# Αναγνώριση Προτύπων

## Εργασία 3


Ονοματεπώνυμο: Μωραΐτη Παναγιώτα

ΑΜ: 58054

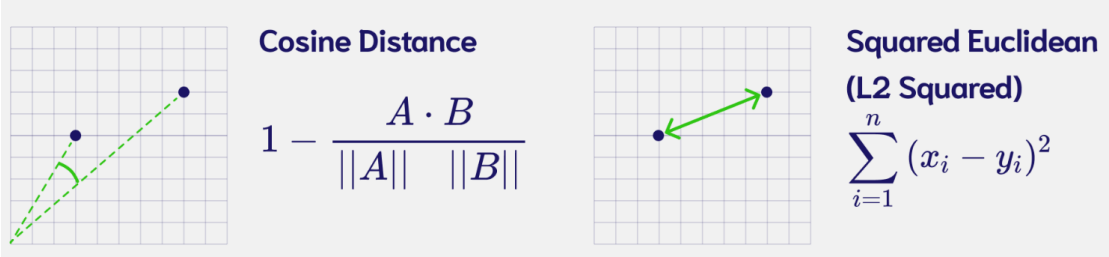
## Πίνακας περιεχομένων

## Άσκηση 1

Το seed dataset περιέχει μορφολογικές μετρήσεις 210 σπόρων από τρεις ποικιλίες σιτηρών (ω1: Kama, ω2: Rosa, ω3: Canadian).

### Α. Απεικονίστε τον πίνακα αποστάσεων των δεδομένων για Ευκλείδεια και cosine μετρικές. Ποιες κλάσεις πιστεύετε ότι είναι ευκολότερο να διαχωριστούν μεταξύ τους; Γιατί;

Οι μετρικές που χρησιμοποιήθηκαν είναι η Ευκλείδεια και cosine απόσταση.



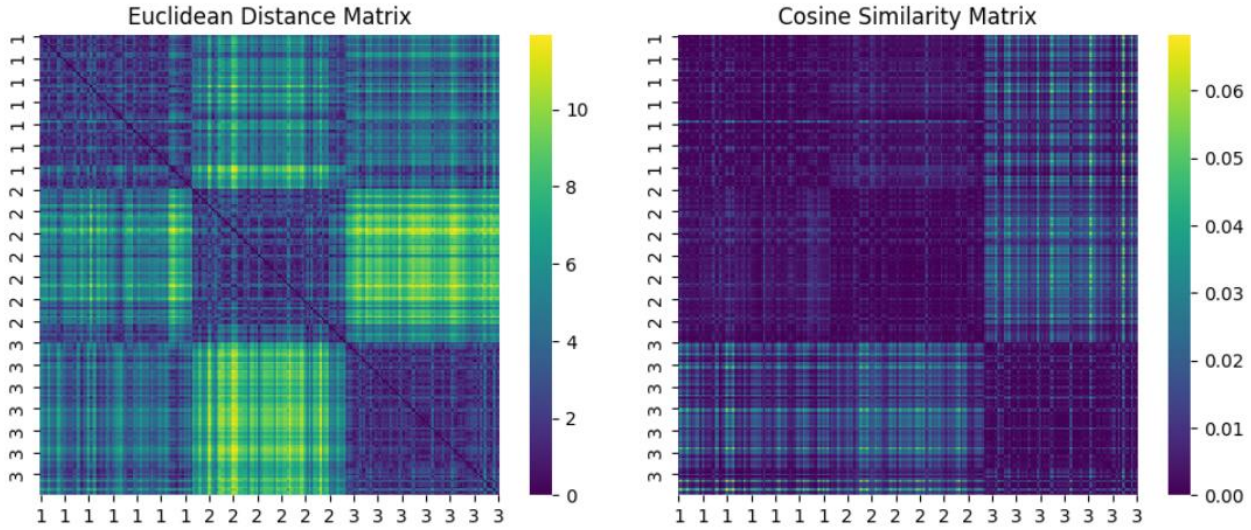

Οι πίνακες των αποστάσεων για τις δύο μετρικές φαίνονται στην παρακάτω εικόνα. Όσο πλησιάζουμε στο μηδέν (σκούρο χρώμα), σημαίνει ότι τα συγκεκριμένα δείγματα έχουν πολύ μικρή απόσταση μεταξύ τους, δηλαδή μοιάζουν πάρα πολύ. Για αυτό το λόγο παρατηρούμε μια μαύρη γραμμή στην κύρια διαγώνιο του κάθε πίνακα, η απόσταση αυτή είναι μηδέν, καθώς αντιπροσωπεύει την απόσταση κάθε δείγματος από τον εαυτό του. Στα σημεία που οι πίνακες έχουν πιο φωτεινό χρώμα, σημαίνει ότι η απόσταση είναι μεγαλύτερη, άρα έχουμε μικρότερη ομοιότητα. Στα τρία τετράγωνα της διαγωνίου των πινάκων, υπάρχουν οι αποστάσεις των σημείων της κάθε κλάσεις από σημεία της ίδιας κλάσης, οπότε εκεί περιμένουμε εξ αρχής να υπάρχει μεγάλη ομοιότητα, άρα και πιο σκούρο χρώμα, κάτι που επιβεβαιώνεται.

Όσον αφορά την Ευκλείδεια απόσταση οι κλάσεις που μοιάζουν λιγότερο μεταξύ τους (έχουν μεγαλύτερη απόσταση-φωτεινότερο χρώμα), είναι οι 2 με 3. Στη συνέχεια, ακολουθούν οι 1 με 2. Τέλος, οι 1 με τη 3 φαίνεται να μοιάζουν περισσότερο, πάλι όμως υπάρχει κάποια απόσταση. Η κλάση 2 φαίνεται να μπορεί να διαχωριστεί ευκολότερα από τις υπόλοιπες.

Όσον αφορά την cosine απόσταση (1-cosine similarity) οι κλάσεις που μοιάζουν λιγότερο μεταξύ τους, είναι οι 1 με 3 και 2 με 3. Οι 1 με 2 φαίνεται να μοιάζουν σε τέτοιο βαθμό, όσο και η κάθε κλάση με τον εαυτό της (παρόμοιο χρώμα με τα τρία τετράγωνα της διαγωνίου). Η κλάση 3 φαίνεται να μπορεί να διαχωριστεί ευκολότερα από τις υπόλοιπες.

Οπότε, οι κλάσεις που μοιάζουν λιγότερο (μεγαλύτερη απόσταση), είναι ευκολότερο να διαχωριστούν μεταξύ τους, χρησιμοποιώντας την αντίστοιχη μετρική.

Με την Ευκλείδεια απόσταση είναι πιθανό να μπορέσουν να διαχωριστούν και οι 3 κλάσεις, επειδή φαίνεται τα σημεία των 3 κλάσεων να απέχουν αρκετά μεταξύ τους. Όμως με την cosine απόσταση, είναι πιθανό οι κλάσεις 1 με 2 να μην μπορέσουν να διαχωριστούν, καθώς τα σημεία τους είναι πολύ κοντά (ακόμα και να διαχωριστούν τα λάθη μεταξύ αυτών των δύο θα είναι περισσότερα).

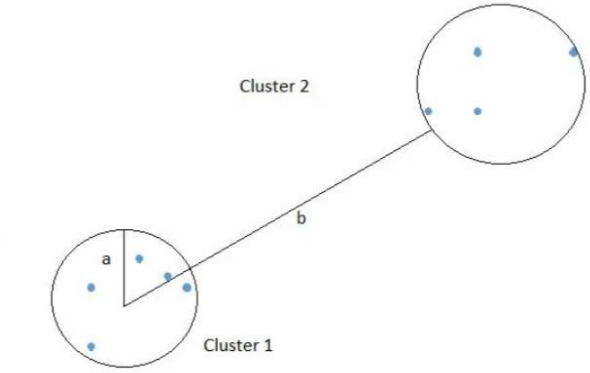



## Β. Να υπολογίσετε το Silhouette Coefficient για την ομαδοποίηση των 7-διάστατων δεδομένων σε k=2,3,..10 κλάσεις με τη μέθοδο k-means και Ευκλείδεια ή squared Euclidean μετρική. Απεικονίστε το διάγραμμα του Silhouette και σχολιάστε ποιος είναι ο βέλτιστος αριθμός κλάσεων σύμφωνα με το κριτήριο αυτό.

Ο k-means είναι ένας αλγόριθμος συσταδοποίησης (clustering) που χρησιμοποιείται για να ομαδοποιήσει ένα σύνολο δεδομένων σε k ομάδες. Επιδιώκει να μειώσει τη διακύμανση εντός της κάθε ομάδας, βρίσκοντας κέντρα που ομαδοποιούν δεδομένα με παρόμοιες ιδιότητες. Η υλοποίηση k-means στη βιβλιοθήκη scikit-learn χρησιμοποιεί squared Euclidean μετρική.

Το Silhouette Coefficient είναι ένας δείκτης που χρησιμοποιείται για να μετρήσει την ποιότητα της ομαδοποίησης σε ένα σύνολο δεδομένων. Αυτός ο δείκτης λαμβάνει υπόψιν το πόσο καλά χωρίζονται οι ομάδες και πόσο κοντά είναι τα μέλη μιας ομάδας μεταξύ τους. Η τιμή του κυμαίνεται από -1 έως 1. Η τιμή 1 σημαίνει ότι τα clusters είναι απομακρυσμένα, οπότε οι κλάσεις μπορούν να διαχωριστούν. Η τιμή 0 σημαίνει ότι η απόσταση μεταξύ των clusters δεν είναι αρκετά μεγάλη. Η τιμή -1 σημαίνει ότι τα clusters έχουν διαχωριστεί με λάθος τρόπο. Γενικά, μια υψηλή τιμή σημαίνει ότι η ποιότητα της ομαδοποίησης είναι καλή.

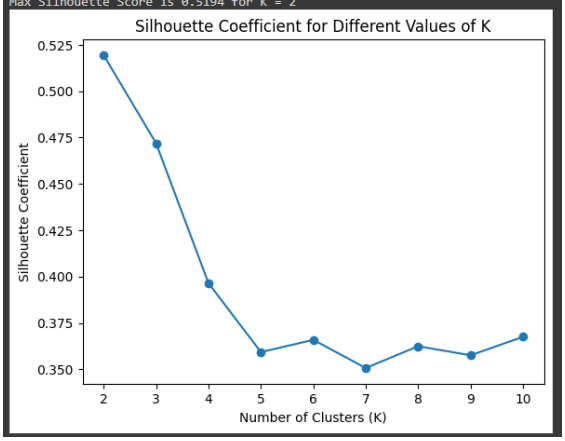
Silhouette Score = (b-a)/max(a,b)




Η μεγαλύτερη τιμή είναι περίπου 0.519 για K=2. Δηλαδή για δύο ομάδες έχουμε τον καλύτερο διαχωρισμό. Βέβαια, κάθε φορά που θα τρέξουμε τον αλγόριθμο, επειδή η αρχικοποίηση των κέντρων είναι τυχαία, θα πάρουμε λίγο διαφορετικό αποτέλεσμα, όμως ο βέλτιστος αριθμός κλάσεων παραμένει 2. Ο αριθμός αυτός δε συμπίπτει με τον πραγματικό αριθμό κλάσεων που υπάρχουν στο dataset.

Οπότε, ο βέλτιστος αριθμός κλάσεων K, είναι ο αριθμός για τον οποίο το Silhouette Coefficient παίρνει τη μέγιστη τιμή.

```
Silhouette Score for K = 2 is: 0.5194
Silhouette Score for K = 3 is: 0.4719
Silhouette Score for K = 4 is: 0.3965
Silhouette Score for K = 5 is: 0.3593
Silhouette Score for K = 6 is: 0.3659
Silhouette Score for K = 7 is: 0.3507
Silhouette Score for K = 8 is: 0.3625
Silhouette Score for K = 9 is: 0.3576
Silhouette Score for K = 10 is: 0.3675

Max Silhouette Score is 0.5194 for K = 2
```



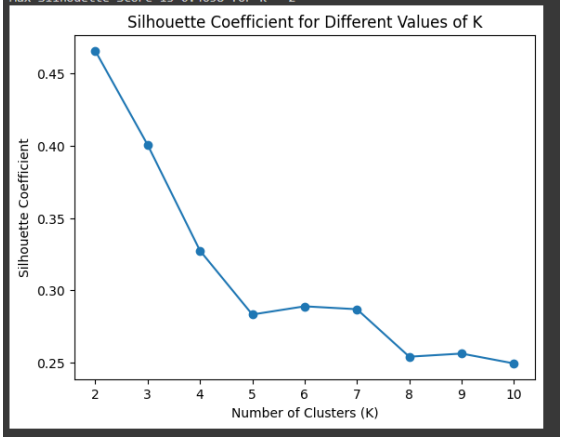

## Γ. Να κανονικοποιηθούν τα δεδομένα ώστε κάθε χαρακτηριστικό να έχει μηδενική τιμή και μοναδιαία variance. Υπολογίσετε εκ νέου το Silhouette Coefficient στα κανονικοποιημένα δεδομένα για cosine μετρική. Απεικονίστε το νέο διάγραμμα του Silhouette. Τι παρατηρείτε;

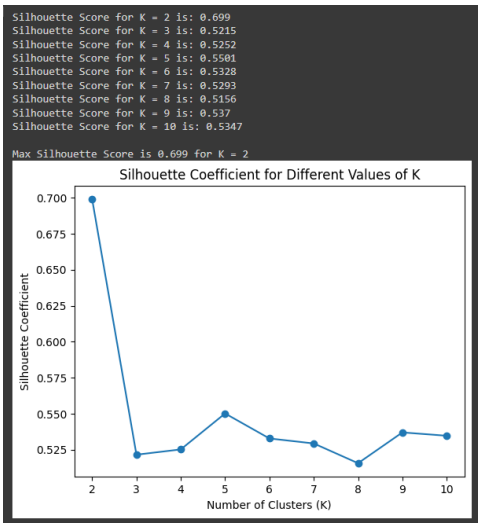
Για να κανονικοποιηθούν τα δεδομένα ώστε κάθε χαρακτηριστικό να έχει μηδενική τιμή και μοναδιαία variance, πρέπει να υπολογίσουμε την μέση τιμή και την τυπική απόκλιση για κάθε χαρακτηριστικό. Στη συνέχεια, από κάθε δείγμα αφαιρούμε την μέση τιμή και διαιρούμε με την τυπική απόκλιση για κάθε χαρακτηριστικό.

Αν με τα κανονικοποιημένα δεδομένα υπολογίσουμε εκ νέου το Silhouette Coefficient για squared Euclidean μετρική παίρνουμε τα αποτελέσματα που φαίνονται στη διπλανή εικόνα.

Η μεγαλύτερη τιμή του Silhouette Coefficient είναι 0.4658 για K=2. Πάλι ο βέλτιστος αριθμός κλάσεων που βρίσκουμε είναι 2, που ξέρουμε ότι δεν συμπίπτει με τον πραγματικό αριθμό κλάσεων στο dataset. Άρα ούτε με την κανονικοποίηση των δεδομένων καταφέραμε να βρούμε τον πραγματικό αριθμό κλάσεων.

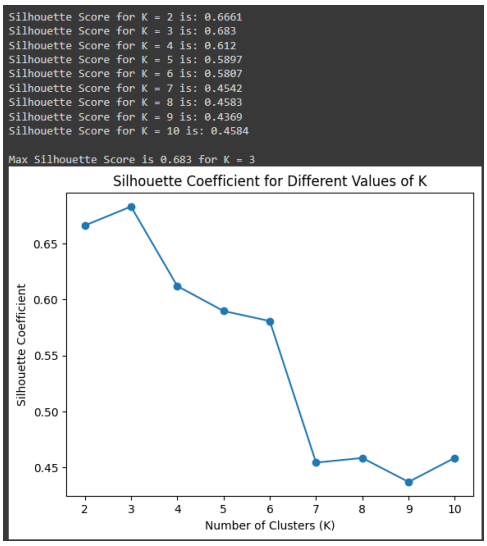
```
Silhouette Score for K = 2 is: 0.4658
Silhouette Score for K = 3 is: 0.4007
Silhouette Score for K = 4 is: 0.3276
Silhouette Score for K = 5 is: 0.2834
Silhouette Score for K = 6 is: 0.289
Silhouette Score for K = 7 is: 0.287
Silhouette Score for K = 8 is: 0.2542
Silhouette Score for K = 9 is: 0.2564
Silhouette Score for K = 10 is: 0.2495

Max Silhouette Score is 0.4658 for K = 2
```






Η βιβλιοθήκη scikit-learn δεν υποστηρίζει k-means με cosine μετρική, οπότε θα χρησιμοποιήσουμε την προσέγγιση της κανονικοποίησης των διανυσμάτων σε μοναδιαίο μήκος ακολουθούμενο από k-means με Ευκλείδεια μετρική. Για κάθε δείγμα βρίσκουμε το μήκος του διανύσματος (μήκος κάθε γραμμής) και διαιρούμε κάθε χαρακτηριστικό του δείγματος με αυτό το μήκος.

Παρατηρούμε ότι ο βέλτιστος αριθμός κλάσεων με βάση το Silhouette Coefficient είναι K=2 και το Silhouette Coefficient παίρνει την τιμή 0.699. Πάλι ο βέλτιστος αριθμός κλάσεων που βρίσκουμε είναι 2, που ξέρουμε ότι δεν συμπίπτει με τον πραγματικό αριθμό κλάσεων στο dataset.




Αν με τα κανονικοποιημένα δεδομένα υπολογίσουμε εκ νέου το Silhouette Coefficient για cosine μετρική παίρνουμε τα αποτελέσματα που φαίνονται στη διπλανή εικόνα.

Πλέον παρατηρούμε ότι ο βέλτιστος αριθμός κλάσεων είναι K=3 και το Silhouette Coefficient παίρνει την τιμή 0.683. Οπότε, έχουμε καλύτερη ομαδοποίηση από πριν. Επίσης, ο αριθμός K συμπίπτει με τον πραγματικό αριθμό των κλάσεων που υπάρχουν στο dataset. Άρα σε αυτήν την περίπτωση η κανονικοποίηση μας βοήθησε να βρούμε το βέλτιστο αριθμό κλάσεων που συμπίπτει και με τον πραγματικό αριθμό κλάσεων που υπάρχουν στο dataset.

## Δ. Ομαδοποιείστε τα δεδομένα σε 3 κλάσεις με τη μέθοδο k-means και squared Euclidean μετρική. Υπολογίστε το Rand Index για τη σύγκριση της παραγόμενης ομαδοποίησης με τις ετικέτες του dataset. Επαναλάβετε 5 φορές (με τυχαία αρχικοποίηση κέντρων), και υπολογίστε την μέση τιμή και το variance του Rand Index.

Το Rand Index υπολογίζει ένα μέτρο ομοιότητας μεταξύ δύο ομαδοποιήσεων σε ένα σύνολο δεδομένων. Είναι χρήσιμο για να μετρήσει την ποιότητα μιας ομαδοποίησης. Παίρνει τιμές από 0 έως 1, με υψηλότερες τιμές να υποδηλώνουν καλύτερη ομοιότητα μεταξύ των ομαδοποιήσεων. Ο Rand Index είναι ειδικά χρήσιμος όταν οι ομαδοποιήσεις δεν είναι υποχρεωτικά ίδιες σε κάθε εκτέλεση, αλλά μπορεί να υπάρχουν μικρές διαφοροποιήσεις.

Στην ουσία, ο Rand Index μετρά τη συμφωνία μεταξύ μιας ομαδοποίησης και των πραγματικών ετικετών (ground truth labels).

Υψηλότερες τιμές Rand Index υποδηλώνουν καλύτερη ομοιότητα με την πραγματική ομαδοποίηση. Για την τιμή 0, καμία ομοιότητα μεταξύ των ομαδοποιήσεων, οι ομαδοποιήσεις είναι πλήρως διαφορετικές. Για την τιμή 1, πλήρης ομοιότητα μεταξύ των ομαδοποιήσεων, οι ομαδοποιήσεις είναι ακριβώς οι ίδιες, όλα τα δείγματα έχουν ομαδοποιηθεί σωστά, σύμφωνα με τις πραγματικές ετικέτες.

$$RI = \frac{\text{Number of Agreeing Pairs}}{\text{Number of Pairs}}$$

Τα αποτελέσματα για 5 τυχαίες αρχικοποιήσεις κέντρων με τη χρήση squared Euclidean μετρικής είναι: Mean Rand Index: **0.8744**, Variance of Rand Index: **0.0**.

Η διακύμανση του Rand Index είναι 0.0, κάτι που υποδηλώνει ότι οι τιμές του Rand Index για τις 5 τυχαίες αρχικοποιήσεις ήταν ακριβώς οι ίδιες. Αυτό μπορεί να ερμηνευτεί ως σταθερότητα στα αποτελέσματα του αλγορίθμου κ-Means, δηλαδή ανεξάρτητα από την αρχικοποίηση ο αλγόριθμος συγκλίνει πάντα σε παρόμοιες ομαδοποιήσεις. Οι υψηλές τιμές του μέσου Rand Index, υποδηλώνουν μεγάλη ομοιότητα της ομαδοποίησης με K-means, με την πραγματική ομαδοποίηση των δεδομένων μας.

## Ε.Επαναλάβατε το (Δ) για cosine μετρική. Ποια μετρική επιτυγχάνει ομαδοποίηση πιο πιστή στις πραγματικές ομάδες των δεδομένων;

Τα αποτελέσματα για 5 τυχαίες αρχικοποιήσεις κέντρων με τη χρήση cosine μετρικής είναι: Mean Rand Index: **0.8616** , Variance of Rand Index: **0.0**.

Οπότε, και η cosine και η squared Euclidean μετρική επιτυγχάνουν ομαδοποίηση αρκετά πιστή στις πραγματικές ομάδες των δεδομένων, αν γνωρίζουμε εξ' αρχής τον αριθμό Κ των κλάσεων. Τα ποσοστά σωστής ομαδοποίησης είναι παρόμοια με την squared Euclidean μετρική να υπερισχύει ελάχιστα.

## Bonus: Έστω ότι σχεδιάζετε ένα σύστημα που χρειάζεται να εκτελεί ταξινόμηση με τη μέθοδο Nearest Neighbor χρησιμοποιώντας ένα πολύ μεγάλο σύνολο δεδομένων αναφοράς (εκπαίδευσης). Μπορείτε να σκεφτείτε έναν τρόπο να μειωθεί το υπολογιστικό κόστος κάθε νέας ταξινόμησης, αξιοποιώντας τεχνικές ομαδοποίησης; Περιγράψτε το σκεπτικό σας και τα βήματα του αλγορίθμου.

Η ταξινόμηση με τη μέθοδο Nearest Neighbor χρησιμοποιώντας ένα πολύ μεγάλο σύνολο δεδομένων αναφοράς εμπεριέχει πολύ μεγάλη υπολογιστική πολυπλοκότητα. Για κάθε νέο δεδομένο που θέλουμε να ταξινομηθεί θα πρέπει να υπολογιστούν οι αποστάσεις από όλα τα δεδομένα του συνόλου δεδομένων αναφοράς. Στη συνέχεια, στον KNN πρέπει να βρεθούν οι κοντινότεροι γείτονες (οι Κ μικρότερες αποστάσεις) και να ανατεθεί το νέο σημείο στην κλάση στην οποία ανήκει η πλειοψηφία των Κ κοντινότερων γειτόνων.

Ένας τρόπος για να μειωθεί το υπολογιστικό κόστος της ταξινόμησης με τη μέθοδο Nearest Neighbor είναι η χρήση τεχνικών ομαδοποίησης. Ένα παράδειγμα είναι το KD δέντρο (K-dimensional tree). Το KD δέντρο είναι ένα δυαδικό δέντρο αναζήτησης που διαχωρίζει τον

χώρο αναζήτησης σε διακριτές ομάδες. Ο αλγόριθμος επιδιώκει να ομαδοποιήσει τα δεδομένα εκπαίδευσης σε ομάδες έτσι ώστε, κατά τη διάρκεια της ταξινόμησης, να περιοριστεί η αναζήτηση στις ομάδες που είναι πιο πιθανό να περιέχουν τα πλησιέστερα δεδομένα και να μη χρειαστεί να υπολογιστούν οι αποστάσεις από όλα τα σημεία του συνόλου εκπαίδευσης.

Αρχικά, κατασκευάζεται το KD δέντρο χρησιμοποιώντας το σύνολο των δεδομένων εκπαίδευσης. Αυτό συμπεριλαμβάνει τον διαχωρισμό του χώρου σε υποσύνολα με βάση τις τιμές των χαρακτηριστικών και την ομαδοποίηση των δεδομένων με βάση τη θέση τους στο χώρο. Κατά τη διάρκεια της ταξινόμησης, το KD δέντρο θα χρησιμοποιηθεί για να περιοριστεί ο υπολογισμός των αποστάσεων, μόνο σε ομάδες που είναι πιθανόν να περιέχουν τα πλησιέστερα δεδομένα. Αυτό μειώνει σημαντικά τον αριθμό των συγκρίσεων και των υπολογισμών που απαιτούνται. Το δέντρο κατασκευάζεται μόνο μία φορά στην αρχή και μετά με βάση αυτό μπορεί να ταξινομηθεί κάθε νέο σημείο.

Ο αλγόριθμος για την κατασκευή του δέντρου θα μπορούσε να ακολουθεί τα παρακάτω βήματα:

1. Επέλεξε τυχαία ένα χαρακτηριστικό
2. Από τις τιμές για το παραπάνω χαρακτηριστικό στο σύνολο δεδομένων, βρες το median.
3. Αν τα σημεία έχουν τιμή για το χαρακτηριστικό που επιλέχθηκε μικρότερη του median τοποθέτησέ τα σε μια ομάδα, αλλιώς σε μια άλλη ομάδα. Χωρισμός δηλαδή με βάση του median σε δυο ομάδες.
4. Αν τα σημεία στις τελικές υποομάδες είναι λιγότερα από κάποιον αριθμό, τερματισμός αλγορίθμου, αλλιώς πήγαινε στο βήμα 5.
5. Για τα σημεία της κάθε υποομάδας επανέλαβε δυο φορές τα βήματα 1,2,3 (μια φορά για την κάθε ομάδα ξεχωριστά).

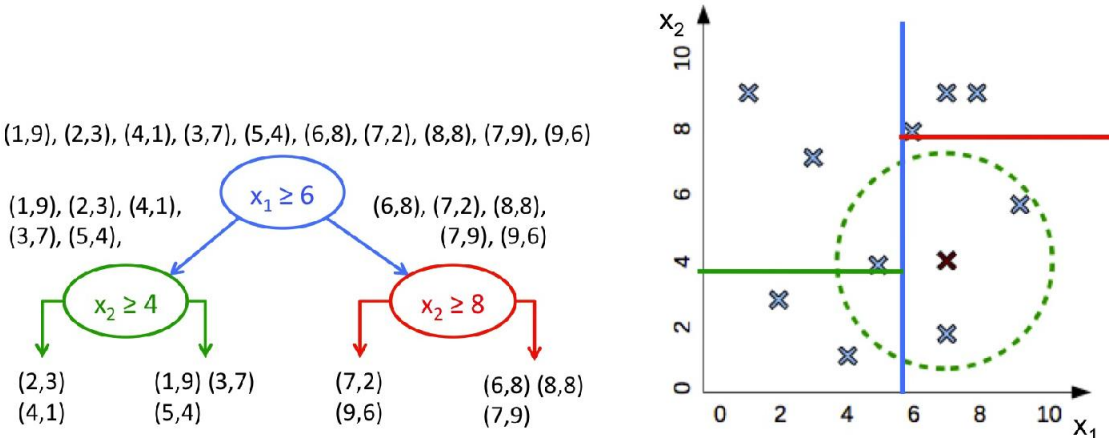
Οπότε, στο τέλος έχουμε το KD δέντρο και τα δεδομένα εκπαίδευσης έχουν ταξινομηθεί κατάλληλα στις νέες ομάδες. Τα φύλλα του δέντρου περιέχουν κάποιο υποσύνολο των δεδομένων εκπαίδευσης.

Ο αλγόριθμος για την ταξινόμηση θα μπορούσε να ακολουθεί τα παρακάτω βήματα:

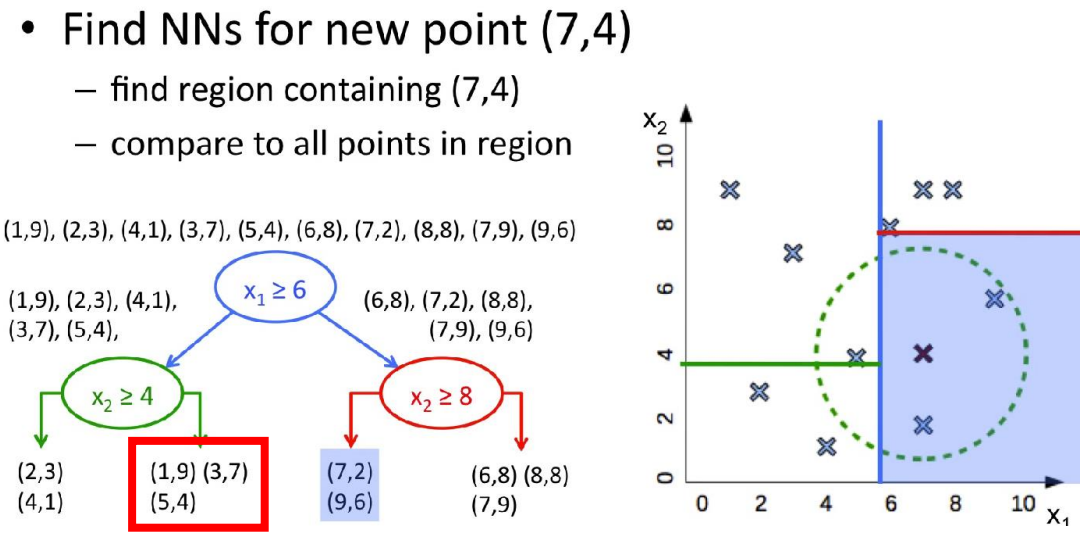
1. Ξεκίνα από τη ρίζα του δέντρου και με βάση την τιμή του σημείου που θέλεις να ταξινομήσεις για το αντίστοιχο χαρακτηριστικό πήγαινε είτε στη μία, είτε στην άλλη ομάδα.
2. Ανάλογα με το κριτήριο ταξινόμησης σε κάθε κόμβο του δέντρου ανέθεσε το νέο σημείο στην αντίστοιχη ομάδα και επανέλαβε το ίδιο για κάθε κόμβο του δέντρου.
3. Όταν φτάσεις σε κάποιο φύλλο, υπολόγισε τις αποστάσεις από όλα τα δεδομένα εκπαίδευσης που βρίσκονται σε αυτό το φύλλο (είναι οι κοντινότεροι γείτονες).
4. Δες σε ποια κλάση από το dataset ανήκει η πλειοψηφία αυτών των σημείων και σε αυτήν την κλάση ανέθεσε το νέο σημείο.

Άρα, αν έχουμε κατασκευάσει το δέντρο, οι αποστάσεις που θα χρειαστεί να υπολογίσουμε είναι σημαντικά λιγότερες. Οπότε, το υπολογιστικό κόστος κάθε νέας ταξινόμησης έχει μειωθεί σημαντικά.

Στην παρακάτω εικόνα υπάρχει ένα παράδειγμα για τη δημιουργία ενός δέντρου και την ταξινόμηση ενός νέου σημείου. Αρχικά, γίνεται διαμερισμός του χώρου με βάση το feature x1 και στη συνέχεια με βάση το x2. Στο τέλος, έχουμε τέσσερα φύλλα (4 ομάδες). Στο διάγραμμα φαίνεται πως έχει διαμεριστεί ο χώρος.




Το νέο δεδομένο θα ακολουθήσει το δεξί κλαδί του δέντρου (x1>=6) και στη συνέχεια το αριστερό (x2<8). Οπότε για την ταξινόμηση, είτε θα λάβουμε υπ' όψιν και τους δυο γείτονες και θα δούμε σε ποια κλάση ανήκει η πλειοψηφία (αν τα σημεία είναι λιγότερα από κ γείτονες), είτε θα υπολογίσουμε αποστάσεις για να βρούμε τους κ κοντινότερους γείτονες και στη συνέχεια θα δούμε σε ποια κλάση ανήκει η πλειοψηφία.




Βέβαια, μπορεί να χάσουμε κάποιον κοντινό γείτονα που είναι κοντά στη γραμμή που διαχωρίζει τις υποομάδες, όπως στο παραπάνω παράδειγμα χάσαμε τον γείτονα που βρίσκεται πάνω στην πράσινη γραμμή. Αυτό το πρόβλημα μπορεί να λυθεί αν ψάχνουμε αναδρομικά το δέντρο όταν η απόσταση από κάποια γραμμή απόφασης είναι μικρότερη από το πιο απομακρυσμένο δεδομένο της περιοχής που βρισκόμαστε. Εδώ η απόσταση από την κόκκινη γραμμή είναι μεγάλη, όμως η απόσταση από την πράσινη γραμμή είναι μικρότερη από την απόσταση από το σημείο (9,6), δηλαδή τον πιο απομακρυσμένο γείτονα από όσους έχουμε βρει. Οπότε, πρέπει να εξετάσουμε και τις αποστάσεις από τα σημεία της δεξιάς πλευράς του πράσινου κόμβου (x2>=4), δηλαδή και την πάνω αριστερά περιοχή το διάγραμμα.

Οπότε, με βάση τον παραπάνω αλγόριθμο συμπεραίνουμε ότι η χρήση τεχνικών ομαδοποίησης, όπως το KD δέντρο, μπορεί να επιταχύνει την ταξινόμηση με τη χρήση του KNN, ειδικά όταν το σύνολο δεδομένων είναι πολύ μεγάλο.

## Άσκηση 2

Θα επιλέξουμε έναν αλγόριθμο ιεραρχικής ομαδοποίησης, agglomerative ή divisive για να τον εφαρμόσουμε στο seed dataset.

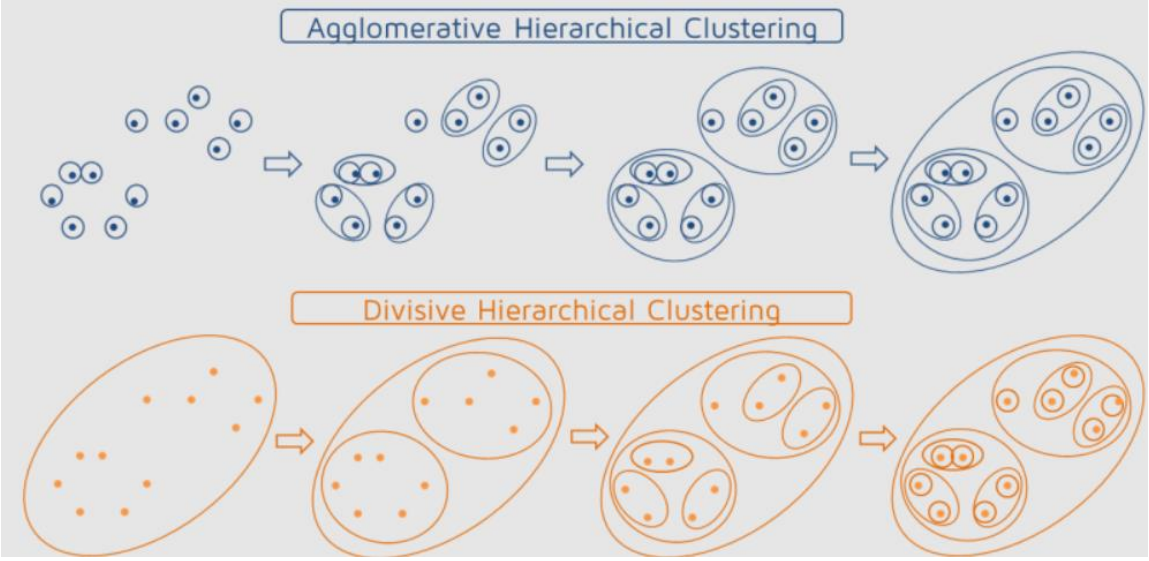
Οι ιεραρχικοί αλγόριθμοι ομαδοποίησης (Hierarchical Clustering) είναι μέθοδοι που χρησιμοποιούνται για την ομαδοποίηση δεδομένων σε ιεραρχικές δομές. Υπάρχουν δύο βασικές κατηγορίες ιεραρχικών αλγορίθμων: συνάθροισης (agglomerative) και διαχωρισμού (divisive).

Οι agglomerative αλγόριθμοι, ξεκινούν με κάθε παρατήρηση να αποτελεί ένα διακριτικό cluster. Στη συνέχεια, ενώνουν επανειλημμένα τα δύο πιο κοντινά clusters μέχρι να σχηματίσουν έναν μεγαλύτερο cluster. Η διαδικασία σταματά όταν όλες οι παρατηρήσεις έχουν ενωθεί σε ένα μοναδικό cluster.

Οι divisive αλγόριθμοι, ξεκινούν με ένα μοναδικό cluster που περιλαμβάνει όλες τις παρατηρήσεις. Στη συνέχεια, διαχωρίζουν επανειλημμένα το μεγαλύτερο cluster σε δύο μικρότερα. Η διαδικασία σταματά όταν κάθε παρατήρηση αποτελεί ένα διακριτικό cluster.

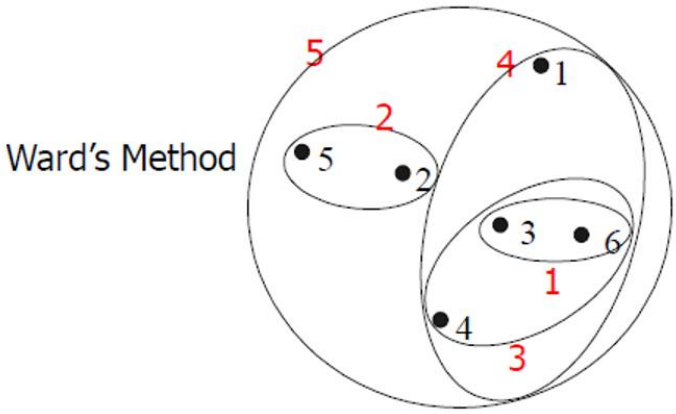
Κατά τη διάρκεια της εκτέλεσης των ιεραρχικών αλγορίθμων, δημιουργείται μια ιεραρχία (δέντρο) που αντιπροσωπεύει τη σειρά με την οποία ομαδοποιούνται ή διαχωρίζονται τα clusters. Τα δέντρα αυτά ονομάζονται δέντρα ομαδοποίησης-δενδρογράμματα (dendrograms).

Οι δύο διαφορετικές κατηγορίες φαίνονται στην παρακάτω εικόνα.



## Α. Περιγράψτε συνοπτικά την τεχνική που επιλέξατε και τις παραμέτρους που χρησιμοποιήσατε (π.χ. μετρική, linkage method κλπ.).

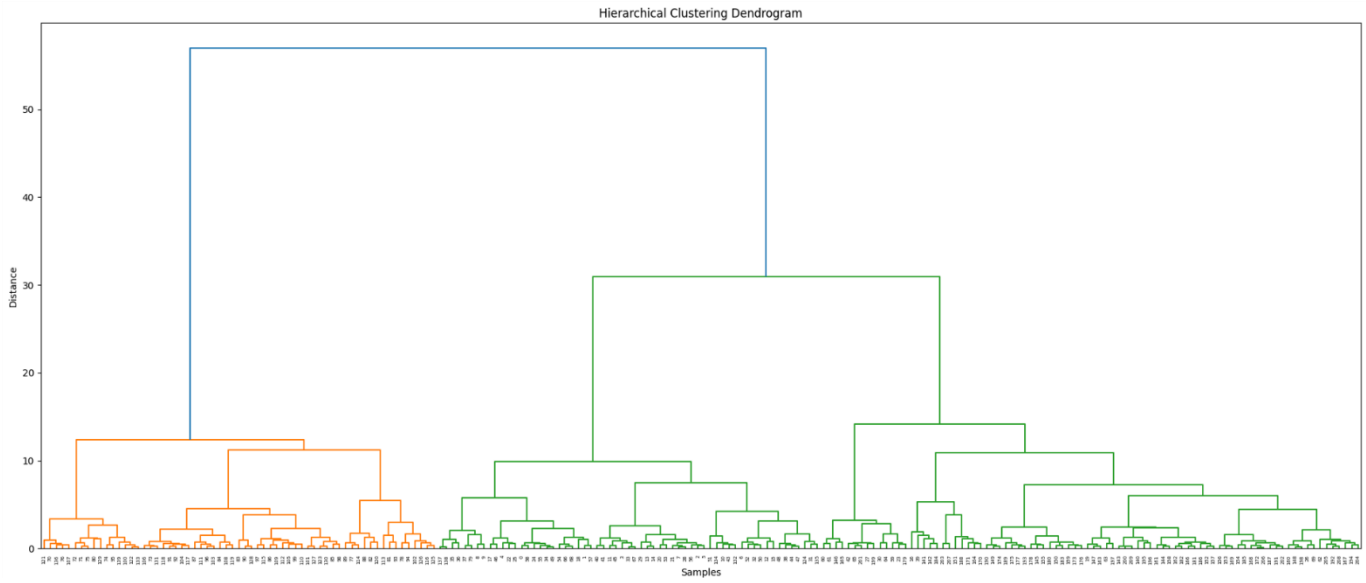
Ο αλγόριθμος ιεραρχικής ομαδοποίησης που επιλέξαμε να χρησιμοποιήσουμε είναι το Agglomerative Clustering. Χρησιμοποιήσαμε τη μέθοδο διασύνδεσης (linkage method) **ward**. Η μέθοδος ward επιδιώκει να ελαχιστοποιήσει τη διακύμανση εντός των συστάδων (clusters) κατά τη συγχώνευσή τους. Συγχωνεύει συστάδες στις οποίες η αύξηση της διακύμανσης είναι η μικρότερη δυνατή. Χρησιμοποιεί τον αλγόριθμο ελαχιστοποίησης της διακύμανσης Ward (Ward variance minimization algorithm). Επίσης, η μετρική που χρησιμοποιήθηκε για τον υπολογισμό των αποστάσεων μεταξύ των παρατηρήσεων είναι η **Ευκλείδεια απόσταση**. Η μέθοδος ward ορίζεται σωστά μόνο όταν χρησιμοποιείται η Ευκλείδεια απόσταση. Εφόσον, χρησιμοποιήθηκε agglomerative αλγόριθμος, πραγματοποιούνται συνεχείς συγχωνεύσεις συστάδων μέχρι να επιτευχθεί ο επιθυμητός αριθμός.



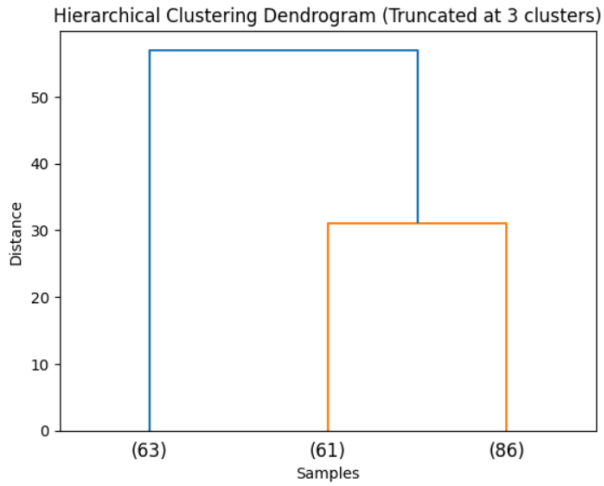

## Β. Κατασκευάστε το δενδρόγραμμα που προέκυψε από την ομαδοποίηση και σχολιάστε πως σχετίζεται με τις ποικιλίες των σιτηρών (κλάσεις) των αντίστοιχων δεδομένων.

Το δενδρόγραμμα (dendrogram) είναι μια γραφική αναπαράσταση της ιεραρχίας των συστάδων που προκύπτουν από τη συσταδοποίηση (clustering). Στο πλαίσιο του agglomerative clustering, το δενδρόγραμμα απεικονίζει τη σειρά των συνεχών συγχωνεύσεων των συστάδων.

Το συνολικό δενδρόγραμμα που που προέκυψε από την ομαδοποίηση φαίνεται στην παρακάτω εικόνα.




Αν θέλουμε μπορούμε να κόψουμε το δενδρόγραμμα στο σημείο όπου δημιουργείται ένας συγκεκριμένος αριθμός συστάδων. Στην παρακάτω εικόνα το δενδρόγραμμα έχει κοπεί στο σημείο όπου σχηματίζονται 3 συστάδες.




Στην άσκηση 1, στο ερώτημα Α είχαμε αναφέρει τα ακόλουθα με βάση τον πίνακα των αποστάσεων:

Όσον αφορά την Ευκλείδεια απόσταση οι κλάσεις που μοιάζουν λιγότερο μεταξύ τους είναι οι 2 με 3. Στη συνέχεια, ακολουθούν οι 1 με 2. Τέλος, οι 1 με τη 3 φαίνεται να μοιάζουν περισσότερο, πάλι όμως υπάρχει κάποια απόσταση. Η κλάση 2 φαίνεται να μπορεί να διαχωριστεί ευκολότερα από τις υπόλοιπες.

Από το δενδρόγραμμα παρατηρούμε ότι αρχικά διαχωρίζεται η μία κλάση από τις υπόλοιπες. Αυτή είναι η κλάση που διαφέρει περισσότερο, οπότε στην αρχή είναι πιο εύκολο να διαχωριστεί, δηλαδή είναι η κλάση 2. Εφόσον οι 1 και 3 μοιάζουν περισσότερο, αρχικά θα δημιουργήσουν ένα ενιαίο cluster. Στη συνέχεια, διαχωρίζονται οι 1 και 3 και καταλήγουμε σε 3 clusters.

## Γ. Χρησιμοποιήστε την ιεραρχική ομαδοποίηση που δημιουργήσατε ώστε να διαχωρίσετε τα δεδομένα σε 3 ομάδες. Χρησιμοποιήστε κάποιο κριτήριο εξωτερικής επικύρωσης (π.χ. rand index, adjusted rand index, mutual information κλπ) και συγκρίνετε την ομαδοποίηση αυτή με την αντίστοιχη που επιτυγχάνει ο k-means σε σχέση με τις ετικέτες των δεδομένων. Ποια μετρική επιτυγχάνει ομαδοποίηση πιο πιστή στις πραγματικές ομάδες των δεδομένων;

Σαν κριτήριο θα χρησιμοποιήσουμε το Rand Index. Από την άσκηση 1, ερώτημα Δ με τη χρήση k-means και squared Euclidean μετρικής έχουμε, Rand Index: **0.8744**

Στην ιεραρχική ομαδοποίηση με agglomerative τεχνική έχουμε, Rand Index: **0.8723**

Οπότε, ο k-means με squared Euclidean μετρική και το agglomerative clustering επιτυγχάνουν ομαδοποίηση αρκετά πιστή στις πραγματικές ομάδες των δεδομένων, αν γνωρίζουμε εξ' αρχής τον αριθμό Κ των κλάσεων. Τα ποσοστά σωστής ομαδοποίησης είναι παρόμοια και στους δύο αλγορίθμους.

## Δ. Σχολιάστε τα πλεονεκτήματα των ιεραρχικών τεχνικών ομαδοποίησης.

Τα πλεονεκτήματα των ιεραρχικών τεχνικών ομαδοποίησης είναι τα ακόλουθα:

- Η ιεραρχική δομή που παράγεται (δενδρόγραμμα) είναι ευκολότερη στην κατανόηση. Οι ομάδες οργανώνονται σε επίπεδα, καθιστώντας πιο εύκολη την αναγνώριση παρόμοιων χαρακτηριστικών.
- Παρέχουν λεπτομερείς πληροφορίες σχετικά με το πόσο όμοιες είναι μεταξύ τους οι παρατηρήσεις. Αντίθετα με πολλούς άλλους αλγορίθμους που επιστρέφουν μόνο τον αριθμό της ομάδας μιας παρατήρησης, η ιεραρχική ομαδοποίηση επιτρέπει μια πιο λεπτομερή κατανόηση των ομοιοτήτων (δενδρόγραμμα).
- Δεν επηρεάζονται από αρχικές συνθήκες, όπως οι τιμές random seed ή η σειρά των δεδομένων. Η επανεκτέλεση της ανάλυσης με διάφορες αρχικές συνθήκες παράγει τα ίδια αποτελέσματα.
- Είναι πιο ανθεκτικές (robust) σε σύγκριση με άλλες μεθόδους, διότι δεν απαιτούν τον προκαθορισμό του αριθμού των ομάδων ή άλλων παραμέτρων.
- Είναι κατάλληλες για περιπτώσεις όπου δεν υπάρχει προηγούμενη γνώση της δομής των δεδομένων.
- Δεν κάνουν τόσο αυστηρές υποθέσεις για το σχήμα των ομάδων.
- Είναι κλιμακούμενες (scalable) μέθοδοι που μπορεί να διαχειριστούν αποτελεσματικά μεγάλα σύνολα δεδομένων.
- Είναι λιγότερο ευαίσθητες στο θόρυβο ή σε ακραία σημεία (outliers) σε σύγκριση με άλλους αλγορίθμους ομαδοποίησης. Αυτό οφείλεται στο ότι τα ακραία σημεία συνήθως δεν εντάσσονται σε μια ομάδα μέχρι το τέλος της διαδικασίας, όταν έχουν ήδη επεξεργαστεί όλες οι άλλες παρατηρήσεις (τουλάχιστον για agglomerative τεχνικές).

Παρόλα αυτά, πρέπει να σημειωθεί ότι οι ιεραρχικές τεχνικές ομαδοποίησης δεν είναι πάντα η καλύτερη επιλογή. Η απόδοση τους εξαρτάται σε μεγάλο βαθμό από τη φύση των δεδομένων και τους στόχους της ανάλυσης.

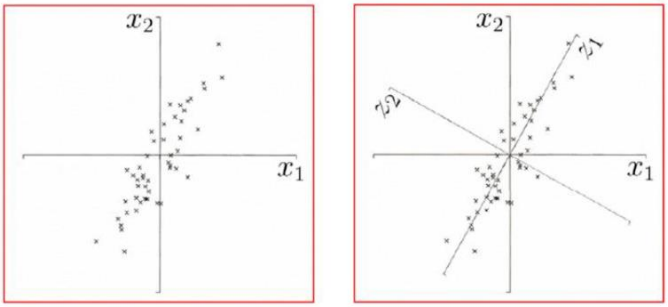
# Άσκηση 3

Θα χρησιμοποιήσουμε το seed dataset για να εφαρμόσουμε διάφορες τεχνικές μείωσης διαστάσεων (Dimensionality Reduction) και να μελετήσουμε τη σημαντικότητα των χαρακτηριστικών.

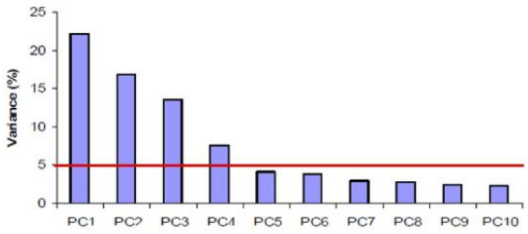
## Α. Να εφαρμοστεί η μέθοδος PCA στα δεδομένα. Ποιες είναι οι ελάχιστες κύριες συνιστώσες που πρέπει να κρατήσετε ώστε να εξηγείται τουλάχιστον το 90% της variance του αρχικού dataset στη νέα απεικόνιση και πόσες για το 99% αυτής;

Η Principal Component Analysis (PCA) είναι μια γραμμική τεχνική μείωσης διαστάσεων που χρησιμοποιείται σε προβλήματα μη επιβλεπόμενης μάθησης. Η PCA χρησιμοποιείται για τη μείωση της διάστασης των δεδομένων, ενώ παράλληλα διατηρεί την πληροφορία που είναι πιο σημαντική για το πρόβλημα. Η βασική ιδέα είναι η ανακάλυψη των κύριων συνιστωσών (principal components), που είναι γραμμικοί συνδυασμοί των αρχικών μεταβλητών και αντιστοιχούν σε μέγιστη διακύμανση (variance). Οι επιλεγμένες κυρίαρχες συνιστώσες χρησιμοποιούνται για την αναπαράσταση των αρχικών δεδομένων σε ένα χώρο χαμηλότερης διάστασης.

Στο παρακάτω παράδειγμα το πρώτο principal component είναι αυτό που μεγιστοποιεί τη διακύμανση και το δεύτερο είναι κάθετο (ορθογώνια) στο πρώτο. Σε αυτό το παράδειγμα δεν αλλάζει η διάσταση των δεδομένων, δημιουργείται όμως μια νέα αναπαράσταση που μπορεί να φανεί χρήσιμη στο πρόβλημα που θέλουμε να επιλύσουμε.




Καθώς οι διαστάσεις μειώνονται χάνεται και πληροφορία. Επιλέγουμε να κρατήσουμε τόσες διαστάσεις (τόσα principal components), έτσι ώστε να εξηγείται μεγάλο ποσοστό της αρχικής διακύμανσης (επιλέγουμε τα ιδιοδιανύσματα του πίνακα συνδιασποράς με τις μεγαλύτερες ιδιοτιμές), ανάλογα με το πρόβλημά μας. Σε κάποιες περιπτώσεις η μείωση της διάστασης δεν οδηγεί σε μείωση της διακύμανσης, δηλαδή η πληροφορία που χάνεται είναι αμελητέα (redundant features, οι ιδιοτιμές είναι μικρές). Οπότε, σε πολλές περιπτώσεις μας συμφέρει να συμπιεστούν τα χαρακτηριστικά σε χαμηλότερη διάσταση.




Ανάλογα με το πρόβλημα συνήθως επιλέγεται η διατήρηση του 90% ή του 99% (πιο αυστηρό) της αρχικής διακύμανσης.

Αρχικά, εφαρμόζουμε PCA στο seed dataset. Όλα τα χαρακτηριστικά τα κρατάμε και υπολογίζουμε τόσα principal components.

Η διακύμανση που εξηγείται από κάθε χαρακτηριστικό είναι:

```
Explained Variance Ratio is:
[8.29385197e-01 1.63632452e-01 5.65790880e-03 9.90306086e-04
 2.11180347e-04 1.20677139e-04 2.27879552e-06]
```

Ο πίνακας προβολής (πίνακας με principal components) είναι:

```
[[ 0.884  0.395  0.004  0.129  0.111 -0.128  0.129]
 [ 0.101  0.056 -0.003  0.031  0.002  0.989  0.082]
 [ 0.265 -0.283  0.059 -0.4    0.319  0.064 -0.762]
 [-0.199  0.579 -0.058  0.436 -0.234  0.025 -0.613]
 [-0.137  0.575 -0.053 -0.787 -0.145 -0.002  0.088]
 [ 0.281 -0.302 -0.045 -0.113 -0.896  0.003 -0.11 ]
 [ 0.025 -0.066 -0.994 -0.001  0.082 -0.001 -0.009]]
```

Κάθε γραμμή του πίνακα αναπαριστά ένα principal component και κάθε στήλη ένα χαρακτηριστικό. Για την πρώτη κύρια συνιστώσα (πρώτη γραμμή), η μέγιστη τιμή (απόλυτη τιμή) είναι 0.884 που βρίσκεται στην πρώτη στήλη. Άρα, για το πρώτο principal component το χαρακτηριστικό που συνεισφέρει περισσότερο είναι το πρώτο. Αντίστοιχα, το 6ο χαρακτηριστικό συνεισφέρει περισσότερο στο 2ο principal component (βάρος 0.989).

Η αθροιστική διακύμανση αθροίζει για κάθε principal component τη διακύμανση που εξηγεί το ίδιο και τη διακύμανση που εξηγούν όλα τα προηγούμενα. Στην ουσία μας δείχνει, αν κρατήσουμε N principal components, ποια θα είναι η συνολική διακύμανση που θα εξηγείται από αυτά.

```
Cumulative Explained Variance Ratio is:
[0.8293852  0.99301765 0.99867556 0.99966586 0.99987704 0.99999772
 1.        ]
```

Παρατηρούμε ότι το πρώτο principal component εξηγεί περίπου το 83% της διακύμανσης, ενώ το πρώτο και το δεύτερο εξηγούν το 99%. Άρα, οι ελάχιστες κύριες συνιστώσες που πρέπει να κρατήσουμε ώστε να εξηγείται τουλάχιστον το 90% και το 99% της διακύμανσης του αρχικού dataset στη νέα απεικόνιση είναι οι πρώτες δύο κύριες συνιστώσες.

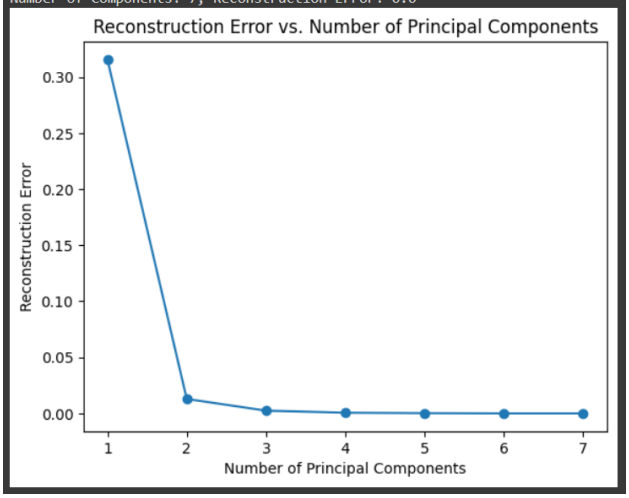
```
To explain 90% of the variance we need at least 2 principal components.
To explain 99% of the variance we need at least 2 principal components.
```

## Β. Να υπολογισθεί το σφάλμα ανακατασκευής των δεδομένων χρησιμοποιώντας από 1 έως 7 κύριες συνιστώσες, και να αποτυπωθεί σε κατάλληλο διάγραμμα.

Χρησιμοποιώντας τον πίνακα προβολής από μια αναπαράσταση χαμηλότερης διάστασης μπορούμε να ανακατασκευάσουμε την αρχική αναπαράσταση (reconstruction). Αν δεν έχει χαθεί μεγάλο μέρος πολύτιμης πληροφορίας (διατήρηση των σημαντικότερων principal components) το σφάλμα ανακατασκευής είναι σχετικά μικρό.

Παρατηρούμε ότι το σφάλμα για την ανακατασκευή με δύο κύριες συνιστώσες είναι σχετικά μικρό (~1%). Ενώ, για την ανακατασκευή με μία κύρια συνιστώσα το σφάλμα είναι αρκετά υψηλό (~31%). Οπότε, καταλαβαίνουμε ότι οι δύο πρώτες κύριες συνιστώσες περιέχουν το 99% της πληροφορίας και δε θα έχουμε κάποια σημαντική απώλεια αν οι διαστάσεις μειωθούν από 7 σε 2. Προφανώς όσο περισσότερες κύριες συνιστώσες χρησιμοποιούμε, τόσο το σφάλμα ανακατασκευής τείνει να μηδενιστεί.

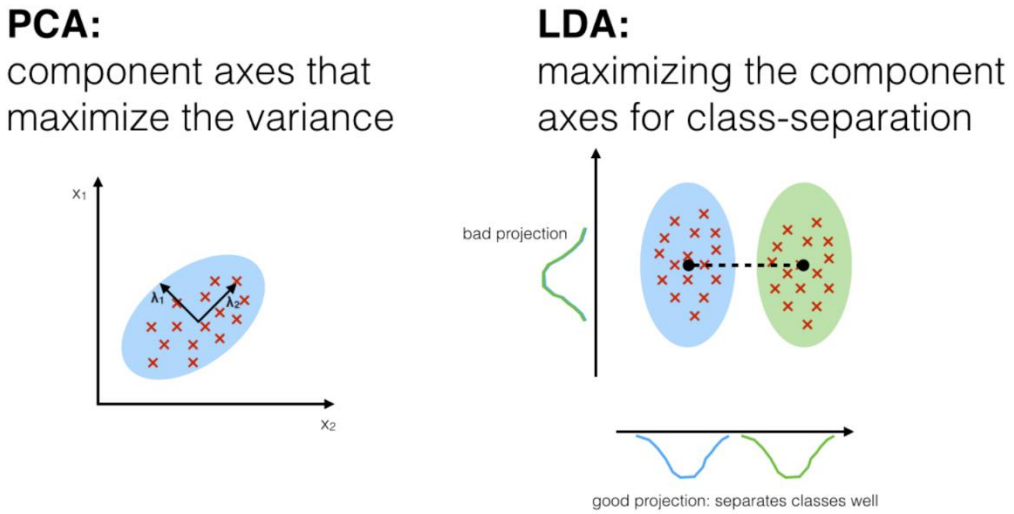
```
Number of Components: 1, Reconstruction Error: 0.3157
Number of Components: 2, Reconstruction Error: 0.0129
Number of Components: 3, Reconstruction Error: 0.0025
Number of Components: 4, Reconstruction Error: 0.0006
Number of Components: 5, Reconstruction Error: 0.0002
Number of Components: 6, Reconstruction Error: 0.0
Number of Components: 7, Reconstruction Error: 0.0
```



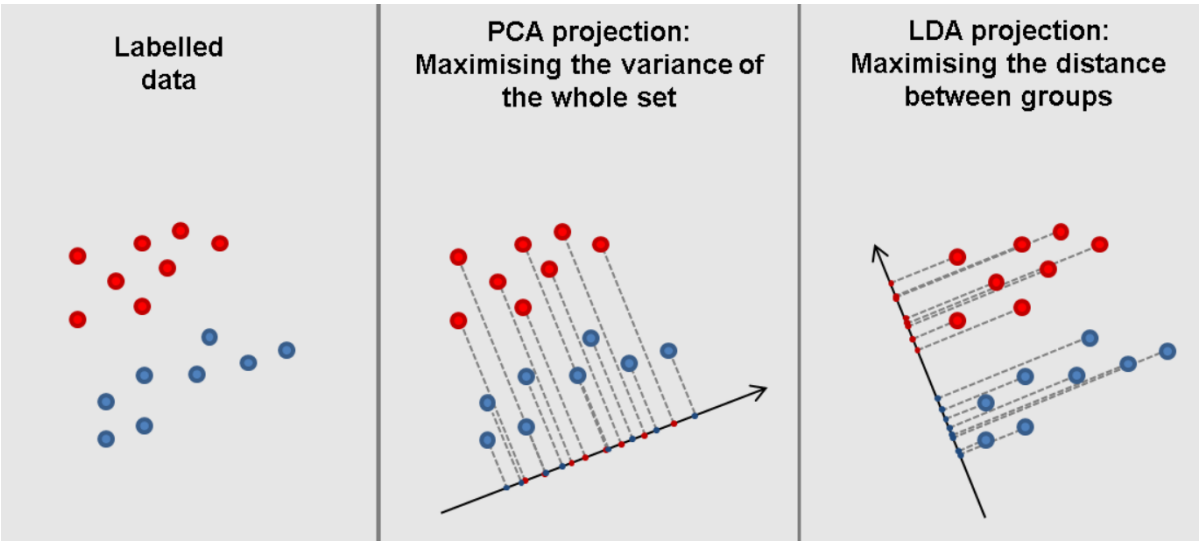

## Γ. Να εφαρμοστεί η μέθοδος LDA για την απεικόνιση του dataset σε 2 διαστάσεις. Να συγκρίνετε την απεικόνιση αυτή με την αντίστοιχη που παράγεται από τη μέθοδο PCA. Ποια τα κυριότερα ποιοτικά χαρακτηριστικά των απεικονίσεων που παράγουν οι δύο μέθοδοι και σε τι οφείλονται; Εξηγήστε.

H Linear Discriminant Analysis (LDA) είναι μια γραμμική τεχνική μείωσης διαστάσεων που χρησιμοποιείται σε προβλήματα επιβλεπόμενης μάθησης. Ο κύριος σκοπός της LDA είναι να βρει τις γραμμικές προβολές των δεδομένων που μεγιστοποιούν τον διαχωρισμό μεταξύ των διαφορετικών κλάσεων. Στον τελικό χώρο που προκύπτει με την LDA, σκοπός είναι να μεγιστοποιηθεί η απόσταση μεταξύ των μέσων των κλάσεων και να ελαχιστοποιηθεί η διακύμανση εντός των κλάσεων. Αυτό οδηγεί σε μία προβολή των δεδομένων όπου οι κλάσεις είναι όσο το δυνατόν πιο κοντά μεταξύ τους και εσωτερικά, ενώ οι διαφορετικές κλάσεις είναι όσο το δυνατόν πιο απομακρυσμένες.

Μια σύγκριση μεταξύ της PCA και της LDA μπορούμε να δούμε στην παρακάτω εικόνα:



Επίσης, στο επόμενο παράδειγμα φαίνεται ότι η μείωση των διαστάσεων με την PCA από 2 σε 1 δεν μας οδηγεί σε καλό αποτέλεσμα, ενώ με την LDA οδηγούμαστε σε διαχωρίσιμες κλάσεις. Η κάθε μέθοδος μπορεί να λειτουργήσει με διαφορετικό τρόπο ανάλογα με τα δεδομένα.

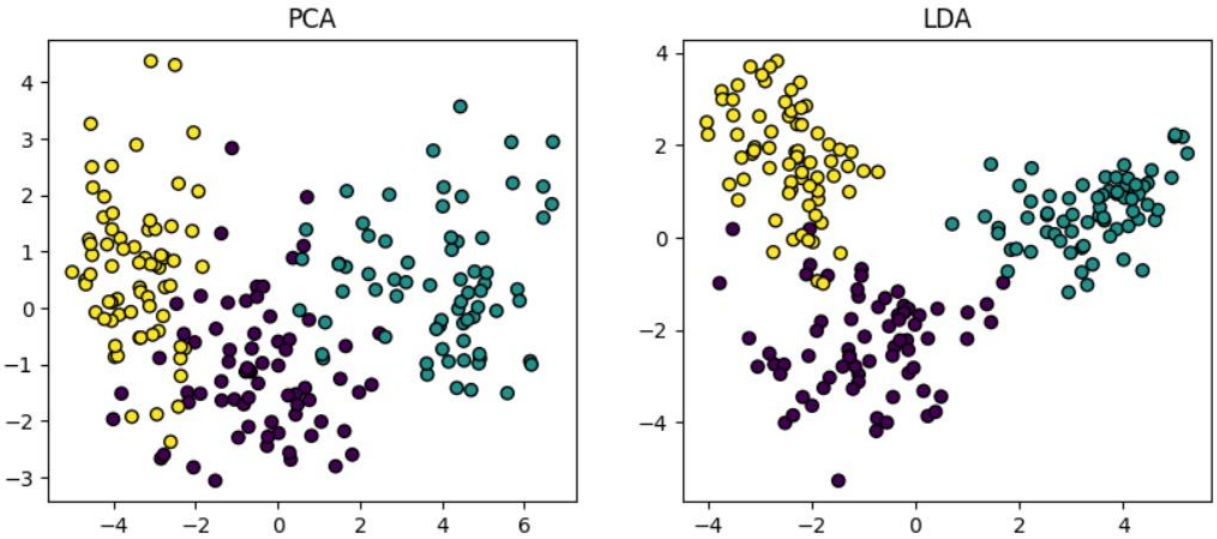



Οι πίνακες προβολής για την PCA και την LDA για μείωση των διαστάσεων από 7 σε 2 είναι:

```
PCA Projection Matrix:
[[ 0.8842285   0.39540542  0.00431132  0.12854448  0.11105914 -0.12761562
   0.1289665 ]
 [ 0.10080577  0.05648963 -0.00289474  0.03062173  0.00237229  0.98941048
   0.08223339]]

LDA Projection Matrix:
[[-4.23778614e-01  3.79919995e+00  5.92772810e+00 -5.98819597e+00
   3.70482197e-02 -4.50472160e-02  3.11807592e+00]
 [ 4.19531669e+00 -8.50579585e+00 -8.69823024e+01 -7.83067468e+00
   7.14104253e-01  3.21253808e-01  6.91384931e+00]]
```

Στην LDA, ο πίνακας προβολής (περιέχει τους συντελεστές κλιμάκωσης για κάθε χαρακτηριστικό στον χώρο γραμμικής διάκρισης), μας δίνεται ανεστραμένος στη βιβλιοθήκη sklearn, οπότε τον αναστρέφουμε για να τον βλέπουμε στην ίδια μορφή που έχουμε και τον πίνακα προβολής από την PCA.

Η απεικόνιση του dataset με τη χρήση των δύο μεθόδων σε δύο διαστάσεις φαίνεται στην παρακάτω εικόνα. Στην απεικόνιση της PCA οι κλάσεις απλώνονται αρκετά (μεγάλη διακύμανση κάθε κλάσης) και σε κάποια σημεία η επικάλυψη μεταξύ τους είναι αρκετά μεγάλη. Στην απεικόνιση της LDA οι κλάσεις είναι πιο συγκεντρωμένες (μικρή διακύμανση κάθε κλάσης) και δεν υπάρχει τόσο μεγάλη επικάλυψη μεταξύ των κλάσεων (μόνο η κίτρινη με τη μωβ κλάση επικαλύπτονται λίγο, ενώ στην PCA η μωβ επικαλύπτεται αρκετά και με τις άλλες δύο κλάσεις). Οπότε, αν έπρεπε να επιλέξουμε μεταξύ των δύο μεθόδων στο συγκεκριμένο dataset θα επιλέγαμε να χρησιμοποιησουμε την LDA, καθώς επιτυγχάνει μεγαλύτερη διαχωρισιμότητα των κλάσεων.




Όπως αναφέραμε και παραπάνω η PCA προσπαθεί να διατηρήσει τη διακύμανση των δεδομένων, ώστε στη νέα απεικόνιση τα δεδομένα να είναι όσο πιο απλωμένα γίνεται και να μην πέσουν τα σημεία το ένα πάνω στο άλλο. Η LDA δίνει μεγαλύτερη σημασία στη διαχωρισιμότητα των κλάσεων, ώστε στη νέα απεικόνιση τα σημεία να είναι συγκεντρωμένα γύρω από το κέντρο της κλάσης τους και τα κέντρα να είναι όσο πιο απομακρυσμένα γίνεται. Η LDA χρησιμοποιεί τις ετικέτες των κλάσεων, οπότε είναι λογικό να δίνει καλύτερο αποτέλεσμα από την PCA, δεν μπορεί όμως να λειτουργήσει σε προβλήματα μη επιβλεπόμενης μάθησης. Η PCA χειρίζεται όλα τα προβλήματα σαν προβλήματα μη επιβλεπόμενης μάθησης (δεν χρησιμοποιεί τις ετικέτες ακόμα και αν υπάρχουν).

## Δ. Με βάση τον πίνακα προβολής που παράγεται από την LDA στο προηγούμενο ερώτημα, ποια είναι τα δύο χαρακτηριστικά (features) που συνεισφέρουν περισσότερο στη διάκριση μεταξύ των κλάσεων και ποια τα δύο που συνεισφέρουν λιγότερο απ' όλα; Δημιουργείστε δυο δυσδιάστατες απεικονίσεις των δεομένων χρησιμοποιώντας το καθένα από τα δύο ζεύγη χαρακτηριστικών που καταδείξατε. Σχολιάστε.

Ξαναβλέπουμε τους πίνακες προβολής των δύο μεθόδων.

```
PCA Projection Matrix:
[[ 0.8842285   0.39540542  0.00431132  0.12854448  0.11105914 -0.12761562
   0.1289665 ]
 [ 0.10080577  0.05648963 -0.00289474  0.03062173  0.00237229  0.98941048
   0.08223339]]

LDA Projection Matrix:
[[-4.23778614e-01  3.79919995e+00  5.92772810e+00 -5.98819597e+00
   3.70482197e-02 -4.50472160e-02  3.11807592e+00]
 [ 4.19531669e+00 -8.50579585e+00 -8.69823024e+01 -7.83067468e+00
   7.14104253e-01  3.21253808e-01  6.91384931e+00]]
```

Στην PCA όπως είχαμε αναφέρει και στο ερώτημα Α, κάθε γραμμή του πίνακα αποτελεί ένα principal component. Για το 1$^{ο}$ principal component το χαρακτηριστικό που συνεισφέρει

περισσότερο είναι το 1ο, ενώ για το 2ο principal component το χαρακτηριστικό που συνεισφέρει περισσότερο είναι το 6ο.

```
Maximum value in row 1 is: 0.884
Most significant feature for principal component 1 is: feature 1

Maximum value in row 2 is: 0.989
Most significant feature for principal component 2 is: feature 6
```

Στον πίνακα προβολής LDA κάθε γραμμή του πίνακα αναφέρεται σε κάποιο άξονα διάκρισης και οι τιμές αναφέρονται στα βάρη του κάθε χαρακτηριστικού σε κάθε άξονα διάκρισης.

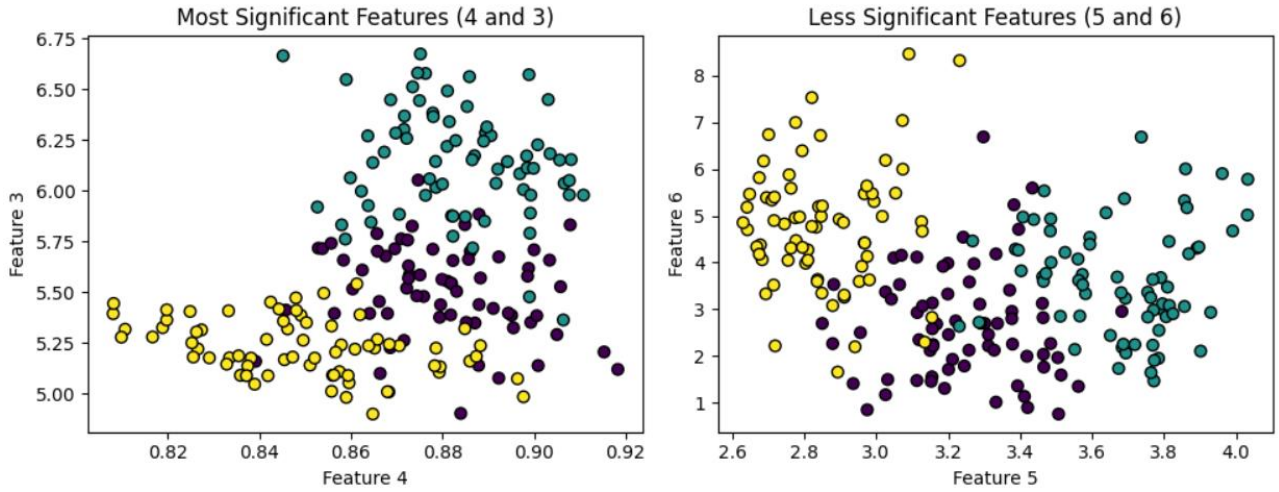
Στην LDA τα χαρακτηριστικά που συνεισφέρουν περισσότερο στη διάκριση μεταξύ των κλάσεων είναι τα 3 και 4 (τα χαρακτηριστικά με τα μεγαλύτερα βάρη κατά απόλυτη τιμή).

```
Maximum value in row 1 is: 5.988
Most significant feature for class discrimination is: feature 4

Maximum value in row 2 is: 86.982
Most significant feature for class discrimination is: feature 3
```

Τα χαρακτηριστικά που συνεισφέρουν λιγότερο στη διάκριση μεταξύ των κλάσεων είναι τα 5 και 6 (τα χαρακτηριστικά με τα μικρότερα βάρη κατά απόλυτη τιμή).

```
Minimum value in row 1 is: 0.037
Less significant feature for class discrimination is: feature 5

Minimum value in row 2 is: 0.321
Less significant feature for class discrimination is: feature 6
```

Στην παρακάτω εικόνα φαίνονται οι δύο δισδιάστατες απεικονίσεις χρησιμοποιώντας τα δύο χαρακτηριστικά που συνεισφέρουν περισσότερο και τα δύο χαρακτηριστικά που συνεισφέρουν λιγότερο στη διάκριση μεταξύ των κλάσεων.




Παρατηρούμε ότι χρησιμοποιώντας τα χαρακτηριστικά που συνεισφέρουν περισσότερο στη διάκριση μεταξύ των κλάσεων, αν και υπάρχει αρκετά μεγάλη επικάλυψη των κλάσεων, μπορούμε να διαχωρίσουμε τις τρείς κλάσεις (μπορούμε να διακρίνουμε τις περιοχές της κάθε κλάσης). Χρησιμοποιώντας τα χαρακτηριστικά που συνεισφέρουν λιγότερο στη διάκριση μεταξύ των κλάσεων, η επικάλυψη είναι αρκετά μεγαλύτερη και τα σημεία από

διαφορετικές κλάσεις σε αρκετές περιοχές πέφτουν το ένα πάνω στο άλλο. Επίσης, περίπου στο κέντρο και προς τα κάτω στο διάγραμμα έχουμε σημεία και από τις τρείς κλάσεις (φαίνεται η κίτρινη και η μπλε κλάση να πέφτουν πάνω στη μωβ), ενώ στο πρώτο διάγραμμα σημεία μόνο δύο κλάσεων επικαλύπτονται μεταξύ τους σε κάθε περιοχή (η κίτρινη και η μπλε κλάση έχουν αρκετή απόσταση μεταξύ τους και υπάρχει κενός χώρος ενδιάμεσα για τη μωβ κλάση). Οπότε, χρησιμοποιώντας τα χαρακτηριστικά που συνεισφέρουν περισσότερο στη διάκριση μεταξύ των κλάσεων, επιτυγχάνουμε καλύτερη διαχωρισιμότητα.

# Άσκηση 4

Θα εφαρμόσουμε την τεχνική Multi-Dimensional Scaling (MDS) στον πίνακα αποστάσεων Distance Matrix world.

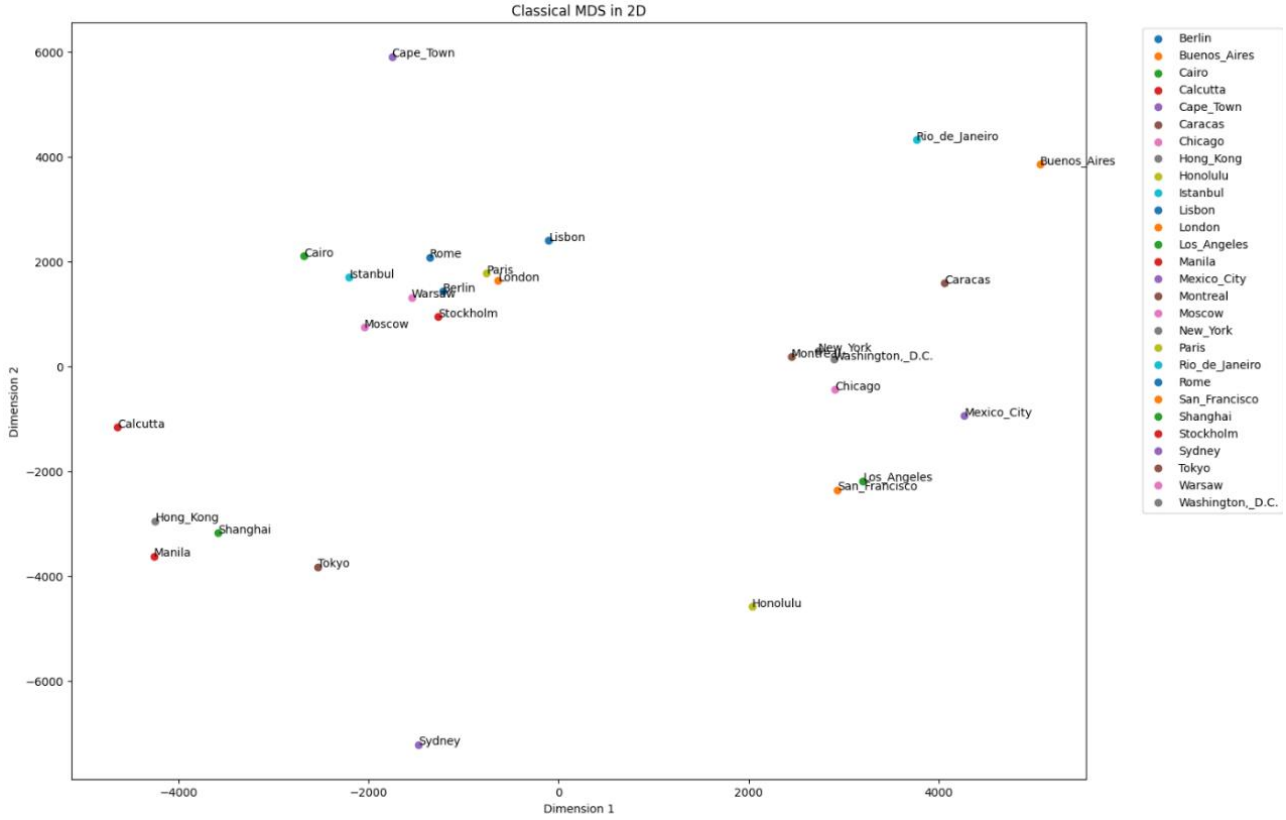
## Α. Χρησιμοποιείστε τη μέθοδο classical MDS για να δημιουργήσετε μία διανυσματική αναπαράσταση των πόλεων του κόσμου (Distance_Matrix_world) στις δύο και στις τρεις διαστάσεις. Απεικονίστε τις αναπαραστάσεις αυτές σε κατάλληλο διάγραμμα και σχολιάστε το αποτέλεσμα.

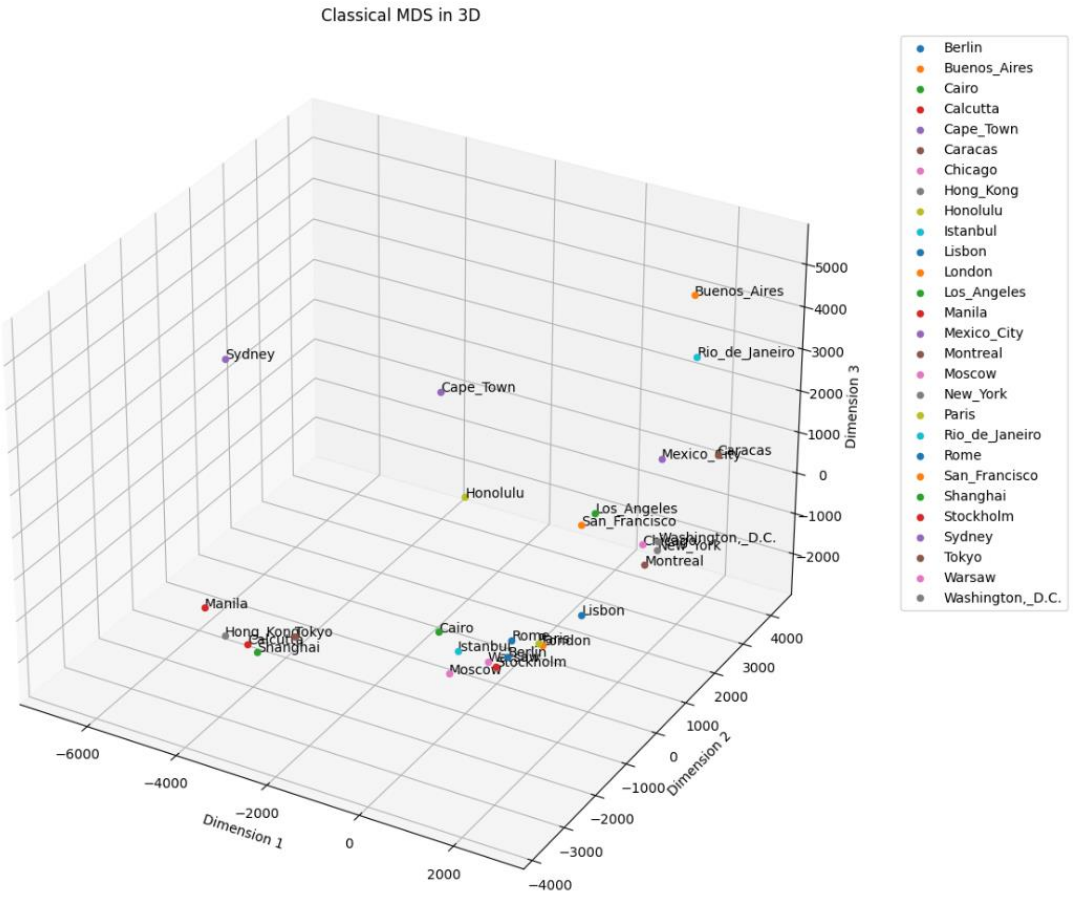
Η τεχνική Multi-Dimensional Scaling (MDS) είναι μια τεχνική ανάλυσης δεδομένων που χρησιμοποιείται για να αναπαριστήσει γραφικά τις αποστάσεις μεταξύ αντικειμένων ή δειγμάτων σε έναν χώρο χαμηλότερης διάστασης. Είναι μια μορφή μη γραμμικής μείωσης διαστάσεων και μπορεί να χρησιμοποιηθεί για την εξερεύνηση ομοιοτήτων ή διαφορών στα δεδομένα. Η βασική ιδέα είναι να μετατρέψει τις αρχικές αποστάσεις μεταξύ των δειγμάτων σε νέες αποστάσεις σε ένα χώρο χαμηλότερης διάστασης, προκειμένου να γίνει πιο εύκολη η οπτικοποίηση. Είναι μια τεχνική μη γραμμικής μείωσης διαστάσεων και μπορεί να χρησιμοποιηθεί για την εξερεύνηση ομοιοτήτων ή διαφορών στα δεδομένα.

Στην ουσία ο MDS προσπαθεί να αντιστοιχίσει τα στοιχεία σε ένα χώρο χαμηλότερης διάστασης (συνήθως 2D για οπτική αναπαράσταση ή μερικές φορές 3D) με τέτοιο τρόπο ώστε τα στοιχεία που ήταν κοντά μεταξύ τους στον αρχικό χώρο υψηλής διάστασης να είναι κοντά μεταξύ τους στο χώρο χαμηλής διάστασης. Τα αντικείμενα που απείχαν αρκετά στο χώρο των υψηλών διαστάσεων απέχουν πολύ και στο χώρο των χαμηλών διαστάσεων.

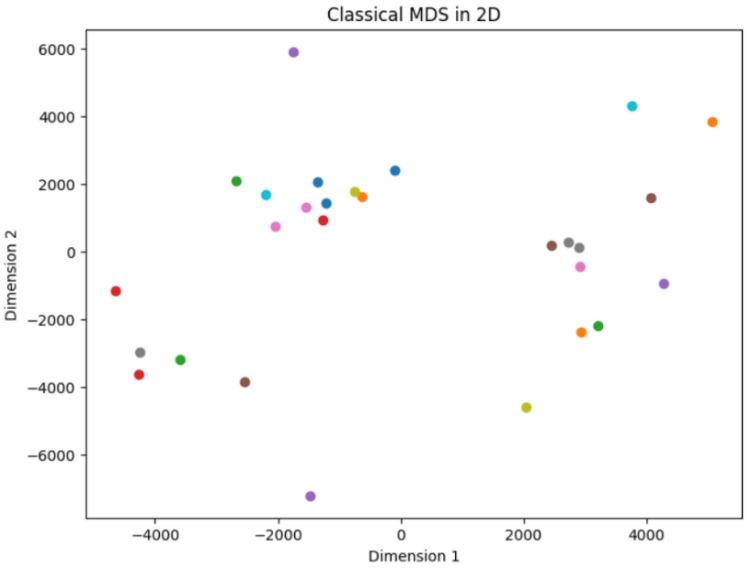
Οι αναπαραστάσεις για δύο και τρείς διαστάσεις φαίνονται στα παρακάτω διαγράμματα.

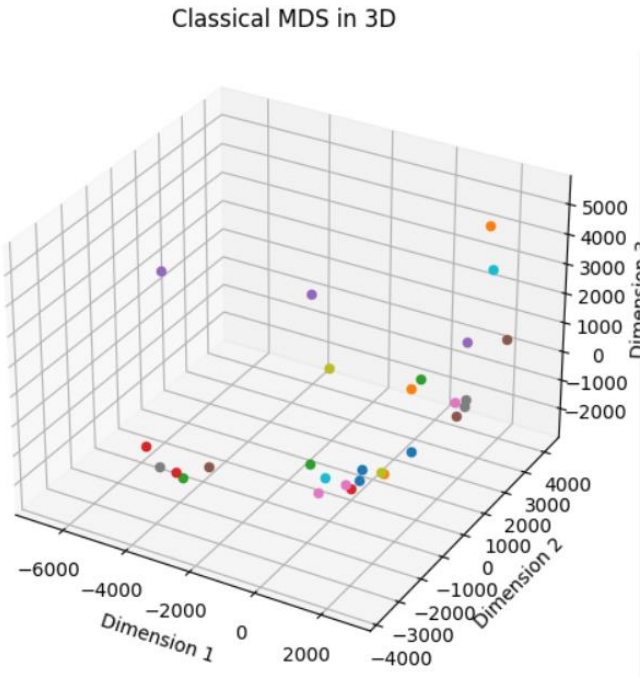
Παρατηρούμε ότι οι πόλεις που απέχουν λίγο στο δισδιάστατο διάγραμμα, απέχουν λίγο και στο τρισδιάστατο (πχ. Buenos_Aires με Rio_de_Janeiro ή San_Francisco με Los_Angeles). Αντίθετα οι απομακρυσμένες πόλεις στο δισδιάστατο διάγραμμα είναι απομακρυσμένες και στο τρισδιάστατο (πχ. Manila με Mexico_City). Οι αρχικές αποστάσεις διατηρούνται και σε χαμηλότερες διαστάσεις. Βέβαια υπάρχουν και κάποιες εξαιρέσεις (πχ. Sydney με Cape_Town) που σε 2D φαίνεται να απέχουν πάρα πολύ μεγάλη απόσταση, ενώ σε 3D απέχουν πάλι κάποια απόσταση αλλά όχι τόσο μεγάλη.


Classical MDS in 2D
Dimension 1
Dimension 2
Cape_Town
Rio_de_Janeiro
Buenos_Aires
Lisbon
Cairo
Rome
Istanbul
Paris
London
Berlin
Warsaw
Stockholm
Moscow
Caracas
New_York
Montreal
Washington,_D.C.
Chicago
Mexico_City
Calcutta
Los_Angeles
San_Francisco
Hong_Kong
Shanghai
Manila
Tokyo
Honolulu
Sydney
Berlin
Buenos_Aires
Cairo
Calcutta
Cape_Town
Caracas
Chicago
Hong_Kong
Honolulu
Istanbul
Lisbon
London
Los_Angeles
Manila
Mexico_City
Montreal
Moscow
New_York
Paris
Rio_de_Janeiro
Rome
San_Francisco
Shanghai
Stockholm
Sydney
Tokyo
Warsaw
Washington,_D.C.


Classical MDS in 3D
Dimension 1
Dimension 2
Dimension 3
Buenos_Aires
Sydney
Rio_de_Janeiro
Cape_Town
Caracas
Mexico_City
Honolulu
Los_Angeles
San_Francisco
Washington,_D.C.
New_York
Montreal
Manila
Lisbon
Hong_Kong
Tokyo
Cairo
Shanghai
Istanbul
Moscow
Berlin
Stockholm
Berlin
Buenos_Aires
Cairo
Calcutta
Cape_Town
Caracas
Chicago
Hong_Kong
Honolulu
Istanbul
Lisbon
London
Los_Angeles
Manila
Mexico_City
Montreal
Moscow
New_York
Paris
Rio_de_Janeiro
Rome
San_Francisco
Shanghai
Stockholm
Sydney
Tokyo
Warsaw
Washington,_D.C.

Για καλύτερη σύγκριση χωρίς τις ετικέτες μπορούμε να δούμε τα διαγράμματα το ένα δίπλα στο άλλο.



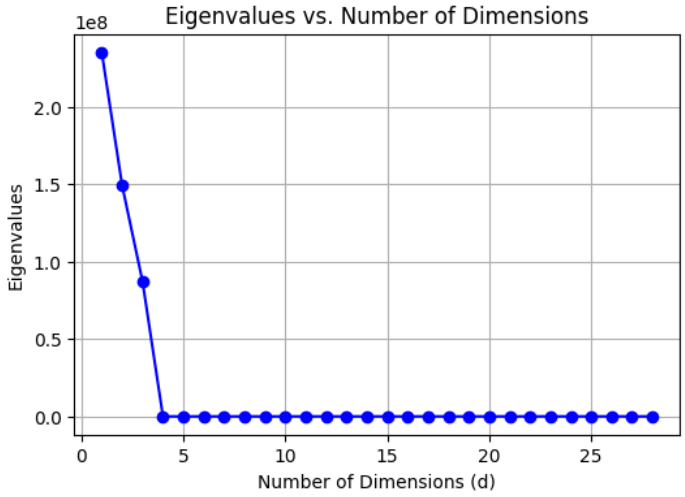

## B. Δημιουργείστε για τις πόλεις μία διανυσματική αναπαράσταση με τις μέγιστες διαστάσεις d που μπορεί να σας επιστρέψει ο αλγόριθμος MDS. Εάν Y ο πίνακας με τις αναπαραστάσεις για τις N πόλεις, να δημιουργήστε το διάγραμμα των ιδιοτιμών του πίνακα Y*Y^T σε φθίνουσα σειρά. Το διάγραμμα αυτό χρησιμοποιείται ως ένδειξη της βέλτιστης διάστασης αναπαράστασης, διατηρώντας τόσες διαστάσεις όσες οι σημαντικές ιδιοτιμές. Με βάση αυτό, πόσες διαστάσεις εκτιμάτε ότι είναι οι βέλτιστες για τα δεδομένα του αρχείου Distance_Matrix_world;

Δημιουργούμε για τις πόλεις μία διανυσματική αναπαράσταση με τις μέγιστες διαστάσεις d=28 (έχουμε 28 πόλεις-28 αποστάσεις) που μπορεί να σας επιστρέψει ο αλγόριθμος MDS (η μέγιστη διάσταση για τον MDS περιορίζεται από τον αριθμό των δειγμάτων στο σύνολο δεδομένων).

Οι ιδιοτιμές του πίνακα Y*Y^T σε φθίνουσα σειρά είναι:

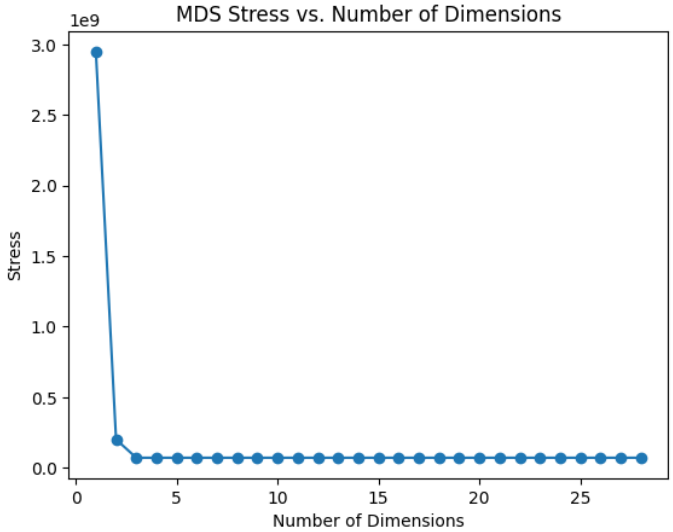
```
Eigenvalues in descending order are:
[2.35339829e+08 1.49210123e+08 8.72224859e+07 8.22874169e+02
 1.94654247e+02 1.23265854e+02 9.77572478e+01 6.72743515e+01
 4.76622519e+01 3.69978891e+01 2.79501668e+01 2.21469855e+01
 1.87263739e+01 9.26629997e+00 7.30275555e+00 6.21958120e+00
 5.12332114e+00 2.47172740e+00 2.08258801e+00 1.84230280e+00
 1.76600162e+00 9.55298721e-01 8.36909205e-01 4.67327954e-01
 2.80091738e-01 6.03792481e-02 4.13042424e-02 3.31307673e-09]
```

Το διάγραμμα των ιδιοτιμών του πίνακα Y*Y^T είναι το ακόλουθο.



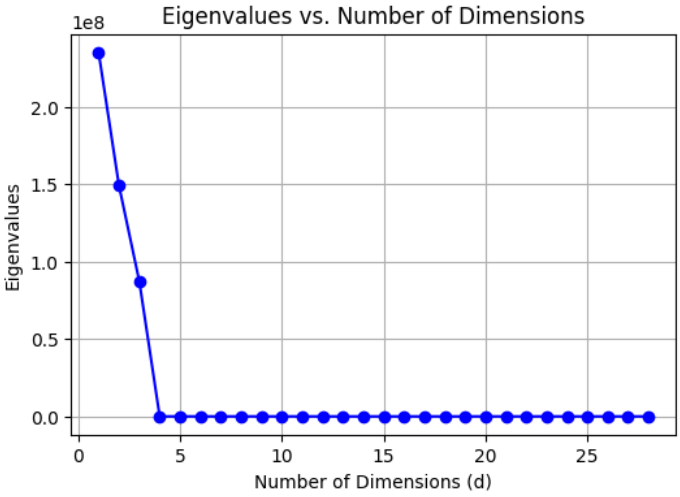

Από το διάγραμμα, για τη βέλτιστης διάστασης αναπαράσταση, πρέπει να διατηρηθούν τόσες διαστάσεις όσες και οι σημαντικές ιδιοτιμές (σταματάω στη διάσταση στην οποία το διάγραμμα πλησιάζει το μηδέν). Οι ιδιοτιμές από 4 και μετά αν και δεν είναι κοντά στο μηδέν, πρακτικά έχουν πάρα πολύ μεγάλη διαφορά στην τάξη μεγέθους (10^7 και 10^2) , οπότε θεωρούνται ασήμαντες σε σχέση με τις άλλες τρείς. Οπότε, οι βέλτιστες για τα δεδομένα του αρχείου Distance_Matrix_world είναι οι **τρείς διαστάσεις** με βάση τις σημαντικότερες ιδιοτιμές.

Ένας άλλος τρόπος να βρούμε τις βέλτιστες διαστάσεις για την απεικόνιση είναι η τιμή stress που επιστρέφει ο MDS. Το stress είναι ένα μέτρο που αναπαριστά την απόκλιση μεταξύ των αρχικών αποστάσεων και των αποστάσεων σε χαμηλή διάσταση. Όσο χαμηλότερο το stress, τόσο καλύτερη η αναπαράσταση, δεδομένου ότι προσεγγίζει περισσότερο τις αρχικές αποστάσεις. Για stress ίσο με 0 έχω τέλεια προσέγγιση των αρχικών αποστάσεων. Οπότε, κρατάμε τις χαμηλότερες διαστάσεις για τις οποίες η τιμή stress γίνεται ελάχιστη (πλησιάζει το μηδέν), δηλαδή για **διάσταση 3**.



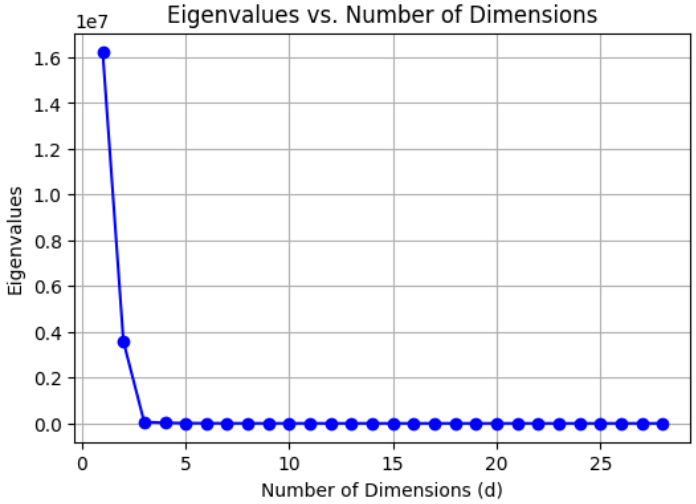
## Bonus: Για ποιο λόγο υπάρχουν μη μηδενικές ιδιοτιμές για περισσότερες των 3 διαστάσεων στο παραπάνω πρόβλημα; Συγκρίνετε το αντίστοιχο διάγραμμα για τα δεδομένα του αρχείου Distance_Matrix_US. Που οφείλεται αυτή η διαφορά και πως σχετίζεται με τη φύση του προβλήματος και των δεδομένων;

Για Distance_Matrix_world ιδιοτιμές και διάγραμμα.




```
Eigenvalues in descending order are:
[2.35339829e+08 1.49210123e+08 8.72224859e+07 8.22874169e+02
 1.94654247e+02 1.23265854e+02 9.77572478e+01 6.72743515e+01
 4.76622519e+01 3.69978891e+01 2.79501668e+01 2.21469855e+01
 1.87263739e+01 9.26629997e+00 7.30275555e+00 6.21958120e+00
 5.12332114e+00 2.47172740e+00 2.08258801e+00 1.84230280e+00
 1.76600162e+00 9.55298721e-01 8.36909205e-01 4.67327954e-01
 2.80091738e-01 6.03792481e-02 4.13042424e-02 3.31307673e-09]
```

Για Distance_Matrix_US ιδιοτιμές και διάγραμμα.




```
Eigenvalues in descending order are:
[1.62434992e+07 3.57768729e+06 5.90520338e+04 2.43261678e+04
 9.53948520e+03 5.71211995e+03 3.12757931e+03 2.34385697e+03
 1.63236373e+03 1.30533171e+03 1.12127392e+03 9.10334900e+02
 6.59680520e+02 5.62432780e+02 4.98803812e+02 2.90635529e+02
 2.20258588e+02 1.91450588e+02 1.50435810e+02 9.89658605e+01
 7.45139032e+01 6.62899394e+01 3.16958279e+01 2.37040556e+01
 1.03148115e+01 5.95971261e+00 1.61134891e+00 1.03113860e-10]
```

Με βάση τις ιδιοτιμές παρατηρούμε ότι για τα δεδομένα του αρχείου **Distance Matrix world οι βέλτιστες διαστάσεις είναι τρείς**, ενώ για τα δεδομένα του αρχείου **Distance Matrix US οι βέλτιστες διαστάσεις είναι δύο**. Επίσης, οι σημαντικές ιδιοτιμές για τα δεδομένα του αρχείου Distance Matrix world είναι μεγαλύτερης τάξης από τις ιδιοτιμές για τα δεδομένα του αρχείου Distance Matrix US.

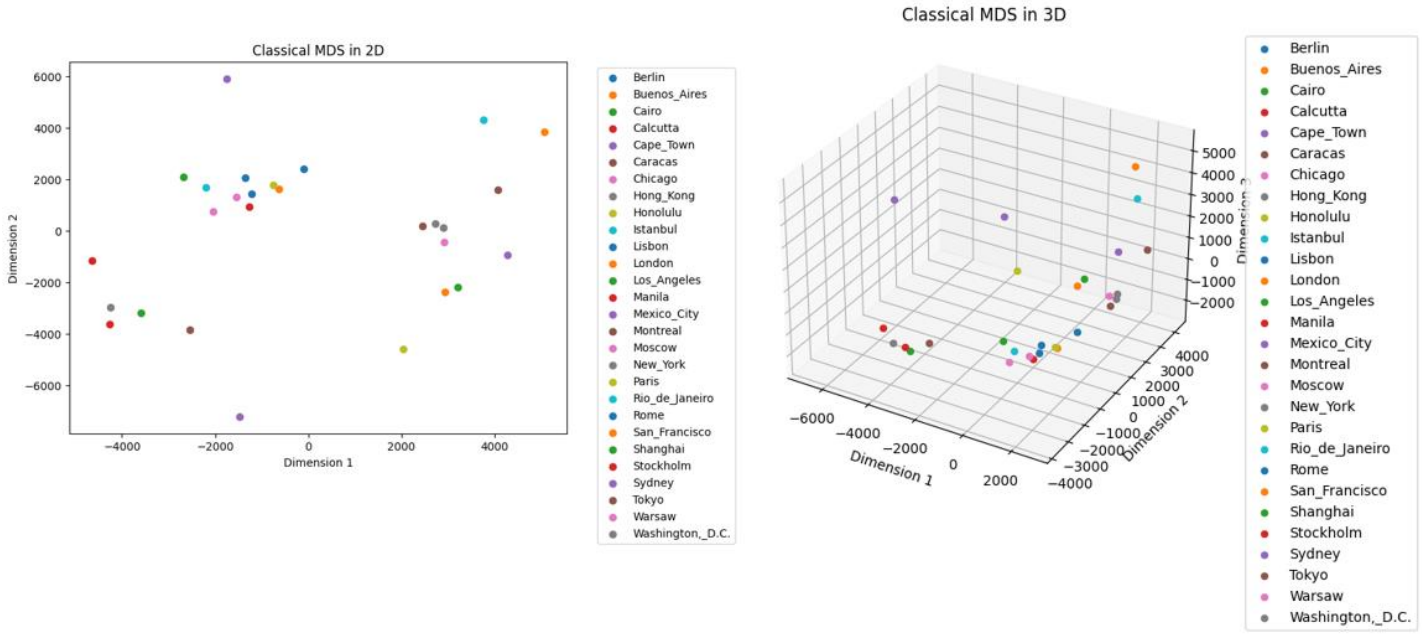
Πιθανότατα η διαφορά στις διαστάσεις για τα δύο προβλήματα να οφείλεται στο γεγονός ότι για να απεικονίσουμε τις αποστάσεις πόλεων σε όλον τον κόσμο, οι οποίες έχουν μετρηθεί κατά μήκος μέγιστων κύκλων (δηλαδή πάνω σε μια υδρόγειο σφαίρα), εφόσον οι πόλεις εκτείνονται σε όλη την επιφάνεια της σφαίρας που έχει τρείς διαστάσεις, ιδανικά θα

θέλαμε να χρησιμοποιήσουμε τρείς διαστάσεις για καλύτερη αναπαράσταση του πραγματικού κόσμου. Στις δύο διαστάσεις (πχ. χάρτης) μπορούμε να λάβουμε αρκετή πληροφορία, η οποία να μας ικανοποιεί σε ορισμένες περιπτώσεις, αλλά σίγουρα θα υπάρχει απόκλιση από την πραγματικότητα και σε κάποια σημεία θα είναι εμφανής. Όσον αφορά τις αποστάσεις πόλεων των Ηνωμένων Πολιτειών, αν και πάλι βρισκόμαστε πάνω σε μια σφαίρα, θα μπορούσαμε να θεωρήσουμε χωρίς μεγάλο σφάλμα ότι βρισκόμαστε σε ένα επίπεδο, επειδή όλες οι πόλεις της Αμερικής είναι συγκεντρωμένες σε μια περιοχή της σφαίρας και δεν εκτείνονται σε όλο το μήκος της (για μικρή απόσταση σε σχέση με τις παγκόσμιες αποστάσεις μπορεί κατά προσέγγιση ο κύκλος να θεωρηθεί ευθεία γραμμή). Άρα λοιπόν οι δύο διαστάσεις μας ικανοποιούν και δε χρειάζεται να πάμε σε περισσότερες.

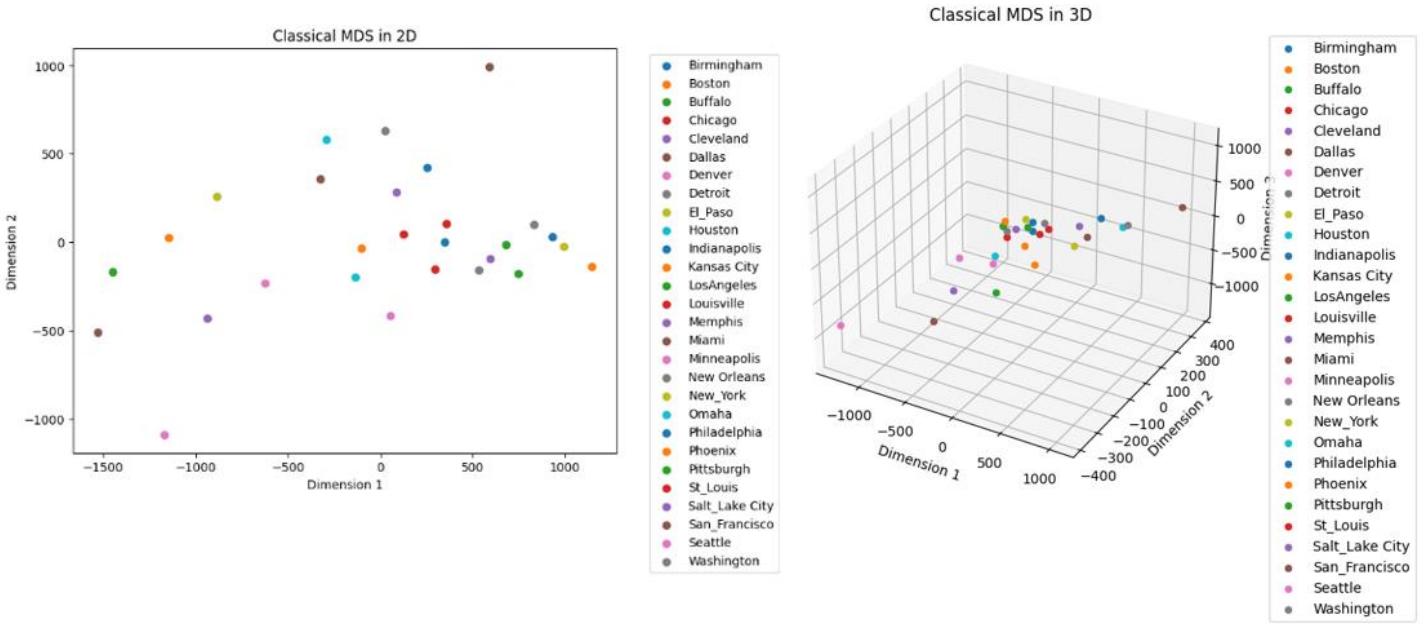
Επίσης, οι παγκόσμιες αποστάσεις είναι κατά βάση μεγαλύτερες από τις αποστάσεις μεταξύ πόλεων της Αμερικής. Ίσως για αυτό οι σημαντικές ιδιοτιμές στο Distance Matrix world να είναι μεγαλύτερες από τις ιδιοτιμές στο Distance Matrix US. Αυτό φαίνεται και στην κλίμακα των διαγραμμάτων, στο 3D διάγραμμα για το Distance Matrix US η έκταση του άξονα για dimension 2 είναι εμφανώς μικρότερη από τις άλλες δύο διαστάσεις. Αυτό μπορεί να υποδεικνύει ότι οι παγκόσμιες αποστάσεις είναι πιο διαφοροποιημένες και περιλαμβάνουν περισσότερες πληροφορίες από τις αποστάσεις μόνο στην Αμερική.

Παρακάτω φαίνονται οι αναπαραστάσεις για τα δύο αρχεία σε 2D και 3D. Στο 3D διάγραμμα οι πόλεις του κόσμου είναι πιο απλωμένες σε σχέση με τις πόλεις της Αμερικής που είναι πιο συγκεντρωμένες στο κέντρο (με εξαίρεση κάποια σημεία).

Για Distance_Matrix_world



Για Distance_Matrix_US




Οι ιδιοτιμές του χώρου χαμηλότερων διαστάσεων αντιπροσωπεύουν την ποσότητα της διακύμανσης που εξηγείται από κάθε διάσταση. Στην περίπτωση που έχουμε βέλτιστες d διαστάσεις για ένα πρόβλημα, οι υπόλοιπες ιδιοτιμές που δεν αντιστοιχούν στις επιλεγμένες διαστάσεις μπορεί να μην μηδενίζονται. Οι μη μηδενισμένες ιδιοτιμές (από τις μη σημαντικές) αντιστοιχούν σε επιπλέον διαστάσεις που περιλαμβάνουν κάποιο ποσοστό πληροφορίας, το οποίο όμως είναι πολύ μικρό σε σχέση με το ποσοστό της πληροφορίας που μας παρέχουν οι σημαντικές ιδιοτιμές και μπορεί να θεωρηθεί αμελητέο.

Η απόσταση κατά μήκος μεγίστων κύκλων λαμβάνει υπόψη της το σχήμα της Γης. Αυτή η απόσταση υπολογίζεται στην επιφάνεια μιας σφαίρας και αντιπροσωπεύει την συντομότερη απόσταση μεταξύ δύο σημείων σε αυτήν τη σφαίρα, λαμβάνοντας υπόψη τη σφαιρική γεωμετρία και προσφέρει μια καλή προσέγγιση των πραγματικών αποστάσεων.

Η υλοποίηση του MDS που παρέχει η βιβλιοθήκη scikit-learn είναι βασισμένη στον κλασικό αλγόριθμο MDS (classical), ο οποίος προσπαθεί να διατηρήσει τις ευκλείδειες αποστάσεις μεταξύ των σημείων. Ο κλασικός αλγόριθμος MDS, δεν λαμβάνει αυτόματα υπόψη τη σφαιρική γεωμετρία της γης, ακόμα και αν οι αποστάσεις που του δίνουμε είναι υπολογισμένες κατά μήκος μέγιστων κύκλων (στην επιφάνεια μιας σφαίρας). Ο MDS προσπαθεί να απεικονίσει τις αποστάσεις μεταξύ των σημείων σε έναν χώρο χαμηλότερης διάστασης, χωρίς να λαμβάνει υπόψη την πραγματική γεωμετρία της Γης. Για αυτό το λόγο οι μη σημαντικές ιδιοτιμές δεν μηδενίζονται.

Συνοπτικά, μπορεί να είναι απαραίτητο να χρησιμοποιήσουμε προσαρμοσμένες μεθόδους MDS που λαμβάνουν υπόψη τη γεωγραφική φύση των δεδομένων για να επιτύχουμε μεγαλύτερη ακρίβεια στην αναπαράσταση.