

Μη-Επιβλεπόμενη Μάθηση

Fundamentals of Machine Learning for Predictive Data
Analytics

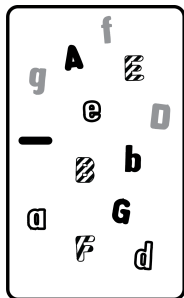
© John D. Kelleher and Brian Mac Namee and Aoife D'Arcy
Machine Learning in Business: An Introduction to the World
of Data Science.

© John Hull

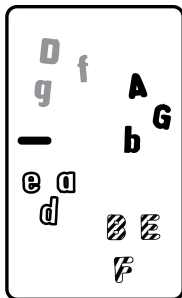
Αθανάσιος Σάκκας, ΟΠΑ

- 1 Βασική Ιδέα
- 2 Μάθηση βασισμένη στην ομοιότητα - Εισαγωγή
- 3 Τυπική Προσέγγιση: Ο Αλγόριθμος Ομαδοποίησης k -Means
- 4 Agglomerative Hierarchical Clustering
- 5 Principal Component Analysis (PCA)
- 6 Σύνοψη
- 7 Εμπειρική Εφαρμογή

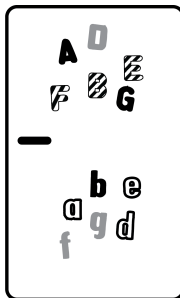
Βασική Ιδέα



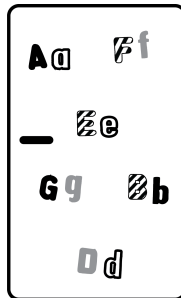
(α) Το ψυγείο



(β) Abigail

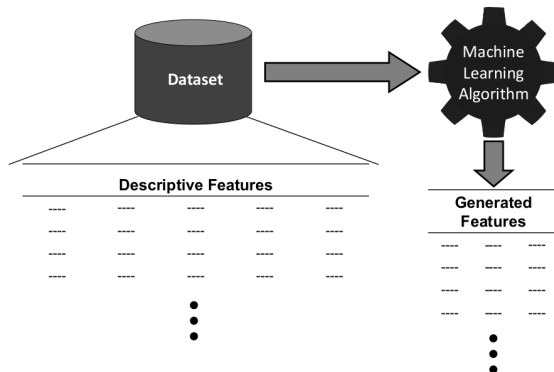


(γ) Andrew



(δ) Amalia

Σχήμα 1: Οι τρεις διαφορετικές διατάξεις των μαγνητικών γραμμάτων που έφτιαξαν τα παιδιά της οικογένειας Murphy στο ψυγείο τους.



Σχήμα 2: Η μη-επιβλεπόμενη μηχανική μάθηση ως διαδικασία ενός βήματος.

Μάθηση βασισμένη στην ομοιότητα - Εισαγωγή

- Το έτος είναι 1798 και είστε ο Αντισυνταγματάρχης David Collins του HMS Calcutta, ο οποίος εξερευνά την περιοχή γύρω από τον ποταμό Hawkesbury, στη Νέα Νότια Ουαλία.
- Μετά από μία αποστολή κατά μήκος του ποταμού, ένας από τους άνδρες σας λέει ότι είδε ένα παράξενο ζώο κοντά στο ποτάμι.
- Του ζητάτε να σας περιγράψει το ζώο και σας εξηγεί ότι δεν το είδε πολύ καλά, αλλά παρατήρησε ότι είχε πατούσες με μεμβράνες (σαν πάπιας), ράμφος πάπιας, και ότι του γρύλισε.
- Για να σχεδιάσετε την επόμενη ημέρα της αποστολής, αποφασίζετε ότι πρέπει να ταξινομήσετε το ζώο, ώστε να καταλάβετε αν είναι επικίνδυνο να το πλησιάσετε ή όχι.

	Γκρρρ!			Βαθμολογία
	✓	✗	✗	1
	✗	✓	✗	1
	✗	✓	✓	2

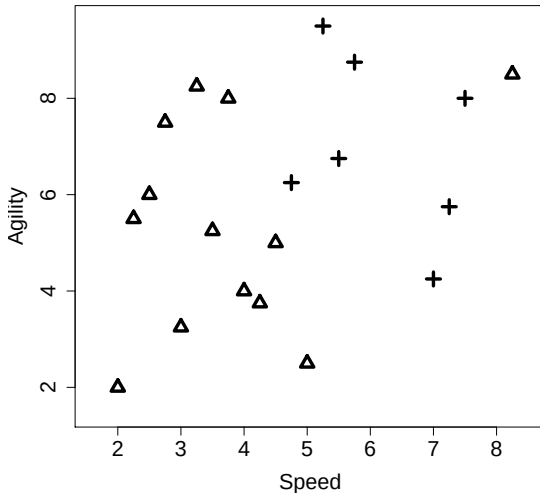
- Η διαδικασία ταξινόμησης ενός άγνωστου ζώου, αντιστοιχίζοντας τα χαρακτηριστικά του με τα χαρακτηριστικά ζώων που θυμάστε, αποτυπώνει με πολύ καθαρό τρόπο τη «μεγάλη ιδέα» που βρίσκεται πίσω από τη μάθηση βασισμένη στην ομοιότητα:

αν προσπαθείτε να ταξινομήσετε κάτι, τότε θα πρέπει να «ψάξετε στη μνήμη» σας για πράγματα που είναι παρόμοια και να του αποδώσετε την ίδια κλάση με το πιο παρόμοιο πράγμα που θυμάστε

- Οι βασικές αρχές της μάθησης βασισμένης στην ομοιότητα είναι:
 - Χώρος χαρακτηριστικών
 - Μετρικές ομοιότητας

Πίνακας 1: Οι αξιολογήσεις ταχύτητας και ευκινησίας για 20 κολεγιακούς αθλητές, με ετικέτα την απόφαση για το αν επιλέχθηκαν (drafted) ή όχι.

ID	Speed	Agility	Draft	ID	Speed	Agility	Draft
1	2.50	6.00	No	11	2.00	2.00	No
2	3.75	8.00	No	12	5.00	2.50	No
3	2.25	5.50	No	13	8.25	8.50	No
4	3.25	8.25	No	14	5.75	8.75	Yes
5	2.75	7.50	No	15	4.75	6.25	Yes
6	4.50	5.00	No	16	5.50	6.75	Yes
7	3.50	5.25	No	17	5.25	9.50	Yes
8	3.00	3.25	No	18	7.00	4.25	Yes
9	4.00	4.00	No	19	7.50	8.00	Yes
10	4.25	3.75	No	20	7.25	5.75	Yes



Σχήμα 3: Διάγραμμα χώρου χαρακτηριστικών για τα δεδομένα του Πίνακα 1. Τα τρίγωνα αντιπροσωπεύουν τις *Μη-επιλογής* (Non-draft) και οι σταυροί αντιπροσωπεύουν τις *Επιλογής* (Draft).

- Ένας **χώρος χαρακτηριστικών** είναι ένας αφηρημένος n -διάστατος χώρος που δημιουργείται θεωρώντας κάθε περιγραφικό χαρακτηριστικό ενός ABT ως άξονα ενός συστήματος αναφοράς. Κάθε παρατήρηση (instance) του συνόλου δεδομένων απεικονίζεται ως ένα σημείο στον χώρο χαρακτηριστικών, με βάση τις τιμές των περιγραφικών της χαρακτηριστικών.

- Μια **μετρική ομοιότητας** μετρά την ομοιότητα μεταξύ δύο παρατηρήσεων, σύμφωνα με έναν χώρο χαρακτηριστικών.
- Μαθηματικά, μια **μετρική** πρέπει να ικανοποιεί τα ακόλουθα τέσσερα κριτήρια:

① **Μη-αρνητικότητα:** $metric(\mathbf{a}, \mathbf{b}) \geq 0$

② **Ταυτότητα:** $metric(\mathbf{a}, \mathbf{b}) = 0 \iff \mathbf{a} = \mathbf{b}$

③ **Συμμετρία:** $metric(\mathbf{a}, \mathbf{b}) = metric(\mathbf{b}, \mathbf{a})$

④ **Τριγωνική ανισότητα:**

$$metric(\mathbf{a}, \mathbf{b}) \leq metric(\mathbf{a}, \mathbf{c}) + metric(\mathbf{b}, \mathbf{c})$$

όπου $metric(\mathbf{a}, \mathbf{b})$ είναι μια συνάρτηση που επιστρέφει την απόσταση μεταξύ δύο παρατηρήσεων $\mathbf{a}$ και $\mathbf{b}$.

- Μία από τις πιο γνωστές μετρικές είναι η **Ευκλείδεια απόσταση**, η οποία υπολογίζει το μήκος της ευθείας γραμμής μεταξύ δύο σημείων. Η ευκλείδεια απόσταση μεταξύ δύο παρατηρήσεων **a** και **b** σε έναν m -διάστατο χώρο χαρακτηριστικών ορίζεται ως:

$$Euclidean(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^m (\mathbf{a}[i] - \mathbf{b}[i])^2} \quad (1)$$

Example

Η ευκλείδεια απόσταση μεταξύ των παρατηρήσεων d_{12} (Speed= 5.00, Agility= 2.5) και d_5 (Speed= 2.75, Agility= 7.5) στον Πίνακα 1 είναι:

Example

Η ευκλείδεια απόσταση μεταξύ των παρατηρήσεων d_{12} (Speed= 5.00, Agility= 2.5) και d_5 (Speed= 2.75, Agility= 7.5) στον Πίνακα 1 είναι:

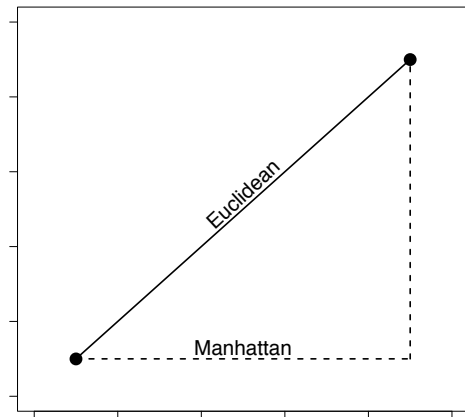
$$\begin{aligned} Euclidean(\langle 5.00, 2.50 \rangle, \langle 2.75, 7.50 \rangle) &= \sqrt{(5.00 - 2.75)^2 + (2.50 - 7.50)^2} \\ &= \sqrt{30.0625} = 5.4829 \end{aligned}$$

- Μία άλλη, λιγότερο γνωστή, μέτρηση απόστασης είναι η απόσταση **Manhattan** ή **απόσταση ταξί (taxi-cab distance)**.
- Η απόσταση Manhattan μεταξύ δύο παρατηρήσεων **a** και **b** σε έναν χώρο χαρακτηριστικών με m διαστάσεις είναι:¹

$$Manhattan(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^m abs(\mathbf{a}[i] - \mathbf{b}[i]) \quad (2)$$

¹ Η συνάρτηση $abs()$ που περιβάλλει τον όρο αφαίρεσης δείχνει ότι χρησιμοποιούμε την απόλυτη τιμή, δηλαδή μη-αρνητική τιμή, όταν αθροίζουμε τις διαφορές· αυτό έχει νόημα επειδή οι αποστάσεις δεν μπορεί να είναι αρνητικές.

Μετρικές Απόστασης



Σχήμα 4: Οι αποστάσεις Manhattan και Ευκλείδεια μεταξύ δύο σημείων.

Example

Η απόσταση Manhattan μεταξύ των παρατηρήσεων d_{12} (Speed= 5.00, Agility= 2.5) και d_5 (Speed= 2.75, Agility= 7.5) στον Πίνακα 1^[11] είναι:

Example

Η απόσταση Manhattan μεταξύ των παρατηρήσεων d_{12} (Speed= 5.00, Agility= 2.5) και d_5 (Speed= 2.75, Agility= 7.5) στον Πίνακα 1^[11] είναι:

$$\begin{aligned} Manhattan(\langle 5.00, 2.50 \rangle, \langle 2.75, 7.50 \rangle) &= abs(5.00 - 2.75) + abs(2.5 - 7.5) \\ &= 2.25 + 5 = 7.25 \end{aligned}$$

- Οι ευκλείδειες και Manhattan αποστάσεις είναι ειδικές περιπτώσεις της **απόστασης Minkowski**.
- Η **απόσταση Minkowski** μεταξύ δύο παρατηρήσεων **a** και **b** σε έναν χώρο χαρακτηριστικών με m περιγραφικά χαρακτηριστικά είναι:

$$Minkowski(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^m abs(\mathbf{a}[i] - \mathbf{b}[i])^p \right)^{\frac{1}{p}} \quad (3)$$

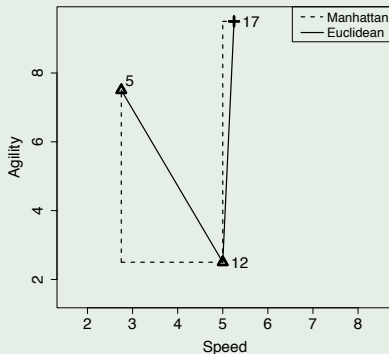
όπου διαφορετικές τιμές της παραμέτρου p οδηγούν σε διαφορετικές μετρικές απόστασης.

- Η απόσταση Minkowski με $p = 1$ είναι η απόσταση Manhattan και με $p = 2$ είναι η ευκλείδεια απόσταση.

- Όσο μεγαλύτερη είναι η τιμή του p , τόσο μεγαλύτερη έμφαση δίνεται στα χαρακτηριστικά με μεγάλες διαφορές τιμών, επειδή αυτές οι διαφορές υψώνονται στη δύναμη p .

Example

ID Παρατήρησης	ID Παρατήρησης	Manhattan (Minkowski $p=1$)	Euclidean (Minkowski $p=2$)
12	5	7.25	5.4829
12	17	7.25	8.25



Οι αποστάσεις Manhattan και Ευκλείδεια μεταξύ των παρατηρήσεων d_{12} (Speed= 5.00, Agility= 2.5) και d_5 (Speed= 2.75, Agility= 7.5), καθώς και μεταξύ των παρατηρήσεων d_{12} και d_{17} (Speed= 5.25, Agility= 9.5).

Τυπική Προσέγγιση: Ο Αλγόριθμος Ομαδοποίησης k -Means

Τα βήματα που απαιτούνται για την ομαδοποίηση (1)

A. Scaling

Πριν χρησιμοποιήσετε αλγόριθμους ML , είναι σημαντικό να κάνετε scaling τις τιμές των χαρακτηριστικών έτσι ώστε να είναι συγκρίσιμες.

Δύο τρόποι:

- α. **Z-score scaling:**

$$x' = \frac{x - \text{Mean}}{\text{SD}}$$

- β. **Min-Max scaling:**

$$x' = \frac{x - \text{Minimum}}{\text{Maximum} - \text{Minimum}}$$

Τα βήματα που απαιτούνται για την ομαδοποίηση (2)

B. Μέτρο απόσταση

- Το απλούστερο μέτρο απόστασης είναι το μέτρο της Ευκλείδειας απόστασης.
- Απόσταση (Distance):

$$Dist(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

- Γενικά, όταν υπάρχουν m χαρακτηριστικά, η απόσταση μεταξύ P και Q είναι:

$$Dist(P, Q) = \sqrt{\sum_{j=1}^m (v_{pj} - v_{qj})^2}$$

όπου v_{pj} και v_{qj} είναι οι τιμές του j -οστού χαρακτηριστικού των παρατηρήσεων P και Q .

Τα βήματα που απαιτούνται για την ομαδοποίηση (3)

Γ. Κέντρα Ομάδων (Cluster Centers)

- Το κέντρο μιας ομάδας (μερικές φορές ονομάζεται κεντροειδής — centroid) προσδιορίζεται ως ο μέσος όρος των τιμών κάθε χαρακτηριστικού για όλα τα σημεία της ομάδας.
- Παράδειγμα:

Observ.	Feature 1	Feature 2	Feature 3	Feature 4	Distance to center
1	1.00	1.00	0.40	0.25	0.145
2	0.80	1.20	0.25	0.40	0.258
3	0.82	1.05	0.35	0.50	0.206
4	1.10	0.80	0.21	0.23	0.303
5	0.85	0.90	0.37	0.27	0.137
Center	0.914	0.990	0.316	0.330	

Υπολογισμός Απόστασης από το Κέντρο

Θυμίζουμε ότι το κέντρο (centroid) είναι:

$$\mathbf{c} = (0.914, 0.990, 0.316, 0.330)$$

Για την Παρατήρηση 1 έχουμε:

$$\mathbf{d}_1 = (1.00, 1.00, 0.40, 0.25)$$

Η Ευκλείδεια απόσταση είναι:

$$Dist(\mathbf{d}_1, \mathbf{c}) = \sqrt{(1.00 - 0.914)^2 + (1.00 - 0.990)^2 + (0.40 - 0.316)^2 + (0.25 - 0.330)^2}$$

$$= \sqrt{0.0074 + 0.0001 + 0.0071 + 0.0064} = \sqrt{0.0210} = 0.145$$

Αντικειμενική Συνάρτηση του k -means

$$\sum_{i=1}^n \min_{\mathbf{c}_1, \dots, \mathbf{c}_k} Dist(\mathbf{d}_i, \mathbf{c}_j) \quad (4)$$

- $\mathbf{d}_i$: η i -οστή παρατήρηση (διάνυσμα χαρακτηριστικών) του συνόλου δεδομένων
- $\mathbf{c}_j$: το κεντροειδές (centroid) της j -οστής ομάδας
- n : το πλήθος των παρατηρήσεων
- k : ο αριθμός των ομάδων
- $Dist(\mathbf{d}_i, \mathbf{c}_j)$: η απόσταση της παρατήρησης από το κεντροειδές (συνήθως Ευκλείδεια)

Ψευδοκώδικας του αλγορίθμου **k-means** ομαδοποίηση.

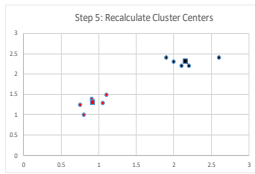
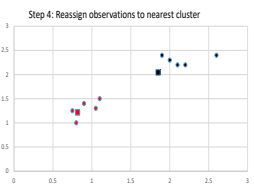
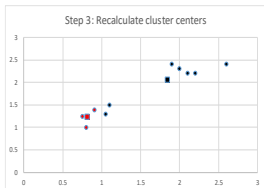
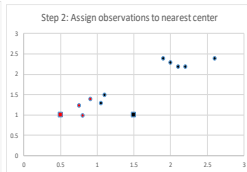
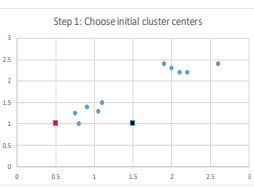
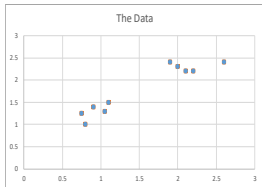
Require: ένα σύνολο δεδομένων $\mathcal{D}$ με n παρατηρήσεις εκπαίδευσης,
 $\mathbf{d}_1, \dots, \mathbf{d}_n$

Require: k , τον αριθμό ομάδων που θέλουμε να βρούμε

Require: ένα μέτρο απόστασης, $Dist$, για να συγκρίνουμε παρατηρήσεις με κεντροειδή

- 1: Επιλέγουμε τυχαία k κεντροειδή, $\mathbf{c}_1$ έως $\mathbf{c}_k$, όπου κάθε κεντροειδής ορίζεται από τιμές για κάθε περιγραφικό χαρακτηριστικό,
 $\mathbf{c}_i = \langle \mathbf{c}_i[1], \dots, \mathbf{c}_i[m] \rangle$
- 2: **repeat**
- 3: Υπολογίζουμε την απόσταση κάθε παρατήρησης $\mathbf{d}_i$ από κάθε κεντροειδές $\mathbf{c}_1$ έως $\mathbf{c}_k$, με χρήση του $Dist$
- 4: Αναθέτουμε κάθε παρατήρηση $\mathbf{d}_i$ στην ομάδα $\mathcal{C}_j$ της οποίας το κεντροειδές $\mathbf{c}_j$ είναι το πιο κοντινό
- 5: Ενημερώνουμε κάθε κεντροειδές $\mathbf{c}_j$ ώστε να είναι ο μέσος όρος των τιμών των περιγραφικών χαρακτηριστικών των παρατηρήσεων που ανήκουν στην ομάδα $\mathcal{C}_j$
- 6: **until** δεν γίνεται καμία νέα ανάθεση ομάδας σε μια επανάληψη

Ένα απλό παράδειγμα



Inertia (Αδράνεια)

- Για κάθε δεδομένο k , ο στόχος είναι να ελαχιστοποιηθεί η αδράνεια, η οποία ορίζεται ως το άθροισμα των τετραγώνων εντός της ομάδας (cluster):

$$\text{Inertia} = \sum_{i=1}^n d_i^2$$

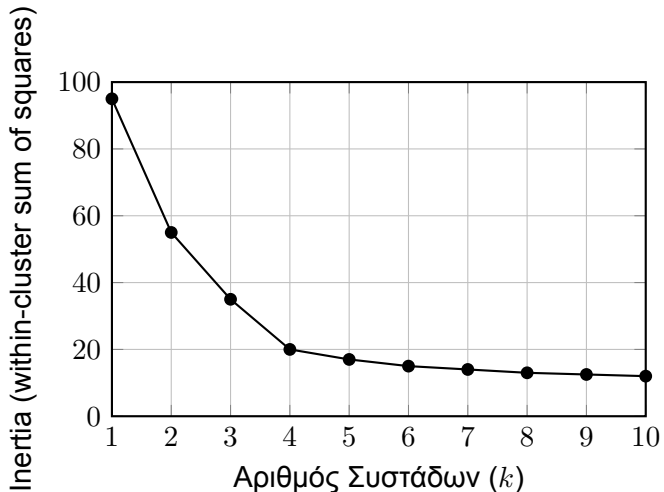
όπου d_i είναι η απόσταση της παρατήρησης i από το κέντρο (cluster center) της ομάδας στην οποία ανήκει.

- Στην πράξη χρησιμοποιούμε τον αλγόριθμο k -means με πολλαπλά διαφορετικά σημεία εκκίνησης και επιλέγουμε το αποτέλεσμα που έχει τη μικρότερη αδράνεια.

Επιλογή του Αριθμού Ομάδων

Επιλέγοντας το k (1)

- Η προσέγγιση elbow:



Επιλέγοντας το k (2)

- Η μέθοδος **silhouette**
- Για κάθε παρατήρηση i υπολογίζουμε:
 - $a(i)$: τη μέση απόσταση από τις άλλες παρατηρήσεις στο ίδιο cluster
 - $b(i)$: τη μέση απόσταση από τις παρατηρήσεις στο πλησιέστερο άλλο cluster
- Το silhouette score για την παρατήρηση i , $s(i)$, ορίζεται ως:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- Επιλέγουμε τον αριθμό των clusters που μεγιστοποιεί το μέσο silhouette score σε όλες τις παρατηρήσεις.

Ψευδοκώδικας για τον υπολογισμό του **silhouette**.

Require: σύνολο δεδομένων $\mathcal{D}$ με n παρατηρήσεις, $\mathbf{d}_1, \dots, \mathbf{d}_n$

Require: μια ομαδοποίηση $\mathcal{C}$ του $\mathcal{D}$ σε k ομάδες, $\mathcal{C}_1, \dots, \mathcal{C}_k$

Require: μέτρο απόστασης $Dist$

- 1: **for** κάθε παρατήρηση $\mathbf{d}_i$ στο $\mathcal{D}$ **do**
- 2: $a(i)$: μέση απόσταση της $\mathbf{d}_i$ από όλες τις άλλες παρατηρήσεις της ίδιας ομάδας (*μέση ενδο-ομαδική απόσταση*)
- 3: υπολογίζουμε τη μέση απόσταση της $\mathbf{d}_i$ από κάθε μία από τις άλλες ομάδες $\mathcal{C} \setminus \mathcal{C}_j$
- 4: $b(i)$: η μικρότερη από αυτές τις μέσες αποστάσεις (*μέση δια-ομαδική απόσταση*)
- 5: υπολογίζουμε:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (5)$$

- 6: **end for**

7: τελικό silhouette: $s = \frac{1}{n} \sum_{i=1}^n s(i)$

Agglomerative Hierarchical Clustering

Περιορισμοί του *k*-means

Ο ***k*-means** (και παρόμοιοι αλγόριθμοι) επιβάλλει μια **καθολική γεωμετρική δομή** στη συσταδοποίηση.

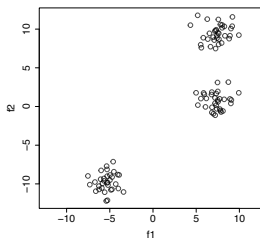
Αναζητά κυρίως:

- **σφιχτές και περίπου σφαιρικές** συστάδες.

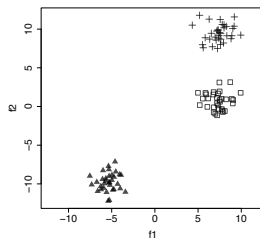
Στην πράξη, τα δεδομένα συχνά έχουν:

- **μη σφαιρικές ή μη κυρτές** δομές,
- **μη γραμμικά** μοτίβα.

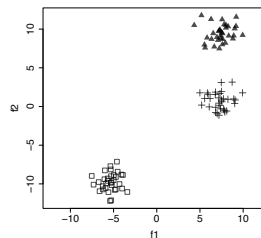
Παραδείγματα 2D: *blobs*, *κύκλοι*, *two moons*. Ο *k*-means αποδίδει σωστά κυρίως στα *blobs*.



(α) Blobs Dataset

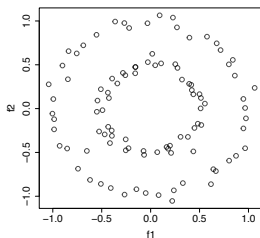


(β) k -means

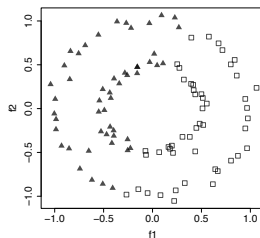


(γ) AHC

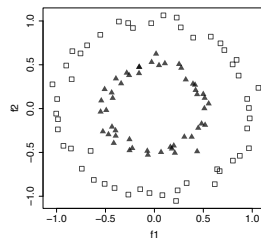
Σχήμα 5: (a)--(i) A plot of the *blobs*, *circles*, and *half-moons* datasets and the clusterings achieved by the k -means clustering and agglomerative hierarchical clustering algorithms (where k is set to 3, 2, and 2, respectively).



(δ) Circles Dataset

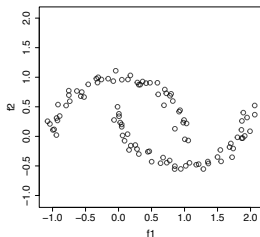


(ε) k -means

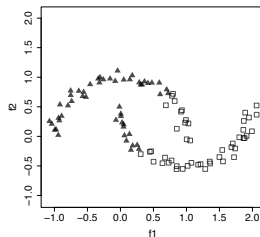


(ζ) AHC

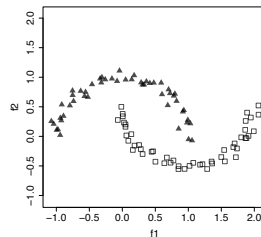
Σχήμα 6: (a)--(i) A plot of the *blobs*, *circles*, and *half-moons* datasets and the clusterings achieved by the k -means clustering and agglomerative hierarchical clustering algorithms (where k is set to 3, 2, and 2, respectively).



(ζ□) Half-moons Dataset



(η□) k -means



(θ□) AHC

Σχήμα 7: (a)--(i) A plot of the *blobs*, *circles*, and *half-moons* datasets and the clusterings achieved by the k -means clustering and agglomerative hierarchical clustering algorithms (where k is set to 3, 2, and 2, respectively).

Συσσωρευτική Ιεραρχική Συσταδοποίηση (AHC)

- Αρχικοποίηση: κάθε παρατήρηση είναι **μία συστάδα** $\Rightarrow n$ αρχικές συστάδες $\mathcal{C}_1, \dots, \mathcal{C}_n$.
- Σε κάθε βήμα:
 - υπολογίζουμε πόσο «κοντά» είναι οι συστάδες (με **απόσταση** $Dist$ και **linkage** $\mathcal{L}$),
 - βρίσκουμε το πλησιέστερο ζεύγος $\mathcal{C}_i, \mathcal{C}_j$,
 - τις **συγχωνεύουμε** σε νέα συστάδα.
- Επαναλαμβάνουμε μέχρι να μείνει **μία** συστάδα που περιέχει όλα τα σημεία.
- Το αποτέλεσμα αποτυπώνεται σε **δενδρόγραμμα** και επιλέγουμε k με **κόψιμο** σε ύψος.

Require: Σύνολο δεδομένων $\mathcal{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_n\}$

Require: Μέτρο απόστασης $Dist(\cdot, \cdot)$ για στιγμιότυπα

Require: Μέθοδος σύνδεσης (linkage) $\mathcal{L}$ για αποστάσεις μεταξύ συστάδων

- 1: Αρχικοποίησε επίπεδο ιεραρχίας: $h \leftarrow 1$
- 2: Αρχικοποίησε συσταδοποίηση: $\mathcal{C} \leftarrow \{\mathcal{C}_1, \dots, \mathcal{C}_n\}$, όπου $\mathcal{C}_i = \{\mathbf{d}_i\}$
- 3: **repeat**
- 4: Βρες το πλησιέστερο ζεύγος συστάδων $(\mathcal{C}_i, \mathcal{C}_j)$
 χρησιμοποιώντας $Dist$ και $\mathcal{L}$
- 5: Συγχώνευσε: $\mathcal{C}_{n+h} \leftarrow \mathcal{C}_i \cup \mathcal{C}_j$
- 6: Αφαίρεσε: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\mathcal{C}_i, \mathcal{C}_j\}$
- 7: Πρόσθεσε: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{C}_{n+h}\}$
- 8: $h \leftarrow h + 1$
- 9: **until** $|\mathcal{C}| = 1$

AHC: Οι δύο βασικές αποφάσεις

Ο αλγόριθμος AHC απαιτεί δύο βασικές επιλογές:

1. Μέτρο απόστασης $Dist(\cdot, \cdot)$

- Χρησιμοποιείται για τη σύγκριση στιγμιοτύπων.
- Μπορεί να είναι:
 - Ευκλείδεια απόσταση
 - Manhattan απόσταση
 - Cosine απόσταση
 - ή οποιοδήποτε άλλο κατάλληλο μέτρο

2. Μέθοδος σύνδεσης (linkage) $\mathcal{L}$

- Ορίζει πώς υπολογίζεται η απόσταση μεταξύ **ολόκληρων συστάδων**.
- Δηλαδή: αν δύο συστάδες περιέχουν πολλά σημεία, πώς ορίζουμε την απόστασή τους;

Πρόβλημα: Απόσταση μεταξύ Συστάδων

Έστω δύο συστάδες $\mathcal{C}_A$ και $\mathcal{C}_B$.

Αν κάθε συστάδα περιέχει πολλά στιγμιότυπα, πώς πρέπει να οριστεί η μεταξύ τους απόσταση;

Πιθανές επιλογές:

- Απόσταση μεταξύ των δύο πλησιέστερων σημείων;
- Απόσταση μεταξύ των δύο πιο απομακρυσμένων σημείων;
- Μέση απόσταση όλων των ζευγών σημείων;
- Απόσταση μεταξύ των κέντρων των συστάδων;

Η επιλογή αυτή καθορίζει τη γεωμετρία της τελικής συσταδοποίησης.

Μέθοδοι Σύνδεσης (Linkage Methods)

1. Μονή σύνδεση (Single Linkage)

$$d(\mathcal{C}_A, \mathcal{C}_B) = \min_{x \in \mathcal{C}_A, y \in \mathcal{C}_B} Dist(x, y)$$

- Βασίζεται στα πιο κοντινά σημεία.
- Τείνει να δημιουργεί επιμήκεις (αλυσιδωτές) συστάδες.

2. Πλήρης σύνδεση (Complete Linkage)

$$d(\mathcal{C}_A, \mathcal{C}_B) = \max_{x \in \mathcal{C}_A, y \in \mathcal{C}_B} Dist(x, y)$$

- Βασίζεται στα πιο απομακρυσμένα σημεία.
- Τείνει να δημιουργεί πιο συμπαγείς συστάδες.

Μέθοδοι Σύνδεσης (συνέχεια)

3. Μέση σύνδεση (Average Linkage)

$$d(\mathcal{C}_A, \mathcal{C}_B) = \frac{1}{|\mathcal{C}_A||\mathcal{C}_B|} \sum_{x \in \mathcal{C}_A} \sum_{y \in \mathcal{C}_B} Dist(x, y)$$

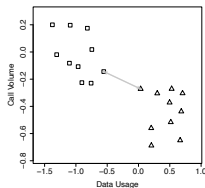
- Χρησιμοποιεί τον μέσο όρο όλων των αποστάσεων.
- Πιο «ισορροπημένη» προσέγγιση.

4. Σύνδεση κέντρων (Centroid Linkage)

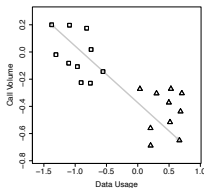
$$d(\mathcal{C}_A, \mathcal{C}_B) = Dist(\mu_A, \mu_B)$$

όπου μ_A και μ_B τα κέντρα των συστάδων.

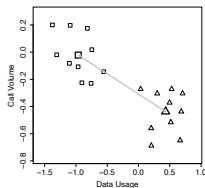
- Παρόμοια φιλοσοφία με k-means.



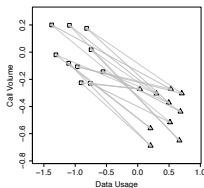
(α) single



(β) complete



(γ) centroid



(δ) average

Σχήμα 8: (a)--(d) Different linkage methods that can be used to compare the distances between clusters in agglomerative hierarchical clustering. (Arrows for only some indicative distances are shown in the average linkage diagram (d).)

Δενδρογράμμα (Dendrogram)

Η ιεραρχική φύση του AHC απεικονίζεται μέσω ενός **δενδρογράμματος**.

Σε ένα δενδρογράμμα:

- Κάθε στιγμιότυπο εμφανίζεται στο κάτω μέρος με το αναγνωριστικό του.
- Οι οριζόντιες γραμμές δείχνουν πού συγχωνεύονται συστάδες.
- Τα κάθετα ύψη συγχώνευσης αντιστοιχούν στην **απόσταση μεταξύ των συστάδων**.

Παρατήρηση:

- Στα χαμηλά επίπεδα οι συγχωνεύσεις γίνονται σε μικρές αποστάσεις.
- Καθώς ανεβαίνουμε στην ιεραρχία, συγχωνεύονται όλο και πιο απομακρυσμένες συστάδες.

Επιλογή του αριθμού συστάδων k

Ένα σημαντικό πλεονέκτημα του AHC:

Δεν απαιτείται προκαθορισμένη τιμή k .

Ο αλγόριθμος παράγει μια πλήρη ιεραρχική δομή, η οποία μπορεί να «κοπεί» σε οποιοδήποτε επίπεδο.

- Κόβοντας το δέντρο χαμηλά → πολλές μικρές συστάδες.
- Κόβοντας το δέντρο ψηλά → λίγες μεγάλες συστάδες.

Η βέλτιστη επιλογή του k μπορεί να καθοριστεί με:

- silhouette score
- elbow method (σε WCSS μετά το cut)
- άλλους δείκτες αξιολόγησης

Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

Η PCA είναι τεχνική:

- Μείωσης διαστατικότητας (dimensionality reduction)
- Εξαγωγής δομής από δεδομένα
- Χωρίς χρήση ετικετών (labels)

Άρα ανήκει στο:

Μη-Επιβλεπόμενη Μάθηση

Στόχος:

Να ανακαλύψει τη βασική δομή μεταβλητότητας των δεδομένων.

Διαισθητική Ιδέα της PCA

Δοθέντων δεδομένων σε υψηλή διάσταση:

- Μπορούμε να τα προβάλουμε σε λιγότερες διαστάσεις
- Χωρίς να χάσουμε μεγάλο μέρος της πληροφορίας

Η PCA βρίσκει:

- Τις κατευθύνσεις μέγιστης διακύμανσης
- Νέους άξονες που είναι:
 - Ορθογώνιοι μεταξύ τους
 - Γραμμικοί συνδυασμοί των αρχικών μεταβλητών

Μαθηματικός Ορισμός

Έστω πίνακας δεδομένων:

$$X \in \mathbb{R}^{n \times p}$$

(κεντραρισμένος: μέσος όρος = 0)

Η PCA βρίσκει διάνυσμα w που μεγιστοποιεί:

$$\max_w \text{Var}(Xw)$$

υπό τον περιορισμό:

$$\|w\| = 1$$

Σχέση με τον Πίνακα Συνδιακύμανσης

Ο πίνακας συνδιακύμανσης είναι:

$$\Sigma = \frac{1}{n} X^T X$$

Η PCA επιλύει το ιδιοτιμικό πρόβλημα:

$$\Sigma w = \lambda w$$

όπου:

- w = ιδιοδιανύσματα (principal components)
- λ = ιδιοτιμές (εξηγούμενη διακύμανση)

Ερμηνεία Ιδιοτιμών

Κάθε ιδιοτιμή λ_k εκφράζει:

Πόση διακύμανση εξηγεί η k-οστή κύρια συνιστώσα

Ποσοστό εξηγούμενης διακύμανσης:

$$\frac{\lambda_k}{\sum_{j=1}^p \lambda_j}$$

Αλγοριθμικά Βήματα PCA

- 1 Κεντράρισμα δεδομένων
- 2 Υπολογισμός πίνακα συνδιακύμανσης
- 3 Υπολογισμός ιδιοτιμών & ιδιοδιανυσμάτων
- 4 Ταξινόμηση κατά φθίνουσα ιδιοτιμή
- 5 Επιλογή πρώτων k συνιστωσών
- 6 Προβολή δεδομένων:

$$Z = XW_k$$

PCA και Clustering

Η PCA χρησιμοποιείται συχνά πριν από clustering:

- Μείωση θορύβου
- Μείωση διαστατικότητας
- Οπτικοποίηση (2D / 3D)

Παράδειγμα:

PCA $\rightarrow$ k-means $\rightarrow$ Καλύτερη σταθερότητα

Άρα η PCA λειτουργεί ως:

Unsupervised Feature Extraction

Γιατί η PCA είναι Unsupervised;

- Δεν χρησιμοποιεί ετικέτες στόχου
- Βασίζεται μόνο στη δομή συνδιακύμανσης
- Ανακαλύπτει κρυμμένες κατευθύνσεις μεταβλητότητας

Η PCA:

Ανακαλύπτει δομή χωρίς επίβλεψη

Σύνοψη

- Οι τεχνικές μη-επιβλεπόμενης μάθησης χρησιμοποιούνται όταν δεν υπάρχει μεταβλητή-στόχος και μοντελοποιούν την υποκείμενη δομή που βρίσκεται μέσα στα περιγραφικά χαρακτηριστικά ενός συνόλου δεδομένων.
- Μπορούμε να θεωρήσουμε ότι η έξοδος των περισσότερων μη-επιβλεπόμενων μοντέλων είναι νέα παραγόμενα χαρακτηριστικά που μπορούν να προστεθούν στο αρχικό σύνολο δεδομένων ώστε να το **ενισχύσουμε** ή να το **εμπλουτίσουμε**.
- Παρουσιάστηκαν αναλυτικά δύο τεχνικές ομαδοποίησης: η **k -means ομαδοποίηση** και η AHC.
- Παρουσιάστηκε η Principal Component Analysis (PCA).

Εμπειρική Εφαρμογή

Εφαρμογή: Κίνδυνος χώρας (Country Risk)

Στόχος είναι η ομαδοποίηση (clustering) των χωρών ανάλογα με την επικινδυνότητά τους για ξένες επενδύσεις χρησιμοποιώντας δεδομένα του 2019. Ανοίξτε τα αρχεία

[Country_risk_2019_data.csv](#), [5.1 K-means_elbow.ipynb](#) , [5.2 K-means_silhouette.ipynb](#)

Μέτρα κινδύνου χώρας

- Πραγματικός ρυθμός ανάπτυξης του ΑΕΠ (ΔΝΤ) - **GDP real growth rate** (IMF)
- Δείκτης διαφθοράς (Διεθνής Διαφάνεια)- **Corruption index** (Transparency international)
- Δείκτης Ειρήνης (Ινστιτούτο Οικονομικών και Ειρήνης) - **Peace index** (Institute for Economics and Peace)
- Δείκτης νομικού κινδύνου (Σύλλογος δικαιωμάτων ιδιοκτησίας) - **Legal Risk Index** (Property Rights Association)

❖ Συλλέχθηκαν δεδομένα για 121 χώρες. Χρησιμοποιήθηκε κλίμακα **Z-score**.

Μέρος των δεδομένων

Country	Corruption Index	Peace Index	Legal Risk Index	Real GDP growth rate (% per yr)
Albania	35	1.821	4.546	2.983
Algeria	35	2.219	4.435	2.553
Argentina	45	1.989	5.087	-3.061
Armenia	42	2.294	4.812	6.000
Australia	77	1.419	8.363	1.713
Austria	77	1.291	8.089	1.605

```
In [5]: # Load raw data
DATA_FOLDER = './'
raw = pd.read_csv(os.path.join(DATA_FOLDER, 'Country_risk_2019_data.csv'))

# check the raw data
print("Size of the dataset (row, col): ", raw.shape)
print("\nFirst 5 rows\n", raw.head(n=5))
print("\nFirst 5 rows and 5 columns\n", raw.iloc[:5, :5])
```

Size of the dataset (row, col): (121, 6)

First 5 rows

	Country	Abbrev	Corruption	Peace	Legal	GDP Growth
0	Albania	AL	35	1.821	4.546	2.983
1	Algeria	DZ	35	2.219	4.435	2.553
2	Argentina	AR	45	1.989	5.087	-3.061
3	Armenia	AM	42	2.294	4.812	6.000
4	Australia	AU	77	1.419	8.363	1.713

First 5 rows and 5 columns

	Country	Abbrev	Corruption	Peace	Legal
0	Albania	AL	35	1.821	4.546
1	Algeria	DZ	35	2.219	4.435
2	Argentina	AR	45	1.989	5.087
3	Armenia	AM	42	2.294	4.812
4	Australia	AU	77	1.419	8.363

Συνοπτικά στατιστικά στοιχεία (Summary statistics) και Πίνακας συσχετίσεων (correlation matrix)

```
In [6]: # print summary statistics
print("\nSummary statistics\n", raw.describe())
print("\nCorrelation matrix\n", raw.corr())
```

Summary statistics

	Corruption	Peace	Legal	GDP Growth
count	121.000000	121.000000	121.000000	121.000000
mean	46.842975	2.001017	5.752529	2.657529
std	18.702499	0.461485	1.373932	2.563741
min	15.000000	1.072000	2.671000	-9.459000
25%	33.000000	1.699000	4.785000	1.249000
50%	41.000000	1.939000	5.455000	2.600000
75%	60.000000	2.294000	6.488000	4.000000
max	87.000000	3.369000	8.712000	7.800000

Correlation matrix

	Corruption	Peace	Legal	GDP Growth
Corruption	1.000000	-0.705002	0.938512	-0.123545
Peace	-0.705002	1.000000	-0.662233	-0.004428
Legal	0.938512	-0.662233	1.000000	-0.150369
GDP Growth	-0.123545	-0.004428	-0.150369	1.000000

- Ο Δείκτης διαφθοράς (Corruption index) και ο Δείκτης νομικού κινδύνου (Legal Risk) είναι υψηλά συσχετιζόμενοι (correlation = 0.939).
- Για τον λόγο αυτόν στον αλγόριθμο θα χρησιμοποιήσουμε τους παρακάτω δείκτες: GDP growth rate, Peace index, Legal risk index
- Ας δούμε πρώτα τα δεδομένα μετά το scaling στην επόμενη διαφάνεια προτού προχωρήσουμε στον αλγόριθμο.

Δεδομένα μετά την κλίμακα Z-score

Country	Corruption Index	Peace Index	Legal Risk Index	Real GDP growth rate (% per yr)
Albania	-0.633	-0.390	-0.878	0.127
Algeria	-0.633	0.472	-0.959	-0.041
Argentina	-0.099	-0.026	-0.484	-2.231
Armenia	-0.259	0.635	-0.685	1.304
Australia	1.612	-1.261	1.900	-0.368

```
In [9]: #K means cluster
#Pick features & normalization
#Since Corruption and Legal are highly correlated, we drop the Courrup
X = raw[['Peace', 'Legal', 'GDP Growth']]
X = (X - X.mean()) / X.std()
print(X.head(5))
```

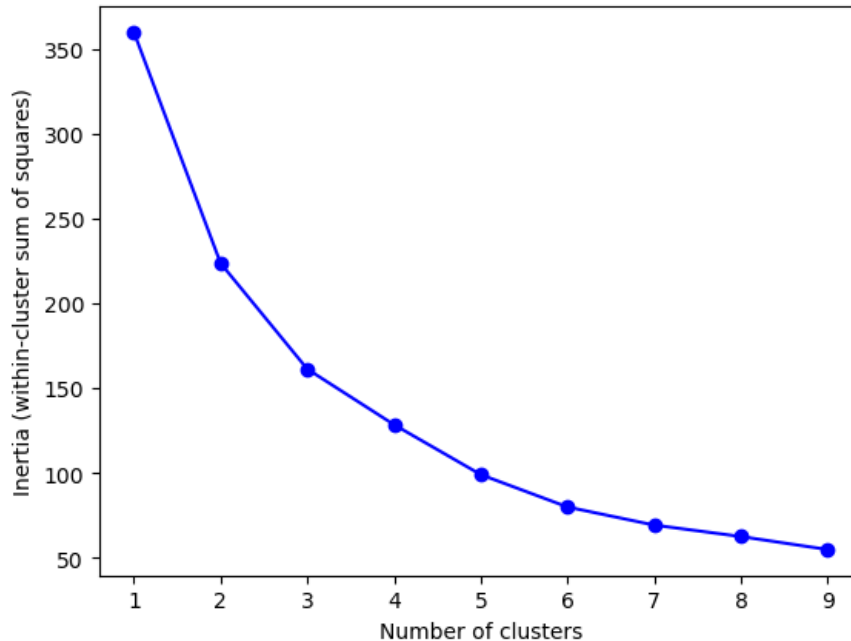
```
      Peace      Legal  GDP Growth
0 -0.390081 -0.878158    0.126952
1  0.472352 -0.958948   -0.040772
2 -0.026039 -0.484397   -2.230541
3  0.634871 -0.684553    1.303747
4 -1.261182  1.900001   -0.368418
```

I. Προσέγγιση elbow

Πώς το συνολικό άθροισμα τετραγώνων εντός του *cluster* μειώνεται καθώς το *k* αυξάνεται όταν χρησιμοποιείται ο αλγόριθμος *k-means*

```
In [10]: #Perform elbow method
# https://stackoverflow.com/questions/41540751/sklearn-kmeans-equivalent-of-elbow-method
Ks = range(1, 10)
inertia = [KMeans(i).fit(X).inertia_ for i in Ks]

fig = plt.figure()
plt.plot(Ks, inertia, '-bo')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia (within-cluster sum of squares)')
plt.show()
```



Inertia
(Αδράνεια)

Με βάση το διάγραμμα προτείνεται $k=3$, άρα 3 clusters.

Cluster centers

	Peace index	Legal index	GDP
High risk	1.23	-0.83	-1.08
Moderate risk	-0.85	1.02	-0.24
Low risk	0.23	-0.54	0.65

Σημειώστε ότι οι υψηλές τιμές για τον peace index είναι κακές ενώ οι υψηλές τιμές για τον legal risk index είναι καλές.

```
In [11]: #K means with k=3
k = 3
kmeans = KMeans(n_clusters=k, random_state=1)
kmeans.fit(X)

# print inertia & cluster center
print("inertia for k=3 is", kmeans.inertia_)
print("cluster centers: ", kmeans.cluster_centers_)

# take a quick look at the result
y = kmeans.labels_
print("cluster labels: ", y)

inertia for k=3 is 161.1333871005255
cluster centers: [[ 1.22506036 -0.83385901 -1.07842464]
 [-0.85097477  1.02149992 -0.23897931]
 [ 0.23006626 -0.54045468  0.65506397]]
cluster labels: [2 2 0 2 1 1 2 2 2 1 2 2 2 1 0 2 0 2 1 0 1 2 2 1 2 1 1 0 1 2 0 2 2 1 2 1 1
 2 2 1 2 2 2 2 1 1 2 2 0 1 2 1 1 1 1 1 2 2 1 1 1 0 0 1 2 2 1 2 2 1 0 2 2 2 2
 2 1 1 0 0 1 1 0 2 0 2 2 1 1 1 1 0 2 0 2 2 2 1 1 1 0 1 2 1 1 1 2 2 2 0 2 0
 2 0 1 1 1 1 2 0 2 0]
```


Cluster centers

	Peace index	Legal index	GDP
(High)1	0.21	0.53	-0.59
2	0.25	-0.53	1.35
3	-1.26	1.79	-0.53
4	1.85	-1.05	-0.15
5	0.71	-0.96	-3.44
6	0.05	-0.69	-0.02
(Low)7	-0.85	0.53	0.20

```
] : #K means with k=7
k = 7
kmeans = KMeans(n_clusters=k, random_state=1)
kmeans.fit(X)

# print cluster center
print("cluster centers: ", kmeans.cluster_centers_)

# take a quick look at the result
y = kmeans.labels_
print("cluster labels: ", y)

cluster centers: [[ 0.20555045  0.52556914 -0.59243165]
 [ 0.25388748 -0.52823312  1.34995125]
 [-1.26104619  1.7919167  -0.52568349]
 [ 1.84989234 -1.05242316 -0.14505601]
 [ 0.70529573 -0.95894794 -3.43893096]
 [ 0.04538919 -0.6852538  -0.01515842]
 [-0.84739893  0.52786017  0.19939546]]
cluster labels: [5 5 4 1 2 2 5 0 1 2 1 5 5 6 0 6 3 3 2 3 6 1 3 6 5 6 6 3 2 1 5 1 5 6 1 2 0
 5 1 2 1 5 5 5 6 2 1 1 4 6 0 0 0 2 0 5 1 0 0 6 3 5 6 1 5 6 3 1 6 3 5 5 5 5
 1 2 2 4 3 2 0 3 6 5 5 1 6 6 6 6 3 1 0 1 5 1 2 6 6 0 6 5 2 2 6 1 5 5 0 5 3
 1 3 0 2 0 0 1 3 5 4]
```

Hierarchical Clustering

- Για τα μέτρα εγγύτητας των clusters μπορείτε να συμβουλευτείτε το [AgglomerativeClustering documentation της Python](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html)

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html>

linkage : {'ward', 'complete', 'average', 'single'}, default='ward'

Which linkage criterion to use. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

- 'ward' minimizes the variance of the clusters being merged.
- 'average' uses the average of the distances of each observation of the two sets.
- 'complete' or 'maximum' linkage uses the maximum distances between all observations of the two sets.
- 'single' uses the minimum of the distances between all observations of the two sets.

Εφαρμογή: Κίνδυνος χώρας (όταν όλα τα χαρακτηριστικά χρησιμοποιούνται)

- Εφαρμογή του Hierarchical Clustering με βάση τη **μέση απόσταση**

Ανοίξτε τα αρχεία

Country_risk_2019_data.csv, 5.3 hierarchical_clustering_averagemethod.ipynb

- Εφαρμογή του Hierarchical Clustering με βάση τη **μέθοδο Ward**.

Ανοίξτε τα αρχεία

Country_risk_2019_data.csv, 5.4 hierarchical_clustering_wardmethod.ipynb

Principal Components Analysis (PCA)

Εφαρμογή: Κίνδυνος χώρας (όταν όλα τα χαρακτηριστικά χρησιμοποιούνται)

- Ανοίξτε το αρχείο *Country_risk_2019_data.csv*, *5.5 PCA example.ipynb*
- Factor Loadings

	PC1	PC2	PC3	PC4
Corruption index	- 0.602	-0.015	0.328	0.728
Peace index	0.524	0.201	0.825	0.065
Legal risk index	-0.594	0.022	0.425	-0.683
GDP Growth rate	0.103	-0.979	0.174	-0.013

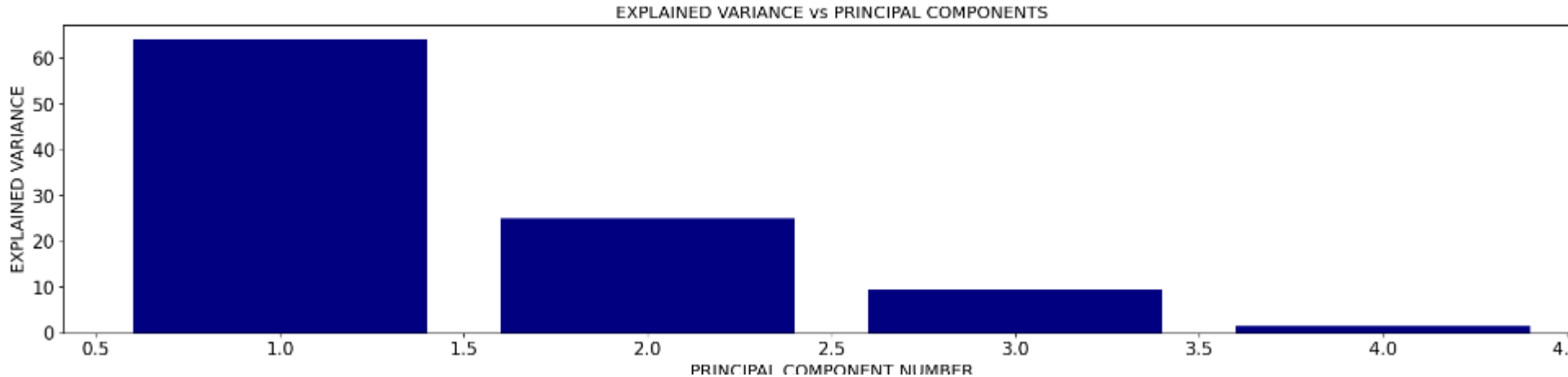
```
[9]: array([[ -0.60188457,  0.52361257, -0.59403386,  0.10338576],  
          [ -0.01512074,  0.20084677,  0.02192676, -0.97926051],  
          [ 0.32824929,  0.82539386,  0.42519586,  0.17374078],  
          [ 0.72784526,  0.06492634, -0.68253313, -0.01320492]])
```

- Ο πρώτος PCA factor έχει περίπου τις ίδιες σταθμίσεις στους Corruption, Peace, Legal Risk indices.

```
In [10]: # Percentage of variation explained by successive eigenvectors/PCS  
model.explained_variance_ratio_.round(2)
```

```
Out[10]: array([0.64, 0.25, 0.09, 0.01])
```

```
Out[11]: <BarContainer object of 4 artists>
```



- Ο πρώτος PCA factor εξηγεί το 64% της διακύμανσης.
- Ο δεύτερος PCA factor εξηγεί το 25% της διακύμανσης.
- Ο τρίτος PCA factor εξηγεί το 9% της διακύμανσης.
- Ο τέταρτος PCA factor εξηγεί το 1% της διακύμανσης.