față de sezonalitate prin faptul că patternul se repetă o dată la câțiva ani, pe când la sezonalitate patternul se repetă în interiorul aceluiași an. Ciclurile nu au neapărat o lungime constantă.

- **Variații neregulate** - se referă la variații necontrolabile și inevitabile; sunt fluctuații nepredictibile care nu sunt asociate cu un pattern.

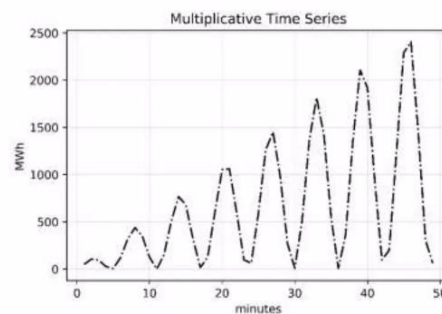
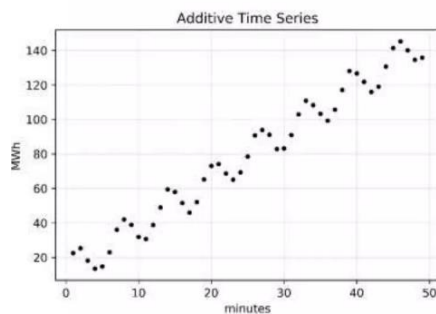
Pentru serii de timp se poate realiza descompunerea **ETS (Error/Random/Irregular, Trend, Seasonality)**

Descompunerile ETS au ca scop determinarea separată a variațiilor neregulate, a trendului și a sezonalițăților; ne ajută de exemplu să ne dăm seama care este trendul real, fără influențe din partea sezonalițății și sezonalițate reală, fără influențe din partea trendului.

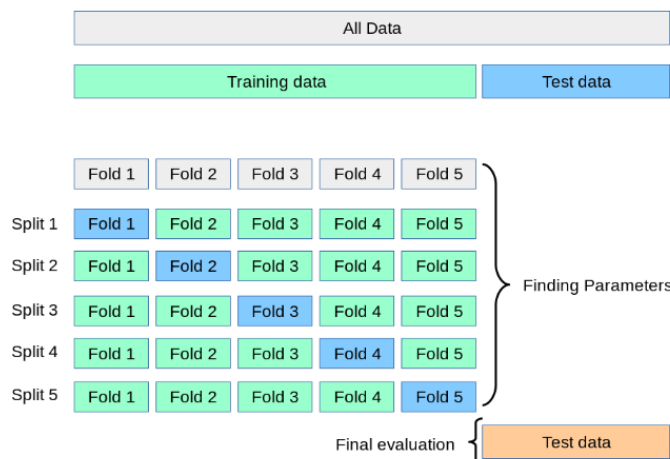
- Există 2 modele clasice de descompunere a seriilor de timp:
 - modelul aditiv $x_t = Trend + Sezonalițate + Error$
 - modelul multiplicativ $x_t = Trend * Sezonalițate * Error$

Modelul aditiv se utilizează când variația de sezonalițate este constantă în timp, cel multiplicativ se folosește când variația de sezonalițate se schimbă în timp.

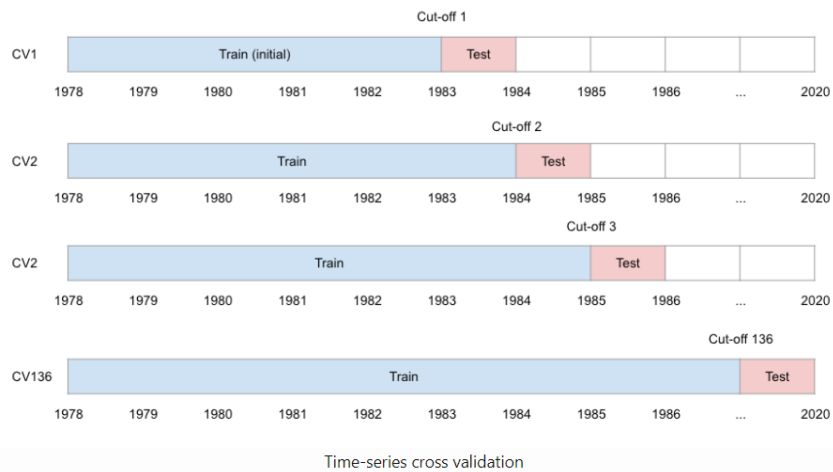
Seria de timp dată este aditivă



- Pentru setarea parametrilor, aveți grijă la următoarele aspecte:
 - Datasetul pe care îl folosiți conține date colectate din oră în oră**
 - Datasetul nu au sezonalițate pe săptămână/lună (ci doar pe zi și pe an).**
- Cross-validarea la modul general (k-fold cross-validation)



- Cross-validarea pentru serii de timp:



horizon = test size

initial = initial train size (increases with each fold)

period = how much train size increases with each fold

Tema 2

Fiecare concept dintr-o ontologie are asociat un URI (Uniform Resource Identifier). De obicei nu construim o ontologie de la zero, ci utilizăm concepte din ontologii bine cunoscute.

Concepte:

- **Class** ↔ concept general (e.g., clasa *Pisică*)
- Relații între clase. Clasele pot fi organizate în ierarhii (relația **SubClassOf**). Alte exemple de relații între clase: **EquivalentTo**, **DisjointWith**, **DisjointUnionOf**.
(e.g., *Pisică* **SubClassOf** *Animal*
Pisică **EquivalentTo** *FelisCatus*
Pisică **DisjointWith** *Câine*
Pisică **DisjointUnionOf** (*PisicaDomestică*, *PisicaSălbatică*)
- **Instance** ↔ concept specific, ce poate fi încadrat într-o clasă
(e.g., instanța *Tigruța* face parte din clasa *Pisică*
Tigruța **rdf:type** *Pisică*
)
- **Data property** ↔ relație definită între o instanță și un atribut (un număr/un șir de caractere/o data/...)
e.g., *Tigruța* **areVârsta** 2 (Domain: *Pisică*, Range: *xsd:int*)
e.g., *Tigruța* **areCuloareOchi** 'verde' (Domain: *Pisică*, Range: *xsd:string*)

Fiecare data property are un **Domain** (o clasă/ intersecția dintre mai multe clase) și un **Range** (care specifică tipul atributului, e.g., xsd:int, xsd:string, xsd:dateTime, etc.).

- **Object property** ↔ relație definită între o instanță și o altă instanță

Fiecare object property are un **Domain** (o clasă/ intersecția dintre mai multe clase) și un **Range** (o clasă/ intersecția dintre mai multe clase).

e.g. Dacă am avea clasa *Pisică* ce conține instanța *Tigruța* și clasa *Rasă* ce conține instanța *Europeană*, putem defini object property-ul *areRasa*, cu domeniul *Pisică* și range-ul *Rasă*:

Tigruța areRasa Europeană

Object properties pot avea asociate caracteristici precum simetrie, tranzitivitate, inversă, etc.

Putem defini ierarhii pentru data properties și pentru object properties.

Unul dintre avantajele ontologiilor este posibilitatea de a realiza inferențe (cu ajutorul unui reasoner). Reasonerul ne poate detecta și anumite inconsistențe din ontologie.

Indicație pentru temă: Pentru a vă asigura ca proprietățile simptomelor nu se amestecă între ele la asocierea cu boala, puteți introduce o clasă Simptom și puteți asocia fiecărui simptom caracteristici.

Alte resurse:

- [Ontology101 Guide](#)
- Alte exemple de ontologii (Pizza Ontology): [Detailed guide](#), Shorter explanations ([videos](#), [Github](#))