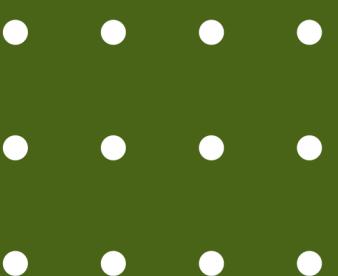


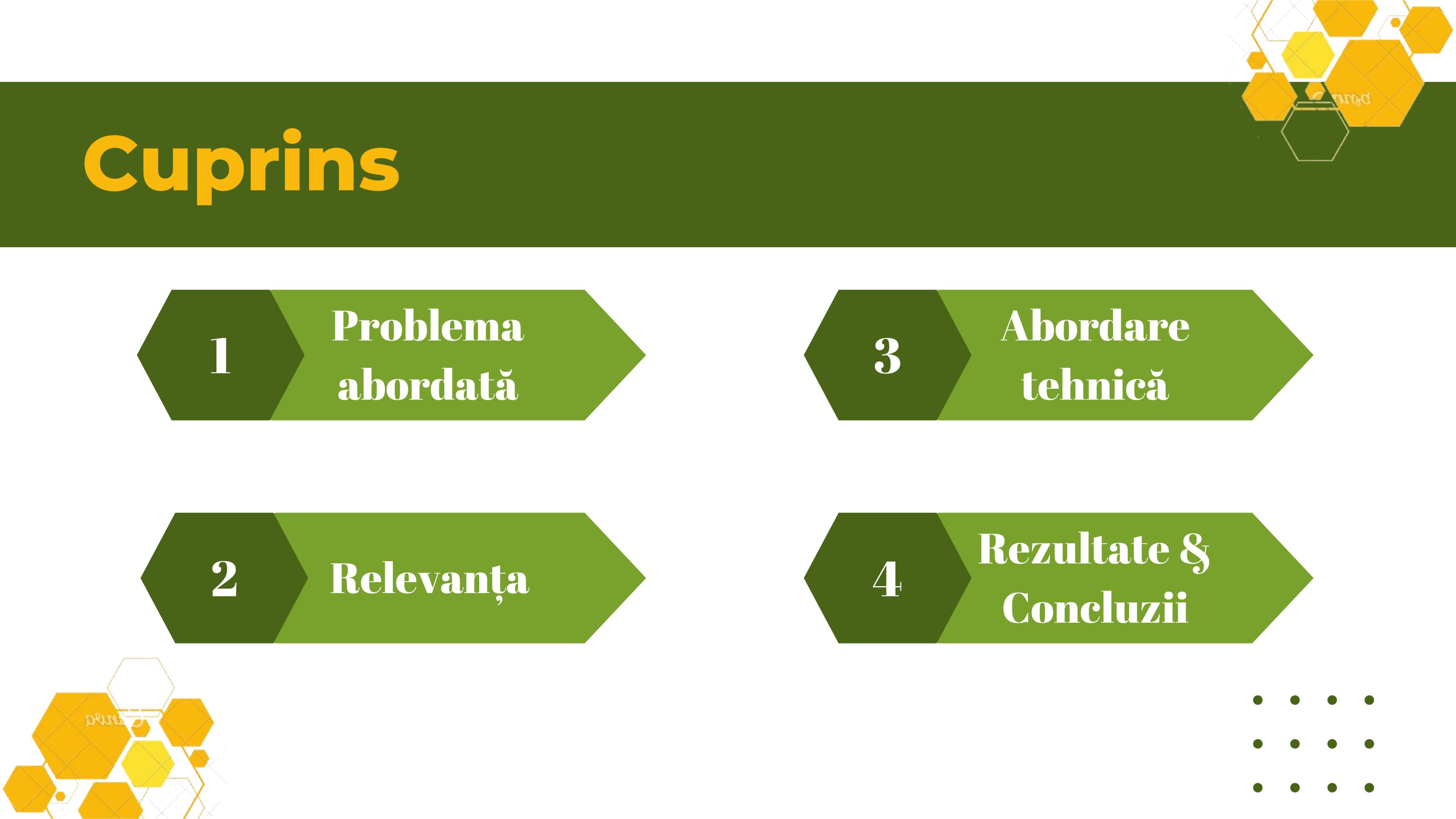
# Procesarea Semnalelor Dinamic Time Warping

Proiect realizat de:  
Andriciuc Andreea  
Dinu Florin  
Tudose Ștefan



# Cuprins

- 1 Problema abordată**
- 2 Relevanță**
- 3 Abordare tehnică**
- 4 Rezultate & Concluzii**

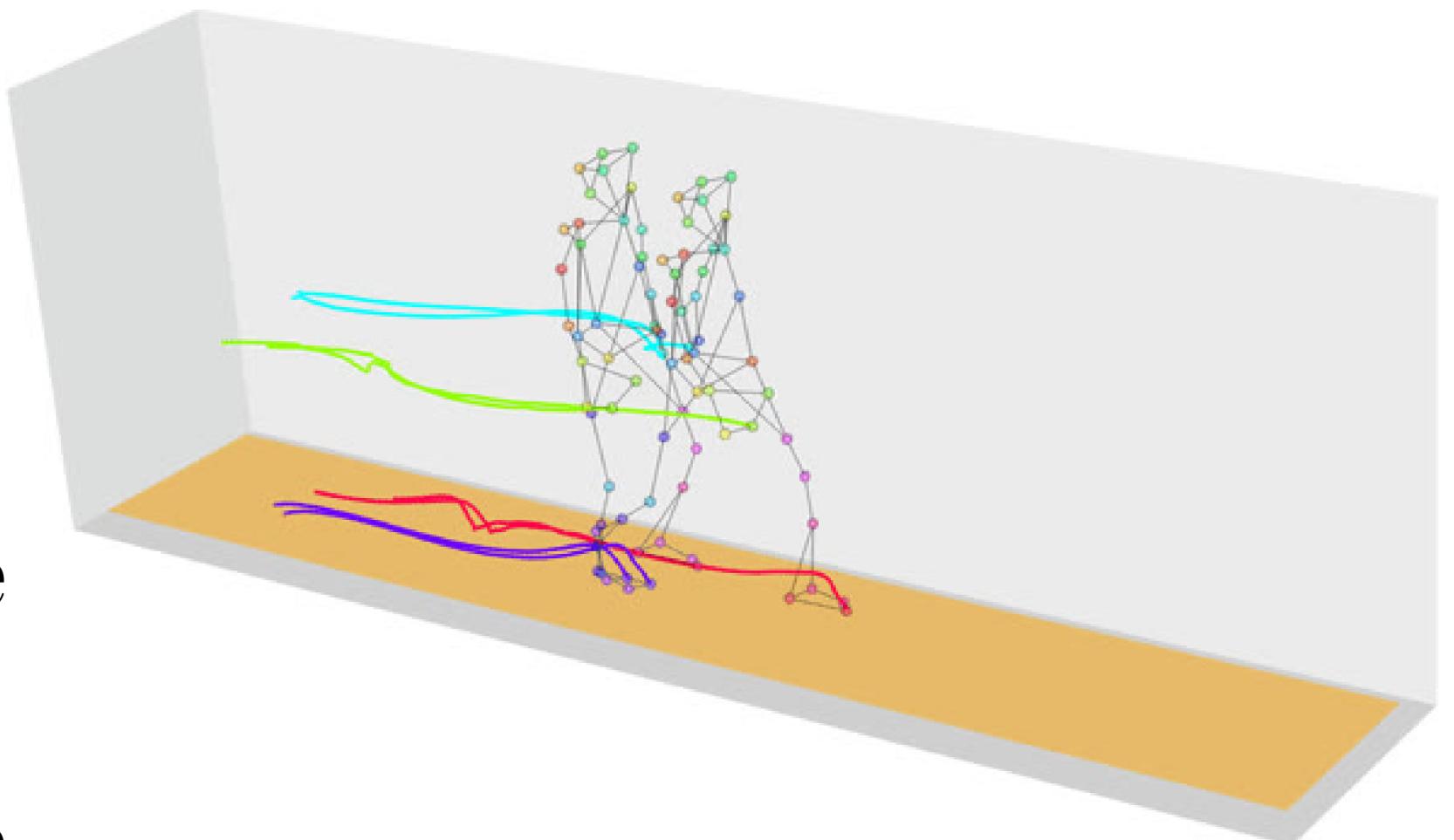




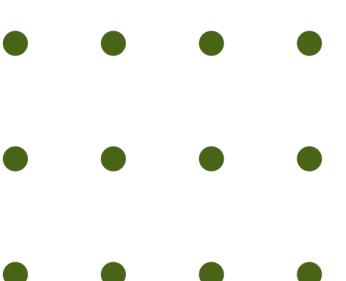
# Problema abordată

## Dinamic Time Warping:

- analizează similitudinea între două secvențe (de timp);
- aliniază în mod flexibil puncte de pe grafic;
- găsește „pattern-uri”.



Sursă: [Wikipedia](#)



# Relevantă

- Recunoașterea Automată a Vorbirii
- Recunoașterea Vorbitorului (indiferent de viteza discursului)
- Handwriting retrieval (recunoașterea semnăturii online)
- Prelucrarea a orice poate fi transformat într-o secvență liniară de timp (semnale video, audio etc)
- Text mining



• • •  
• • •  
• • •

# Reguli generale

Pentru a proteja forma seriilor de timp în urma mecanismului de dilatare și compresie al dimensiunii pe axa Ox, urmatoarele reguli au fost stabilite ca fiind universale pentru orice implementare de DTW:

• • •  
• • •  
• • •

01

• • •  
**Asocierea Reciprocă a Indicilor**

Fiecare indice din prima secvență trebuie asociat cu cel puțin un indice din cealaltă secvență, și invers.

02

**Asocierea Specifică a Primului și Ultimului Indice**

Primul indice din prima secvență trebuie asociat cu primul indice din cealaltă secvență, iar ultimul indice din prima secvență trebuie asociat cu ultimul indice din cealaltă secvență.

03

**Cerința de Monotonie în Asocierea Indicilor**

Asocierea indicilor dintre cele două secvențe trebuie să fie monoton crescătoare

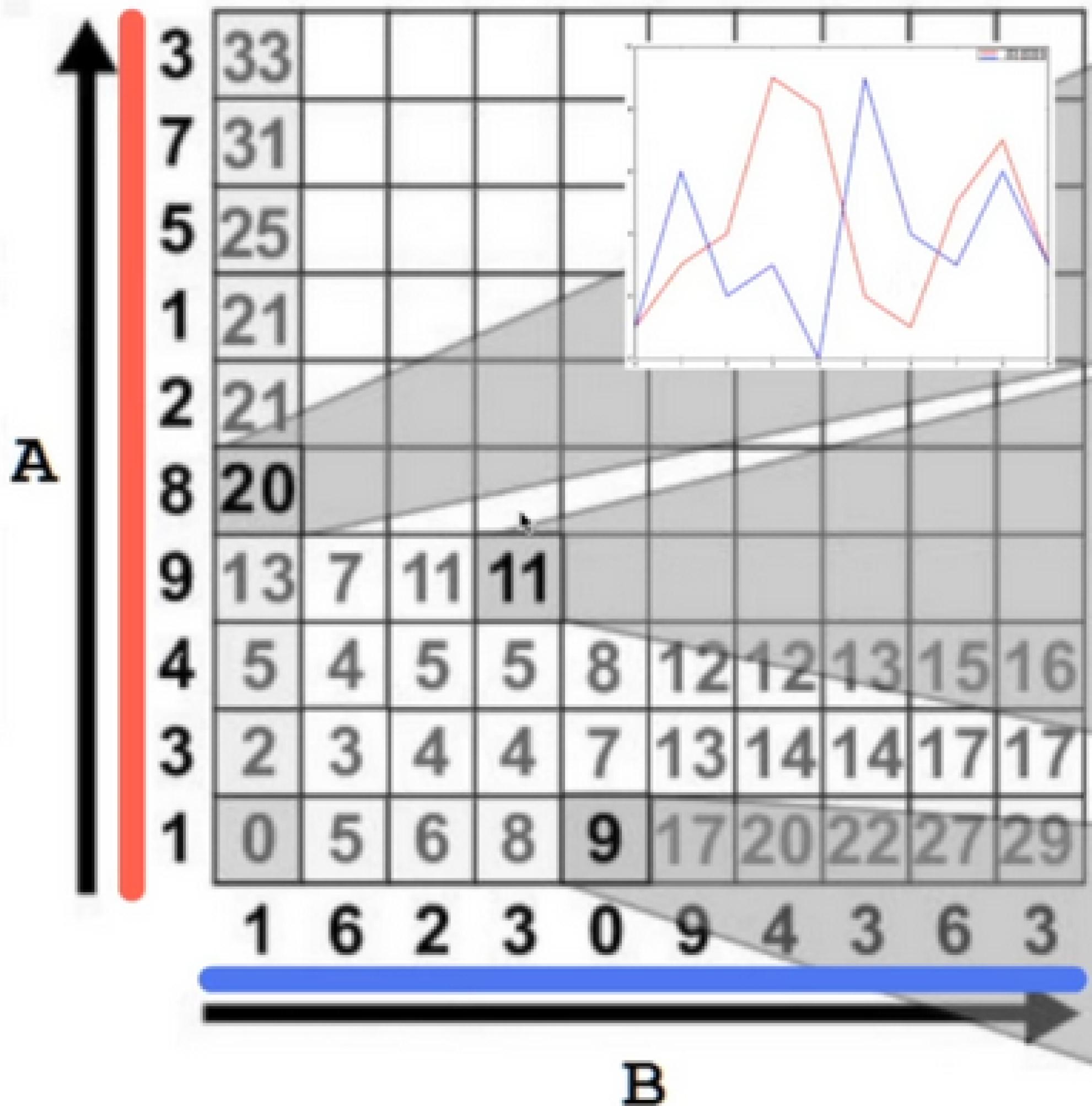
# Algoritm

Acest algoritm generic calculează, folosind programarea dinamică, distanțele optime între perechi de eşantioane

```
int DTWDistance(s: array [1..n], t: array [1..m]) {  
    DTW := array [0..n, 0..m]  
  
    for i := 0 to n  
        for j := 0 to m  
            DTW[i, j] := infinity  
    DTW[0, 0] := 0  
  
    for i := 1 to n  
        for j := 1 to m  
            cost := d(s[i], t[j])  
            DTW[i, j] := cost + minimum(DTW[i-1, j ], // insertion  
                                         DTW[i , j-1], // deletion  
                                         DTW[i-1, j-1]) // match  
  
    return DTW[n, m]  
}
```

Sursă: [Wikipedia](#)

# Sursa: How DTW (Dynamic Time Warping) algorithm works



=  $|A_i - B_j| + D[i-1, 0]$

=  $|8 - 1| + 13$

= 20

=  $|A_i - B_j| + \min(D[i-1, j-1], D[i-1, j], D[i, j-1])$

=  $|9 - 3| + \min(5, 5, 11)$

= 6 + 5

= 11

=  $|A_i - B_j| + D[0, j-1]$

=  $|1 - 0| + 8$

= 9

# Cum se evaluatează costul



Luând 2 serii de timp  $X (x_1, x_2, \dots, x_n)$  de lungime  $N \in \mathbb{N}$  și  $Y (y_1, y_2, \dots, y_m)$  de lungime  $M \in \mathbb{N}$ , costul distanței locale între  $x_i$  și  $y_j$  din spațiul de feature-uri  $F$ ,  $c(x_i, y_j)$ , definită ca  $c : F \times F \rightarrow \mathbb{R} \geq 0$  este mai mic cu cât  $x_i$  și  $y_j$  sunt mai similare și mai mare cu cât  $x_i$  și  $y_j$  sunt mai diferite.

Matricea de cost  $C \in \mathbb{R}^{N \times M}$  definită ca  $C(n,m) := c(x_n, y_m)$ . Scopul este de a găsi alinierea dintre  $X$  și  $Y$  cu costul total cel mai mic.

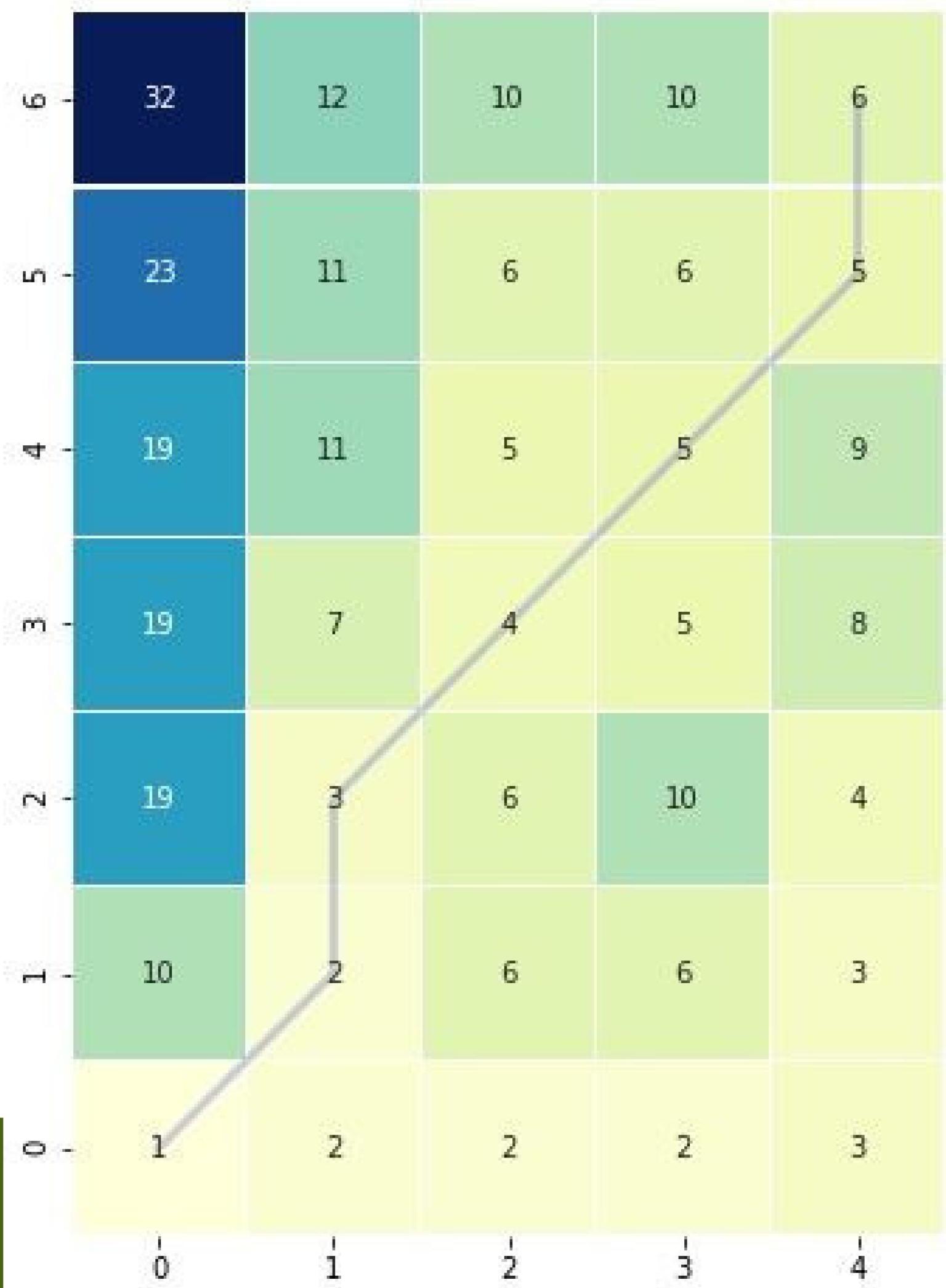
Warping path  $(N,M)$  este definită ca fiind calea în care  $N$  și  $M$  sunt libere de context și este secvența  $p = (p_1, \dots, p_L)$  cu  $p_l = (n_l, m_l) \in [1:N] \times [1:M]$  pentru  $l \in [1:L]$  ce satisfac următoarele 3 condiții:

- 1) Condiția de graniță:  $p_1 = (1,1)$  și  $p_L = (N, M)$
- 2) Condiția de monotonie:  $n_1 \leq n_2 \leq \dots \leq n_L$  și  $m_1 \leq m_2 \leq \dots \leq m_L$
- 3) Condiția de pas:  $p_{l+1} - p_l \in \{(1,0), (0,1), (1,1)\}$  pentru  $l \in [1:L-1]$



# Cum se evaluatează costul

$\min(DTW[i-1][j-i],$   
 $DTW[i][j-1],$   
 $DTW[i-1][j])$

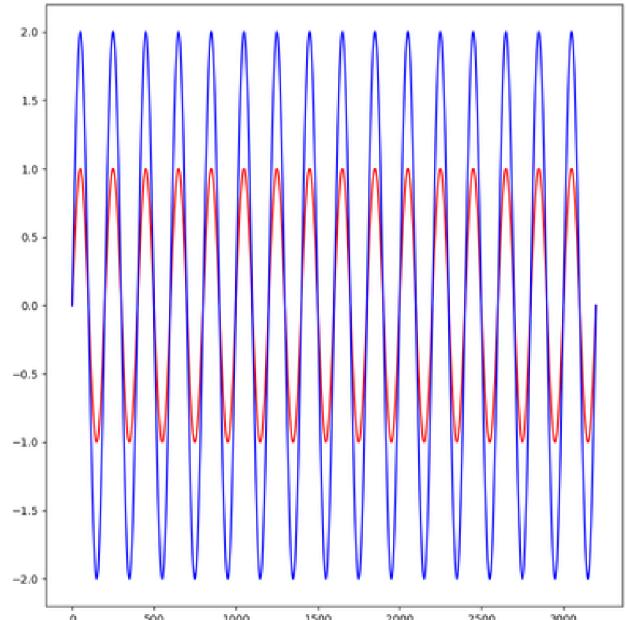




# Cercetare - comparatii între distanțe -

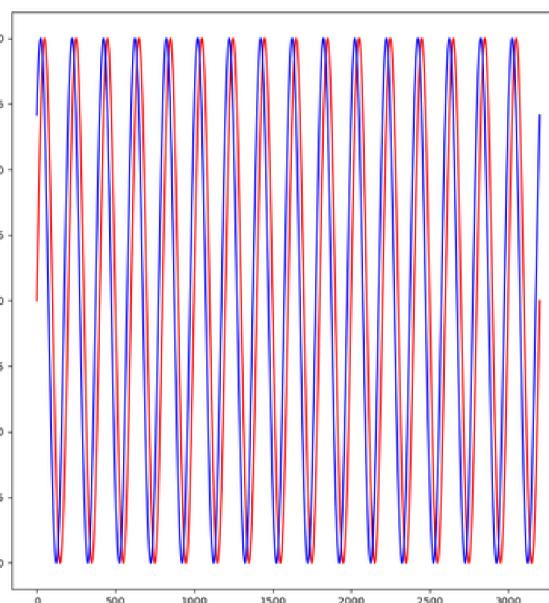
## Distanța Euclidiană vs DTW

```
sinusoidal_a = np.sin(2*np.pi*800*t)  
sinusoidal_b = 2 * np.sin(2*np.pi*800*t)
```



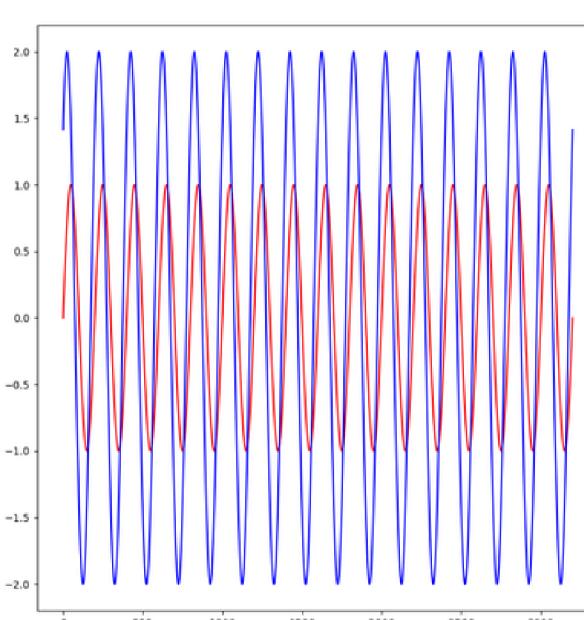
**Euclidian distance: 39.99374951164247**  
**DTW distance: 33.28512015897229**

```
sinusoidal_a = np.sin(2*np.pi*800*t)  
sinusoidal_b = np.sin(2*np.pi*800*t + np.pi/4)
```

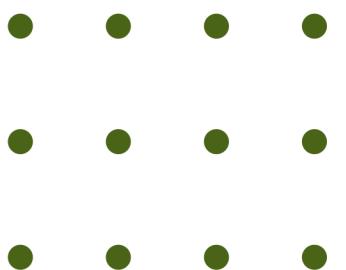


**Euclidian distance: 30.618056877996548**  
**DTW distance: 2.983868872050996**

```
sinusoidal_a = np.sin(2*np.pi*800*t)  
sinusoidal_b = 2 * np.sin(2*np.pi*800*t + np.pi/4)
```



**Euclidian distance: 58.952784615897976**  
**DTW distance: 34.04965823659342**





# Cercetare - algoritmi de aproximare -

## Fast DTW

Pași:

1. Coarsening: micșorarea seriei de timp în serii de timp mai mici prin calcularea mediei între perechile adiacente de puncte
2. Proiecție: Găsirea distanței minime a warping path-ului ca pas inițial pentru ghicirea warping path-ului de rezoluție mai mare
3. Rafinare: Rafinarea warping path-ului de la rezoluție mică la rezoluție mare cu ajustări locale

```
timel = np.linspace(0, 0.02, 3200)
```

```
s1 = np.sin(2*np.pi*800*timel)
```

```
s2 = np.sin(2*np.pi*800*timel + np.pi/4)
```

```
dtw_distance = pyts_dtw(s1, s2)
```

```
distance= pyts_dtw(s1, s2, method='fast', options={'radius': 1})
```

DTW time: 1.4440770149230957

DTW distance: 2.983868872050996

Fast DTW time: 2.087134838104248

Fast DTW distance: 3.029737736737499

```
dtw_distance = pyts_dtw(s1, s2)
```

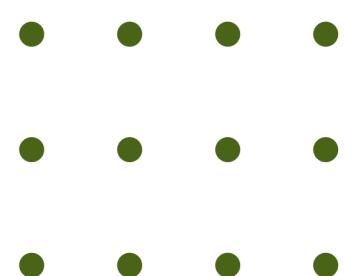
```
distance= pyts_dtw(s1, s2, method='fast', options={'radius': 10})
```

DTW time: 1.7194950580596924

DTW distance: 2.983868872050996

Fast DTW time: 3.0670578479766846

Fast DTW distance: 2.985195797108467





# Cercetare - comparatie intre librarii -

```
t = np.linspace(0, 0.02, 3200)
```

```
sinusoidal_a = np.sin(2*np.pi*800*t)
sinusoidal_b = 2 * np.sin(2*np.pi*800*t)
```

**DTA distance time:** 0.0877840518951416  
**Tslearn time:** 1.1982133388519287  
**Pyts time:** 1.016982078552246  
**DTW python time:** 0.29955387115478516

## Librării folosite::

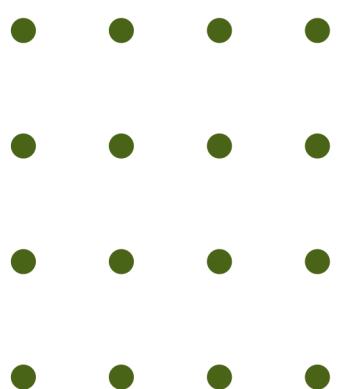
**DTA Distance:** <https://github.com/wannesm/dtaidistance>

**TSLearn:** <https://github.com/tslearn-team/tslearn>

**Pyts:** <https://github.com/johannfaouzi/pyts>

**DTW-python:** <https://github.com/DynamicTimeWarping/dtw-python>

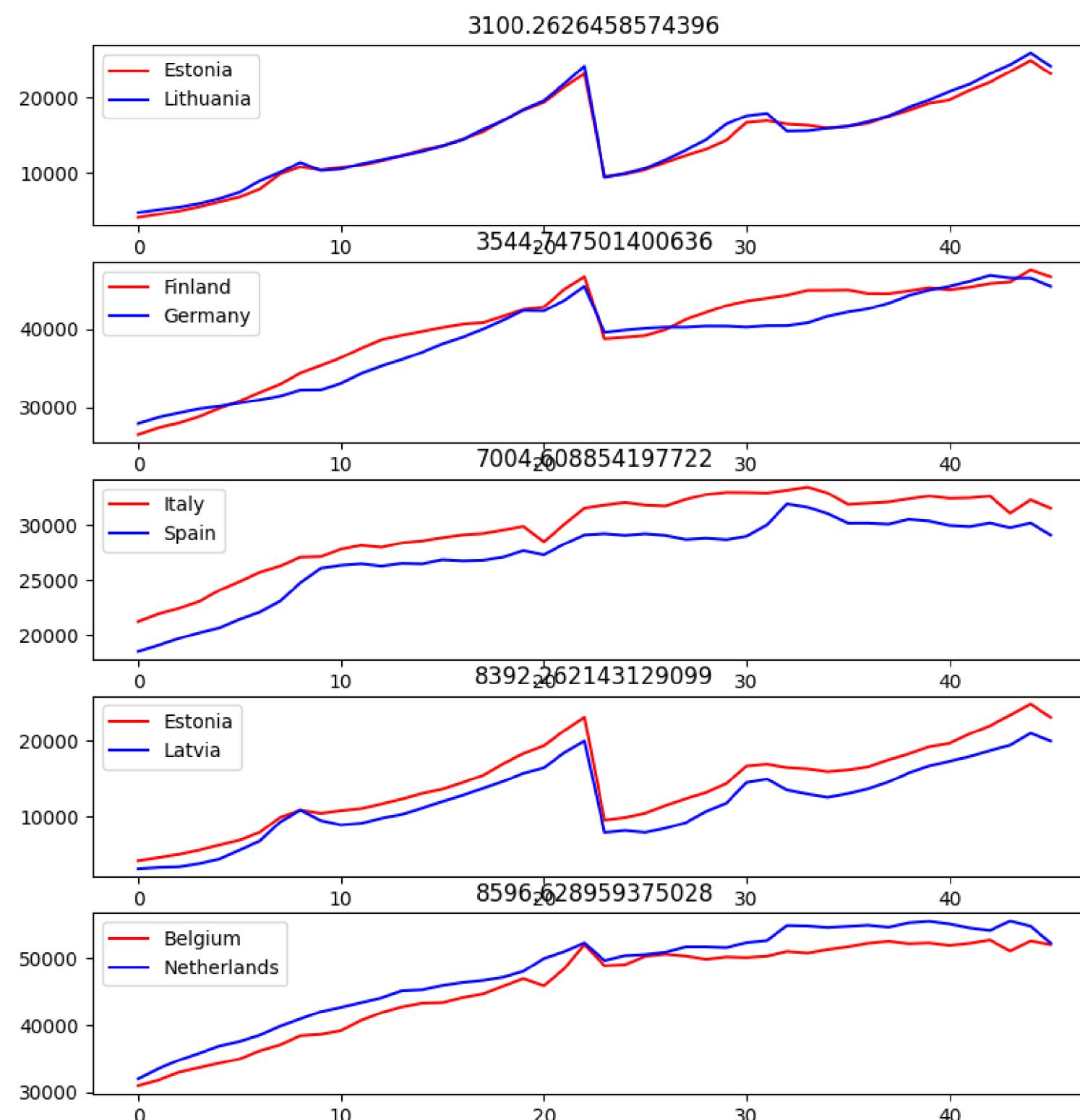
**(Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. J. Stat. Soft., 31 (2009) doi:10.18637/jss.v031.i07.)**



**Sursa:** Mario Filho - 2023 -

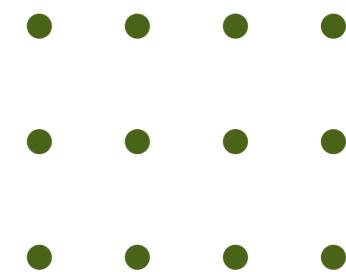
<https://forecastegy.com/posts/dynamic-time-warping-dtw-libraries-python-examples/>

# Cercetare - comparație între salariile medii -

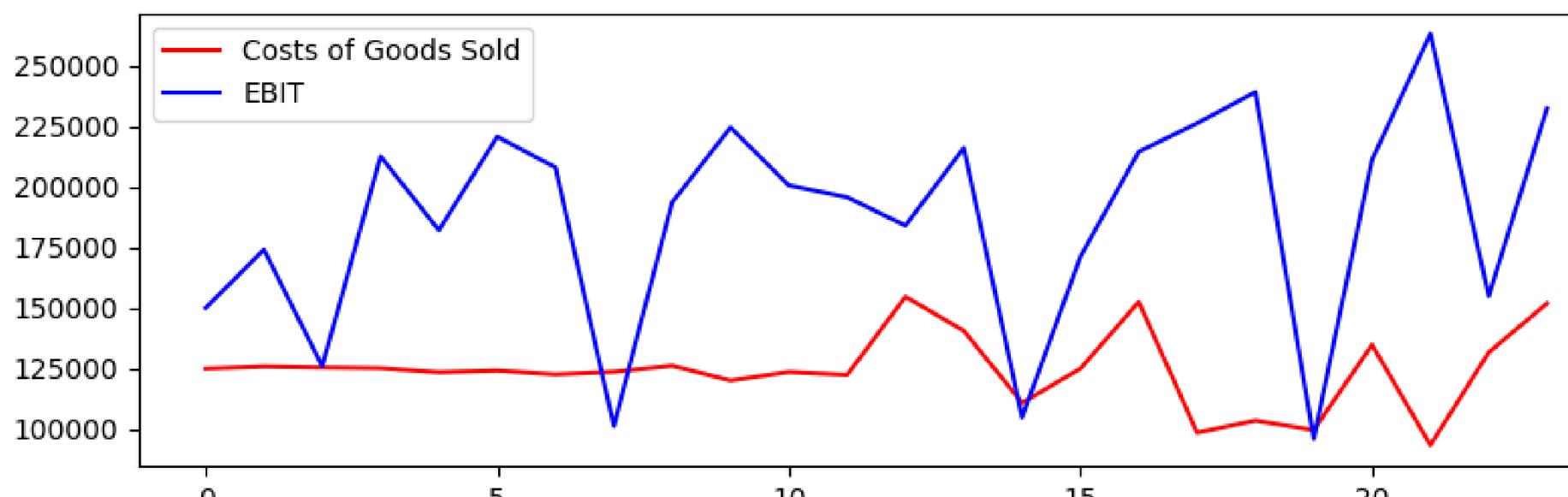
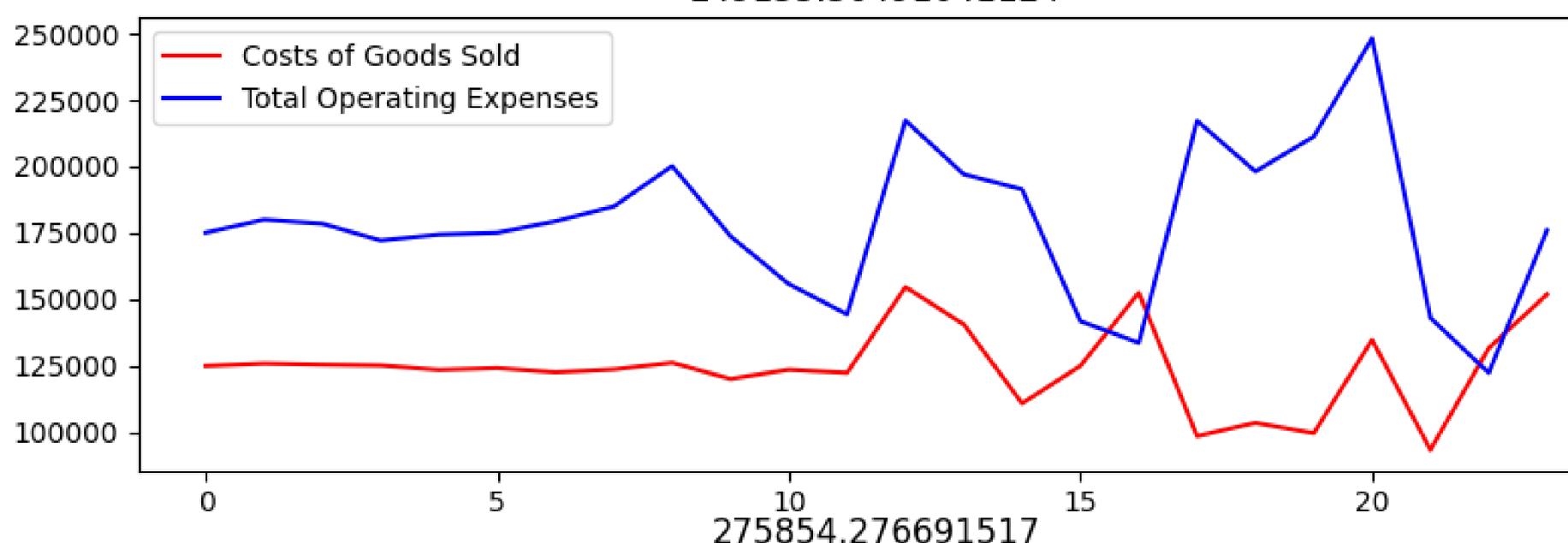
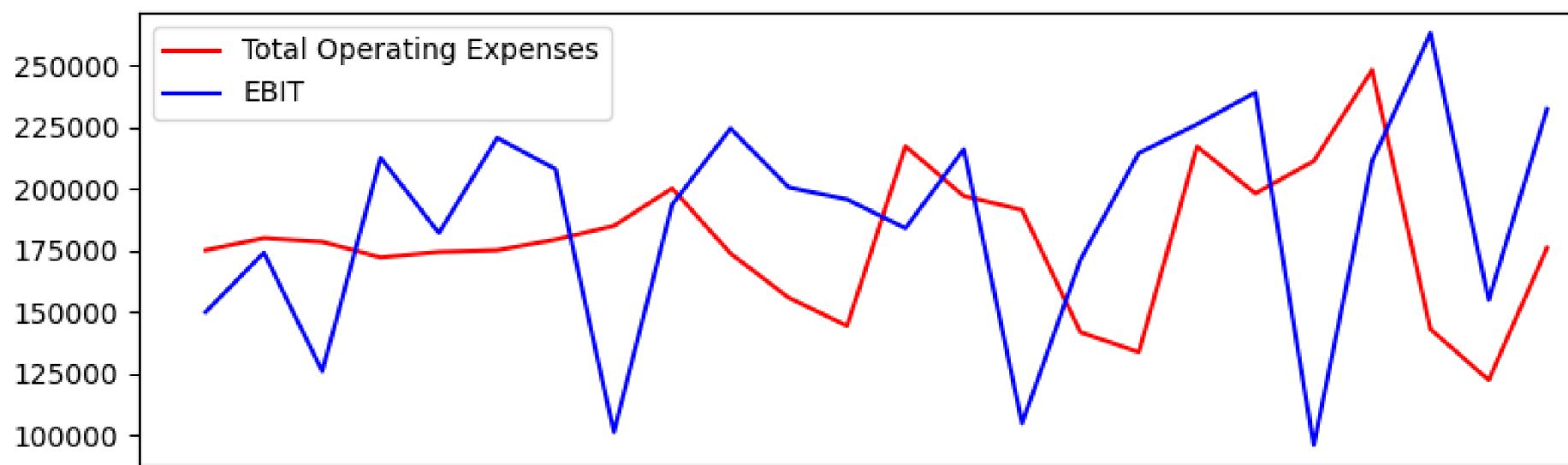


Folosind datele furnizate de OECD privind salariile medii exprimate în EUR în intervalul 2000-2022, am găsit top 5 țări cu evoluții similare ale salariilor

**Sursa:** [https://stats.oecd.org/Index.aspx?  
DatasetCode=AV\\_AN\\_WAGE](https://stats.oecd.org/Index.aspx?DatasetCode=AV_AN_WAGE)

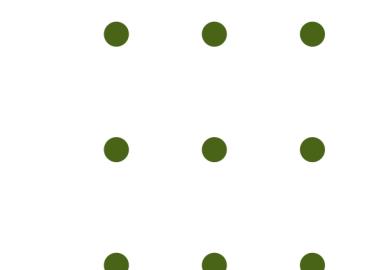


# Cercetare - date din situațiiile financiare -



Sursa: <https://data.world/vizwiz/profit-loss-sample-data/workspace/file?>

**Determinare DTW între sumele  
înregistrate ca Venit, Costul  
Vânzărilor, Profit Brut, Cheltuieli de  
Exploatare și Profit Înainte de  
Dobânzi și Taxe**



# Cercetare - text mining -

## Comparări între folosirea cuvântului

**“Dumnezeu”/”God”/”Gott” în ferestre constând în capitole din Bibliele traduse în română (traducere Cornilescu), engleză (King James Version) și germană (Luther)**

**English: 4111**

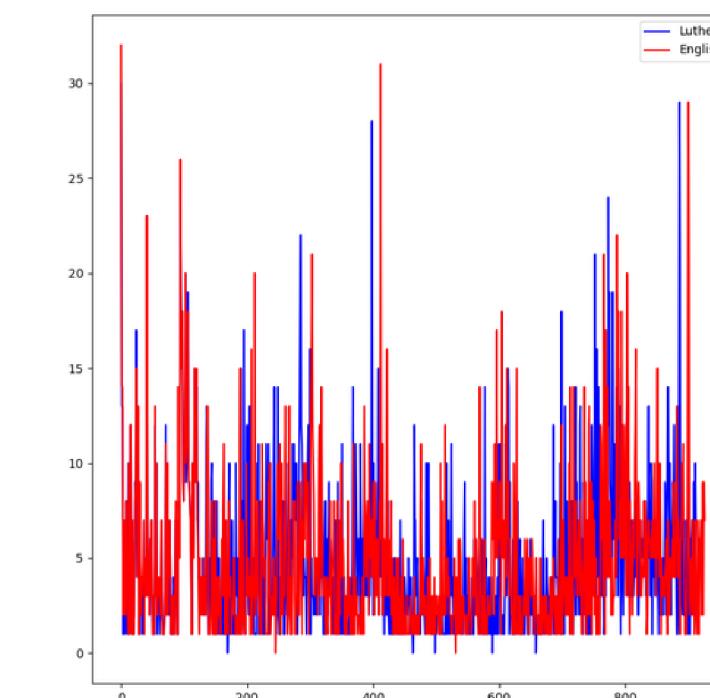
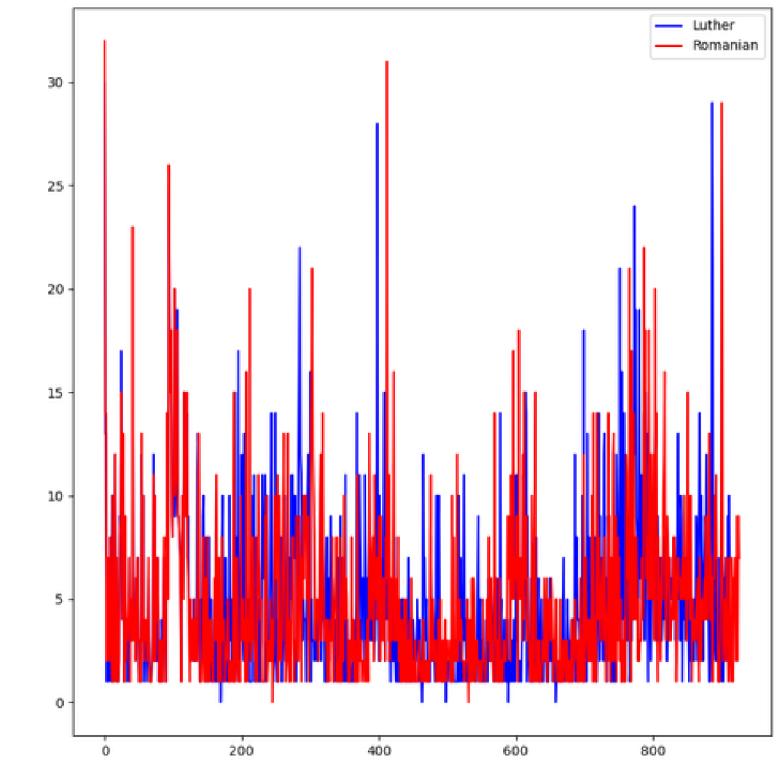
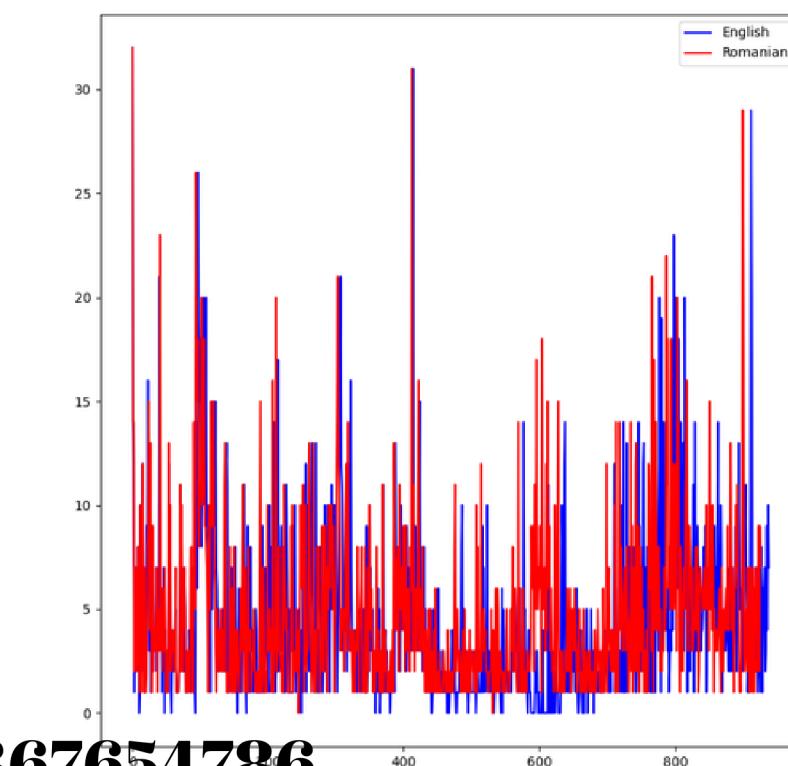
**Romanian: 4468**

**Luther: 4353**

**DTW distance between english and romanian: 35.369478367654786**

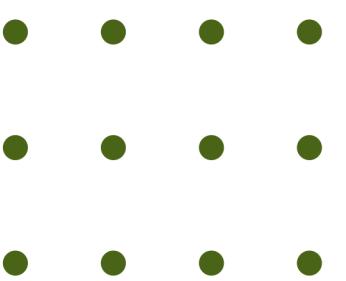
**DTW distance between luther and romanian: 40.34848200366403**

**DTW distance between luther and english: 35.958309192730404**



**Sursa: Chotirat Ann Ratanamahatana, Eamonn Keogh -**  
[https://www.cs.ucr.edu/~eamonn/DTW\\_myths.pdf](https://www.cs.ucr.edu/~eamonn/DTW_myths.pdf)

# Bibliografie



- [1] Using Dynamic Time Warping to Find Patterns in Time Series,  
Donald J. Bemdt, JamesCliffor, 1994, <https://cdn.aaai.org/Workshops/1994/WS-94-03/WS94-03-031.pdf>
- [2] Anomaly Detection Using Dynamic Time Warping, IEEE 2019, <https://ieeexplore.ieee.org/document/8919604>
- [3] A Comparison of k-NN Methods for Time Series Classification and Regression, Vivek Mahato, Martin O'Reilly, and Pádraig Cunningham School of Computer Science University College Dublin Dublin 4, Ireland, [https://ceur-ws.org/Vol-2259/aics\\_11.pdf](https://ceur-ws.org/Vol-2259/aics_11.pdf)
- [4] An Algorithm for the Machine Calculation of Complex Fourier Series, James W. Cooley, John W. Tukey, 1965, <https://www.ams.org/journals/mcom/1965-19-090/S0025-5718-1965-0178586-1/S0025-5718-1965-0178586-1.pdf>
- [5] Vintsyuk, T. K. (1968). "Speech discrimination by dynamic programming". *Kibernetika*. 4: 81–88, <https://link.springer.com/article/10.1007/BF01074755>
- [6] Understanding Dynamic Time Warping, Ricardo Portilla and Brenner Heintz, 2019, <https://www.databricks.com/blog/2019/04/30/understanding-dynamic-time-warping.html>
- [7] Meinard Müller (2007). "Dynamic time warping" [https://www.researchgate.net/publication/285279006\\_Dynamic\\_time\\_warping](https://www.researchgate.net/publication/285279006_Dynamic_time_warping)
- [8] Annagiulia Tiozz (2022). "Dynamic time warping for time series classification" <https://medium.com/eni-digitalks/dynamic-time-warping-for-time-series-classification-6482cf9d7c3b>
- [9] Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *J. Stat. Soft.*, 31 (2009) doi:[10.18637/jss.v031.i07](https://doi.org/10.18637/jss.v031.i07).
- [10] Mario Filho (2023). "5 Dynamic Time Warping (DTW) Libraries in Python With Examples" <https://forecastegy.com/posts/dynamic-time-warping-dtw-libraries-python-examples/>
- [11] [https://stats.oecd.org/Index.aspx?DatasetCode=AV\\_AN\\_WAGE](https://stats.oecd.org/Index.aspx?DatasetCode=AV_AN_WAGE)
- [12] <https://data.world/vizwiz/profit-loss-sample-data/workspace/file?filename=Finance+Dashboard+Template.xlsx>
- [13] Cornilescu, Dumitru, translator. Biblia: Versiunea Dumitru Cornilescu. Bucharest: Institutul Biblic și de Misiune al Bisericii Ortodoxe Române, 1921. <https://www.biblesupersearch.com/bible-downloads/>
- [14] The Holy Bible: King James Version, 1604 <https://www.biblesupersearch.com/bible-downloads/>
- [15] Luther, Martin. Die Bibel, oder die ganze Heilige Schrift des Alten und Neuen Testaments. Edited by [Editor's Name]. Wittenberg: Hans Lufft, 1545. <https://www.biblesupersearch.com/bible-downloads/>
- [16] Chotirat Ann Ratanamahatana, Eamonn Keogh. "Everything you know about Dynamic Time Warping is Wrong" [https://www.cs.ucr.edu/~eamonn/DTW\\_myths.pdf](https://www.cs.ucr.edu/~eamonn/DTW_myths.pdf)



vă  
mulțumim

