Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Χειμερινό εξάμηνο 2022-23

Αναγνώριση Προτύπων

Προπαρασκευή 2ης Εργαστηριακής Άσκησης: Αναγνώριση φωνής με Κρυφά Μαρκοβιανά Μοντέλα και Αναδρομικά Νευρωνικά Δίκτυα

Δημήτριος Κοκκίνης 03118896
Χριστίνα Ρεντίφη 03118217

Περιγραφή Προπαρασκευής

Στο εργαστήριο αυτό θα δημιουργήσουμε ένα σύστημα επεξεργασίας και αναγνώρισης φωνής με εφαρμογή σε αναγνώριση μεμονωμένων λέξεων. Στην συγκεκριμένη άσκηση θα τροφοδοτούμε το σύστημά μας με φωνητικά δεδομένα που περιέχουν την ανάγνωση μεμονωμένων ψηφίων από το 0 έως το 9 και θα προσπαθούμε να εντοπίσουμε σε ποια κλάση καθένα από αυτά ανήκει. Μας δίνονται για τον σκοπό αυτό 133 αρχεία, που αντιστοιχούν στις εκφωνήσεις των 9 ψηφίων από 15 διαφορετικούς εκφωνητές έχοντας αφαιρέσει 2 αρχεία που θεωρήθηκαν προβληματικά. Κάθε αρχείο από αυτά έχει ονοματιστεί με τέτοιο τρόπο ώστε να δηλώνει α) το ψηφίο που εκφωνείται και β)τον εκφωνητή.

Επειδή τα φωνητικά δεδομένα δεν είναι σε μορφή που μπορεί να χειριστεί το σύστημά μας, πρώτος μας στόχος είναι η εξαγωγή κατάλληλων ακουστικών χαρακτηριστικών από αυτά, σε μορφή συντελεστών που είναι γνωστά με το όνομα cepstrum. Σε κατάλληλη μορφή πλέον (cepstrum συντελεστές) θα εισάγουμε τα δεδομένα μας στο σύστημα αναγνώρισης και θα τα ταξινομήσουμε έπειτα στις 9 κλάσεις των ψηφίων.

Ο κώδικας μας θα υλοποιηθεί σε python και για τους σκοπούς του συγκεκριμένου εργαστηρίου θα χρησιμοποιήσουμε πέρα από τις γνωστές βιβλιοθήκες και την librosa για την ανάγνωση αρχείων ήχου και εξαγωγή χαρακτηριστικών.

Επίσης για την εμφάνιση των κυματομορφών των φωνητικών δεδομένων θα χρησιμοποιήσουμε και το λογισμικό Praat.

Βήμα 1

Αρχικά ανοίγουμε με την βοήθεια του Praat τα αρχεία onetwothree1.wav και onetwothree8.wav που περιέχουν την πρόταση “one two three” από τους ομιλητές 1 και 8 που είναι άντρας και γυναίκα αντίστοιχα.
Οι κυματομορφές που προκύπτουν από τους δύο ομιλητές είναι οι εξής:

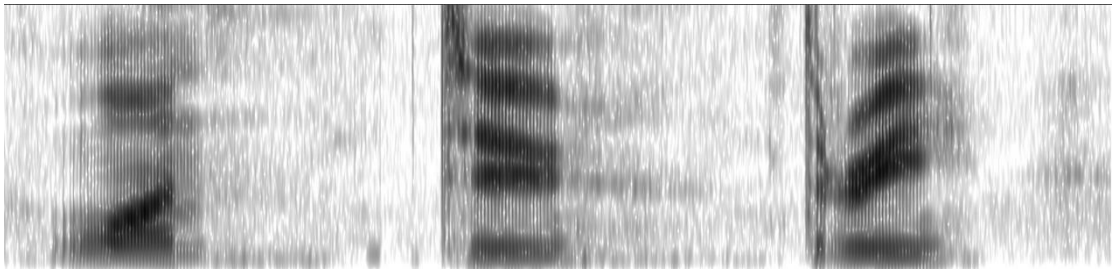
- Άντρας

- Γυναίκα

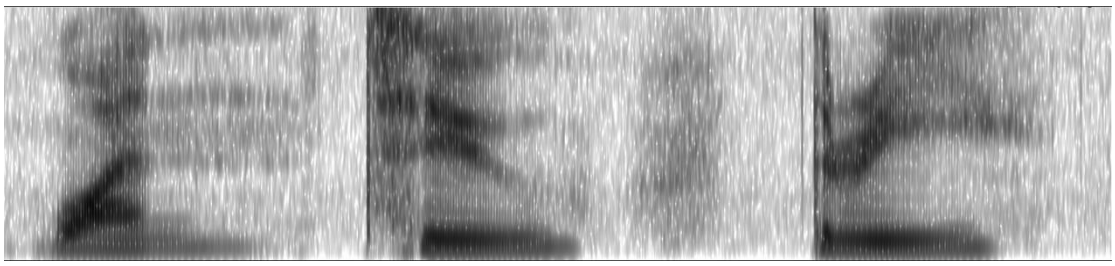
Όπως βλέπουμε η κυματομορφή που αντιστοιχεί στην γυναίκα έχει μεγαλύτερες συχνότητες σε σχέση με αυτήν του άντρα. Επίσης η ένταση φωνής της γυναίκας εκφωνήτριας είναι υψηλότερη σε σχέση με αυτήν του άντρα εκφωνητή (κυρίως κατά την εκφώνηση της λέξης “two”)

Τα αντίστοιχα spectrograms που προκύπτουν για τους δύο ομιλητές είναι:

- Άντρας



- Γυναίκα



Με τα spectrograms βλέπουμε την φασματική πυκνότητα της φωνής, μπορούμε δηλαδή να διακρίνουμε το πόσο δυνατά ή σιγανά μίλησε ο κάθε εκφωνητής. Βλέπουμε δύο διαστάσεις όπως φαίνεται και παραπάνω και η τρίτη διάσταση παρουσιάζεται με την μορφή χρώματος (εδώ αποχρώσεις από λευκό εως μάυρο). Στον οριζόντιο άξονα έχουμε την διάσταση του χρόνου (αυξάνεται από αριστερά προς τα δεξιά). Ο κατακόρυφος άξονας αντιστοιχεί στην συχνότητα και μπορεί να θεωρηθεί ως το pitch (τόνος) της φωνής με τις χαμηλότερες συχνότητες να ξεκινάνε από κάτω και να αυξάνονται καθώς ανεβαίνουμε κατακόρυφα. Το χρώμα που αντιπροσωπεύει την τρίτη διάσταση μας δείχνει το πλάτος (ή ενέργεια ή διαφορετικά ένταση ”loudness”) μιας δεδομένης συχνότητας σε μια δεδομένη χρονική στιγμή. Τα πιο σκούρα χρώματα αντιπροσωπεύουν χαμηλότερα πλάτη (πιο σιγανή φωνή) και τα πιο ανοιχτά πιο υψηλά πλάτη (πιο δυνατή φωνή).

Με το φασματογράφημα επιβεβαιώνεται και η παρατήρηση που κάναμε προηγουμένως βασισμένοι στις κυματομορφές, ότι η γυναίκα μιλάει με πιο έντονη-δυνατή φωνή.

Με την χρήση του Praat και πάλι, επιλέγουμε το διάστημα όπου εκφωνούνται τα φωνήεντα α”, “ου”, “ι” μέσα στις λέξεις “one”, “two” και “three” αντίστοιχα για κάθε ομιλητή και εξάγουμε την μέση τιμή του pitch (τόνου φωνής). Έτσι παίρνουμε τις εξής τιμές:

Φύλο	Φωνήεν	Mean_pitch value
Άνδρας	α	134.53
Άνδρας	ου	132.49
Άνδρας	ι	132.52
Γυναίκα	α	178.94
Γυναίκα	ου	188.84
Γυναίκα	ι	179.54

Παρατηρούμε ότι η μέση τιμή του pitch κάθε φωνήεντος για την γυναίκα είναι μεγαλύτερη σε σχέση με τις αντίστοιχες τιμές για τον άνδρα. Αυτό σημαίνει ότι η γυναικεία φωνή εμφανίζει μεγαλύτερη οξύτητα σε σχέση με την αντρική.

Στη συνέχεια εξάγουμε τα 3 πρώτα formants από τα τρία φωνήεντα για τους δύο ομιλητές μας. Τα formants σχετίζονται με τον λάρυγγα και τις φωνητικές χορδές και είναι διαφορετικά για τον κάθε άνθρωπο έχοντας σαν αποτέλεσμα την μοναδικότητα της φωνής του. Ουσιαστικά δηλώνουν τα σημεία όπου μία συχνότητα μεγιστοποιείται και άρα εκεί όπου έχουμε συσσωρευμένη ενέργεια. Η φωνή μας παράγει έναν πολύ μεγάλο αριθμό από formants με τα τρία πρώτα από αυτά να είναι τα πιο σημαντικά.

Τα αποτελέσματα που παίρνουμε είναι τα εξής:

Φύλο	Φωνήεν	Formant 1	Formant 2	Formant 3
Άνδρας	α	821.9	1350.22	2428.63
Άνδρας	ου	417.65	1796.64	2455.87
Άνδρας	ι	434.11	1957.22	2420.44
Γυναίκα	α	708.74	1809.74	3035.72
Γυναίκα	ου	338.79	1732.86	2691.69
Γυναίκα	ι	424.06	2132.10	2816.29

Παρατηρούμε εδώ πως τα αντίστοιχα formants για τον άνδρα είναι υψηλότερα σε σχέση με αυτά της γυναίκας ομιλήτριας. Επίσης για κάθε φωνήεν τόσο στον άνδρα όσο και στην γυναίκα τα αντίστοιχα formants παρουσιάζουν μια σημαντική διαφορά στις τιμές τους, την οποία και θεωρούμε αρκετή ώστε να μπορούμε να διακρίνουμε για ποιο φωνήεν γίνεται λόγος κάθε φορά.

Από την άλλη παρατηρώντας τις τιμές του mean pitch των τριών φωνηέντων για κάθε εκφωνητή αντίστοιχα, βλέπουμε πως αυτές δεν διαφέρουν σχεδόν καθόλου μεταξύ τους. Θα μπορούσαμε να υποθέσουμε λοιπόν πως το mean

pitch αποτελεί μια μέτρηση που θα μας βοηθήσει να κατατάξουμε τον ομιλητή σε άντρα ή γυναίκα.

Βήμα 2

Δημιουργούμε μια συνάρτηση data_parser() που διαβάζει όλα τα αρχεία που δίνονται στον φάκελο digits/ και επιστρέφει 3 λίστες που περιέχουν:

- Τον αριθμό του αρχείου wav που διαβάστηκε με librosa (wav_file λίστα)
- τον αντίστοιχο ομιλητή (speaker λίστα)
- το ψηφίο (digit λίστα)

Τυπώνουμε ενδεικτικά τα στοιχεία που προκύπτουν από τις τρεις αυτές λίστες για τα δέκα πρώτα αρχεία :

```
First 10 parsed files:

wav_file: 1 speaker =7 digit =5
wav_file: 2 speaker =3 digit =5
wav_file: 3 speaker =8 digit =8
wav_file: 4 speaker =11 digit =5
wav_file: 5 speaker =12 digit =5
wav_file: 6 speaker =4 digit =5
wav_file: 7 speaker =1 digit =8
wav_file: 8 speaker =14 digit =8
wav_file: 9 speaker =10 digit =5
wav_file: 10 speaker =2 digit =8
```

Βήμα 3

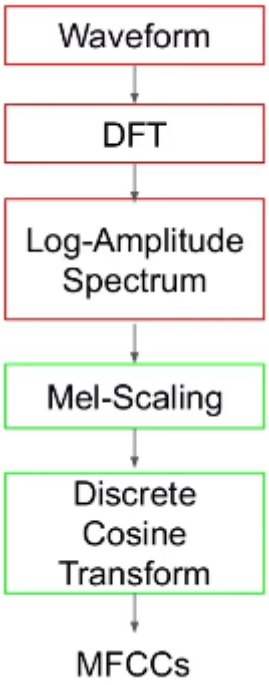
Χρησιμοποιώντας την βιβλιοθήκη librosa εξάγουμε τα 13 πρώτα Mel-Frequency Cepstral Coefficients (MFCCs) για κάθε αρχείο ήχου.

Τα MFCCs ενός σήματος είναι ουσιαστικά ένα μικρό set συντελεστών-χαρακτηριστικών (συνήθως 10-20) που περιγράφει συνοπτικά το συνολικό σχήμα του φάσματος ισχύος ενός ήχου. Συχνά οι συντελεστές αυτοί χρησιμοποιούνται για να περιγράψουν το ηχόχρωμα ενός ήχου, δηλαδή την “υφή” του.

Περιληπτικά οι συντελεστές αυτοί προκύπτουν από τον Fourier Transform ενός μέρους του συνολικού σήματος που έχει παρθεί χρησιμοποιώντας κάποιο παράθυρο (window). Ο λόγος που χρησιμοποιούμε window είναι για να μπορούμε απλά να διαχειριστούμε ευκολότερα ένα μικρό κομμάτι της κυματομορφής αντί για ολόκληρη. Το φάσμα ισχύος που προκύπτει από τον Fourier Transform αντιστοιχίζεται στην mel-scale (melody κλίμακα). Έπειτα παίρνουμε τους λογάριθμους (log) των ενεργειών σε κάθε mel-frequency και εφαρμόζοντας σε αυτούς Discrete Cosine Transform παίρνουμε ένα καινούριο

φασματογράφημα. Τα πλάτη που προκύπτουν από το τελευταίο spectrogram είναι οι MFCCs.

Παραθέτουμε εποπτικά ένα σχήμα που περιγράφει την παραπάνω διαδικασία:



Υλοποιούμε το βήμα 3 χρησιμοποιώντας την συνάρτηση librosa.feature.mfcc (). Για τα ορίσματα της συνάρτησης αυτής έχουμε το wav_file[i] από το οποίο διαβάζουμε τα αρχεία ήχου, συχνότητα δειγματοληψίας 16kHz (sr = 16000) όπως μας διευκρινίζεται και στην εκφώνηση, αριθμό συντελεστών mfcc ίσο με 13(n_mfcc=13), μήκος παραθύρου 25 ms (n_fft = int(sr * 0.025) και βήμα 10 ms (hop_length = int(sr * 0.01)

Υπολογίζουμε ακόμη, όπως μας ζητείται, την πρώτη και δεύτερη τοπική παράγωγο των χαρακτηριστικών, τις λεγόμενες deltas και delta-deltas χρησιμοποιώντας τις έτοιμες υλοποιήσεις της librosa :

```
    # Derivatives
    delta.append(librosa.feature.delta(mfcc))
    delta2.append(librosa.feature.delta(mfcc, order=2))
```

Η delta ορίζεται ως

$$\Delta_k = f_k - f_{k-1}$$

όπου fk και fk-1 τα χαρακτηριστικά για μια δεδομένη χρονική στιγμή k και k-1,

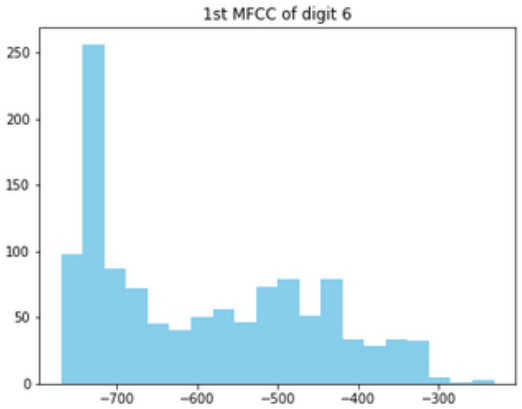
και η delta2 ορίζεται ως

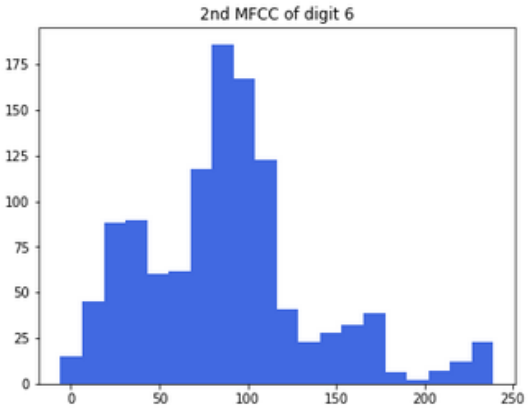
$$\Delta\Delta_k = \Delta_k - \Delta_{k-1}$$

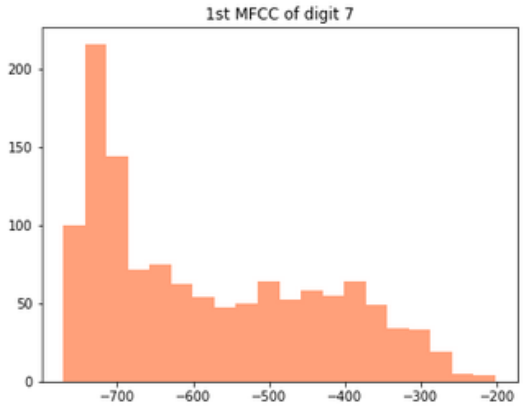
Βήμα 4

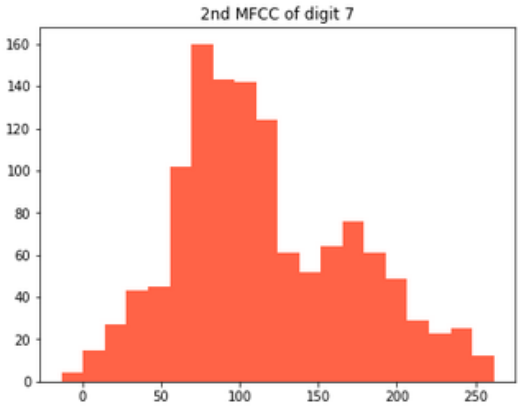
Αφού υλοποιήσαμε το προηγούμενο βήμα, αναπαριστούμε τώρα τα ιστογράμματα που προκύπτουν για το 1ο και 2ο MFCC των ψηφίων 6 και 7 (που αντιστοιχούν στα τελευταία ψηφία των ΑΜ μας) βάσει όλων των εκφωνήσεών τους. Η λίστα mfcc1_d6 αναφέρεται στους πρώτους MFC του ψηφίου 6 από όλες τις εκφωνήσεις του και η λίστα mfcc2_d6 αναφέρεται στους δεύτερους συντελεστές MFC του ψηφίου 6 από όλες τις εκφωνήσεις. Αντίστοιχα ορίζονται οι λίστες mfcc1_d7 και mfcc2_d7 για το ψηφίο 7. Για να βάλουμε τώρα αυτές τις λίστες στην συνάρτηση plt.hist() δημιουργούμε τις αντίστοιχες flattened εκδοχές τους όπου έχουμε μειώσει τις 13 διαστάσεις σε μία. Στην συνάρτηση plt.hist() χρησιμοποιούμε bins=20 (αριθμός “κουτιών” που φαίνονται)

Οι εικόνες που παίρνουμε είναι οι εξής:









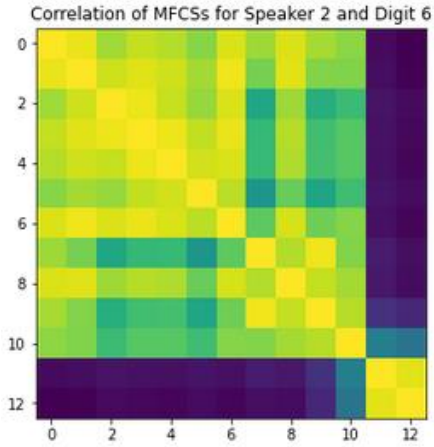
Παρατηρούμε πως οι κατανομές που προκύπτουν τόσο για τον 1ο όσο και για τον 2ο mfcc δεν εμφανίζουν κάποια σημαντική απόκλιση μεταξύ των δύο διαφορετικών ψηφίων παρότι η εκφωνήσεις “six” και “seven” αντίστοιχα είναι αρκετά διαφορετικές μεταξύ τους.

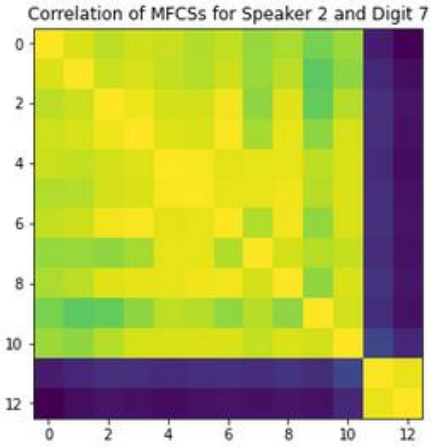
Για καθένα από τα δύο ψηφία, η απόκλιση του 2ου mfcc από το 1ο mfcc ισούται με την πρώτη τοπική παράγωγο delta που υπολογίστηκε στο προηγούμενο βήμα

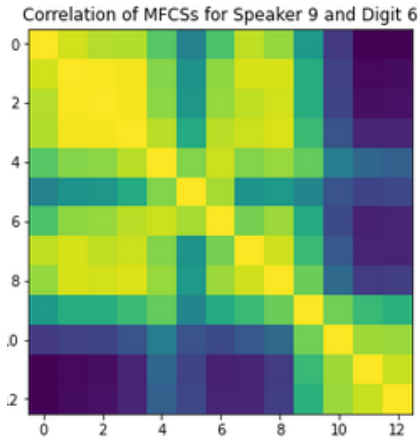
Έπειτα εξάγουμε για τις εκφωνήσεις των ψηφίων 6 και 7 από δύο διαφορετικούς ομιλητές (επιλέξαμε τυχαία τον εκφωνητή 2 και 9) τα Mel Filterbank Spectral Coefficients.

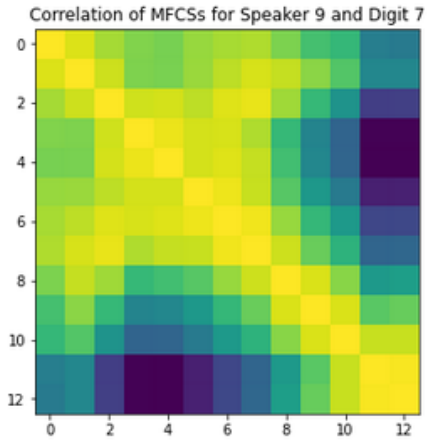
Οι συντελεστές MFSCs προκύπτουν ουσιαστικά αφού έχει εφαρμοσθεί η συστοιχία φίλτρων της κλίμακας Mel στο σημείο πριν να γίνει όμως ο μετασχηματισμός DCT.

Παρακάτω εμφανίζουμε γραφικά τη συσχέτιση (correlation) των MFSCs για την κάθε εκφώνηση καθενός από τα ψηφία 6 και 7:



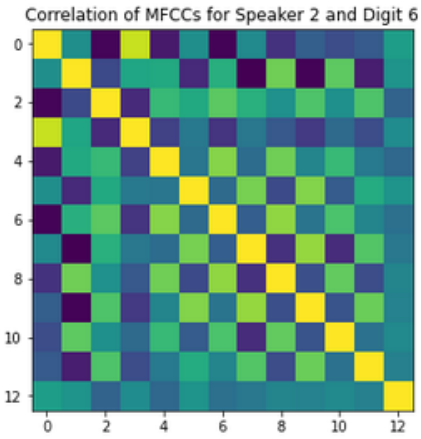


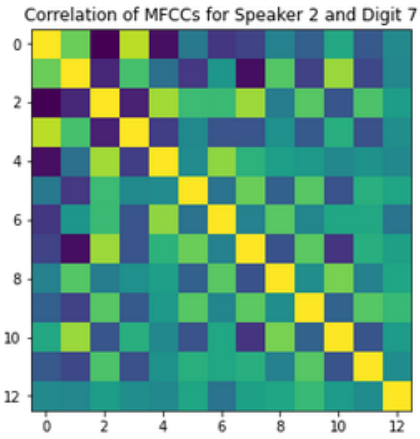


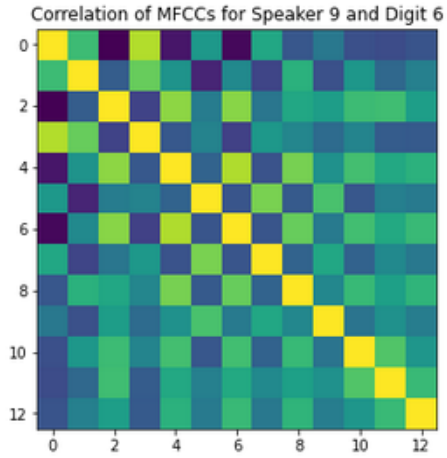


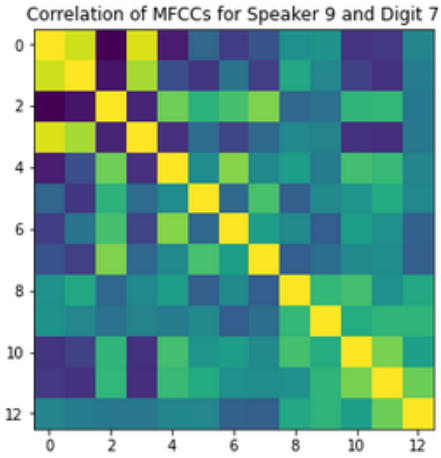
Όπως εύκολα μπορούμε να διακρίνουμε από τα παραπάνω σχήματα, οι MFCSs συντελεστές εμφανίζουν μεγάλη συσχέτιση μεταξύ τους στη διάρκεια του χρόνου και δεν μπορούμε έτσι να βγάλουμε με βεβαιότητα συμπεράσματα για τα διαφορετικά χαρακτηριστικά.

Σε ξεχωριστά διαγράμματα εμφανίζουμε αντίστοιχα την συσχέτιση των MFCCs για κάθε ψηφίο:









Βλέπουμε τώρα εδώ από την άλλη πως οι MFCCs συντελεστές εμφανίζουν ελάχιστη συσχέτιση μεταξύ τους και διακρίνουμε εμφανώς τις εναλλαγές από χαμηλές σε υψηλές τιμές και αντίστροφα (αλλαγές στα χρώματα) με τις πιο σκούρες αποχρώσεις να αντιστοιχούν πάντα στις μικρότερες τιμές.

Τα παραπάνω αποτελέσματα θα μπορούσαμε να πούμε πως δεν μας προξενούν εντύπωση καθώς οι συντελεστές που προκύπτουν απευθείας από τον Fourier Transform είναι μεγάλοι και ιδιαίτερα περίπλοκοι αριθμοί, οι οποίοι μπορεί να παρουσιάζουν ελάχιστη διαφορά μεταξύ τους όπως φαίνεται από τις πολύ ήπιες χρωματικές μεταβάσεις των δύο πρώτων φασματογραφημάτων διαβάζοντάς τα κατακόρυφα, για κάθε δεδομένη χρονική στιγμή που βρίσκεται στον οριζόντιο άξονα.

Από την άλλη, εφαρμόζοντας DCT μετασχηματισμό, οι τιμές που παίρνουμε είναι πραγματικοί και λιγότερο πολύπλοκοι αριθμοί που όπως φαίνεται από το 2ο set φασματογραφημάτων που προκύπτουν είναι αρκετά διακριτοί μεταξύ τους, καθώς οι χρωματικές μεταβάσεις κατά τον κατακόρυφο άξονα είναι πιο απότομες.

Όπως έχουμε αναφέρει οι συντελεστές cepstral που εξάγουμε από ένα φωνητικό δείγμα μας παρέχουν πληροφορία για το περιεχόμενό του, επομένως είναι λογικό να θέλουμε να προκύπτουν όσο το δυνατόν πιο πολλές διαφορετικές τιμές καθώς κάθε μία από αυτές τις τιμές μας δίνει και μια νέα διαφορετική πληροφορία.

Έτσι, είναι επόμενο βάσει των παραπάνω παρατηρήσεων να προτιμήσουμε να χρησιμοποιήσουμε MFCCs έναντι των MFSCs καθώς για δεδομένο αριθμό συντελεστών, με ασυσχέτιστους συντελεστές μπορούμε να λάβουμε μεγαλύτερο όγκο πληροφορίας σε σχέση με το να είχαμε σχετιζόμενους συντελεστές όπου υπάρχει σημαντική αλληλοεπικάλυψη πληροφορίας.

Βήμα 5

Έχοντας υπολογίσει τα ακουστικά χαρακτηριστικά που μπορούμε να εξάγουμε από τα φωνητικά μας αρχεία και βλέποντας τις διαφορετικές πληροφορίες που παίρνουμε μελετώντας καθένα από αυτά χωριστά, θέλουμε τώρα να κατασκευάσουμε ένα μοναδικό διάνυσμα για καθένα από τα 133 αρχεία που να συνδυάζει όλα αυτά τα επιμέρους χαρακτηριστικά και να μας δίνει έτσι την συνολική πληροφορία για κάθε αρχείο.

Καθώς έχουμε πληροφορία από 15 διαφορετικούς εκφωνητές για κάθε ψηφίο θα χρησιμοποιήσουμε τη μέση τιμή καθενός από αυτά τα χαρακτηριστικά καθώς και την τυπική απόκλιση που προκύπτει για όλα τα παράθυρα της εκφώνησης.

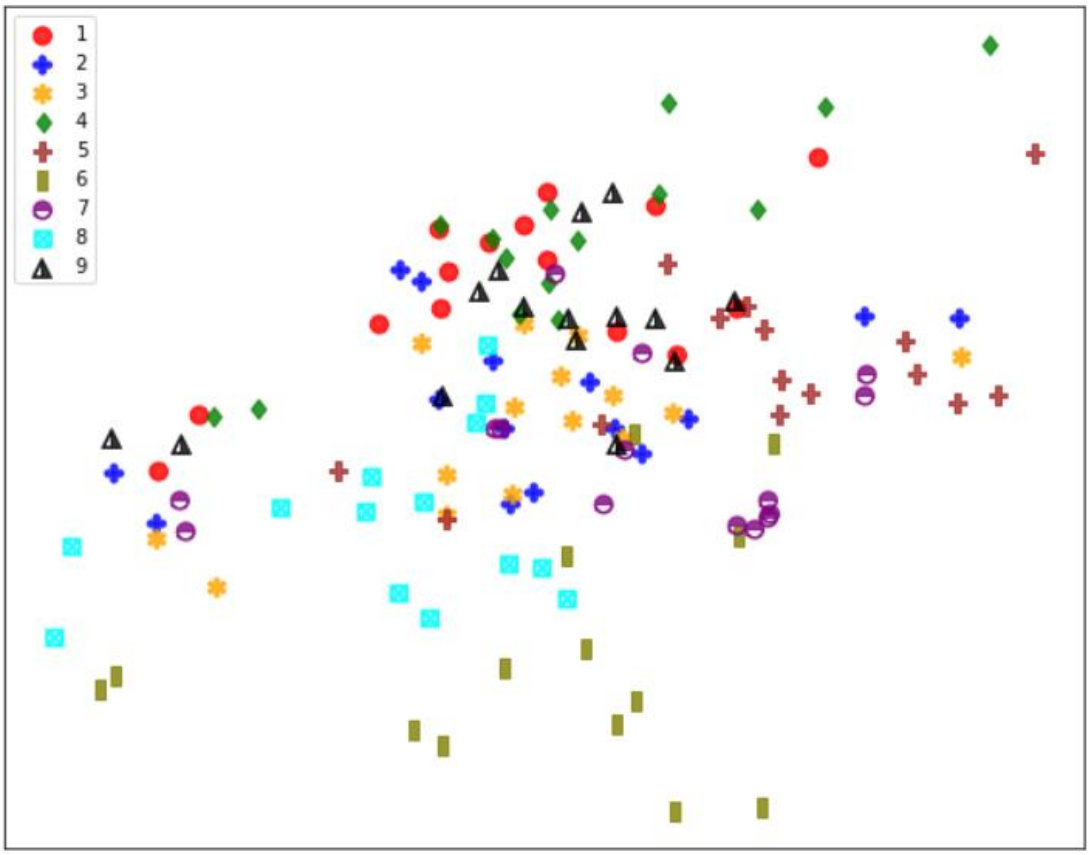
Επομένως σε κάθε μοναδικό διάνυσμα θα έχουμε συνολικά:

- 13 μέσες τιμές που αντιστοιχούν στους 13 συντελεστές MFC
- 13 τιμές που αντιστοιχούν στην τυπική απόκλιση που προκύπτει από τους προηγούμενους δεκατρείς συντελεστές
- 13 μέσες τιμές για την πρώτη παράγωγο delta

- 13 μέσες τιμές για την δεύτερη παράγωγο delta2
- 13 τιμές για την τυπική απόκλιση του delta
- 13 τιμές για την τυπική απόκλιση του delta2

Συνολικά λοιπόν κάθε μοναδικό διάνυσμα θα αποτελείται από 6x13=78 attributes (χαρακτηριστικά-διαστάσεις)

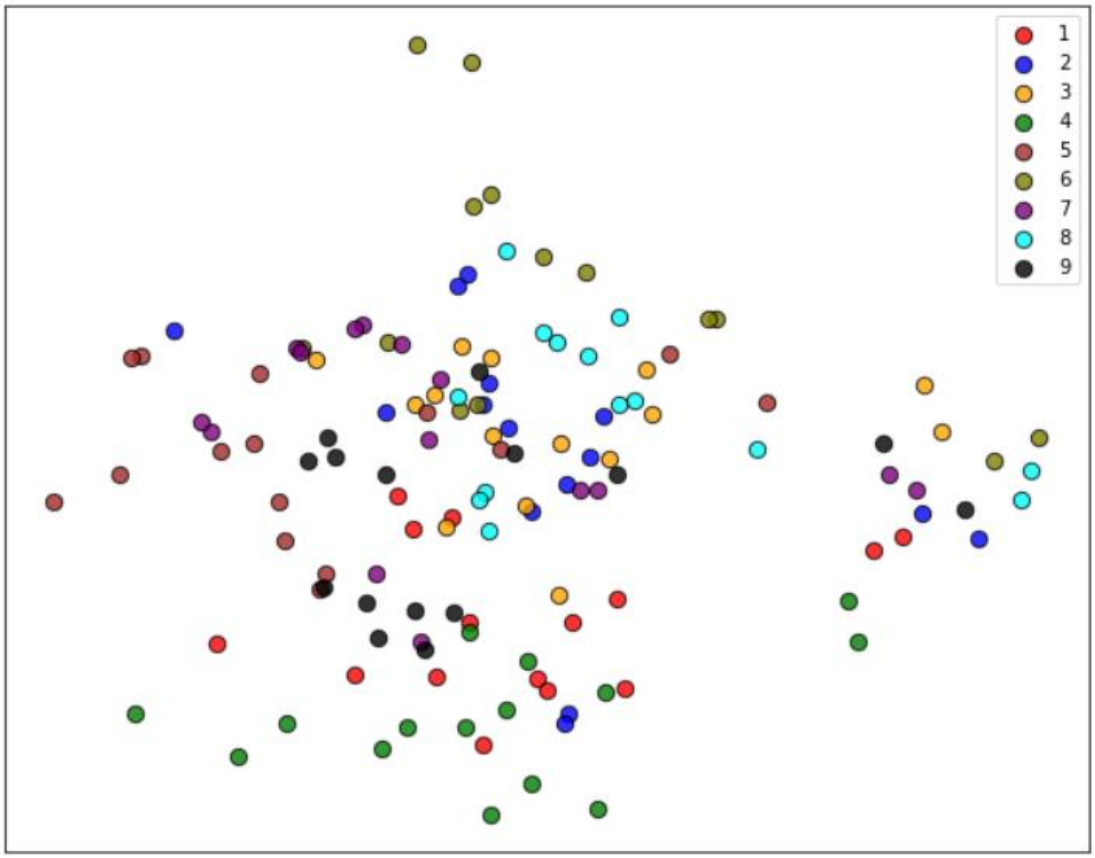
Εισάγοντας τώρα υπό αυτήν την μορφή των μοναδικών διανυσμάτων τα δεδομένα μας στην συνάρτηση scatter προκύπτει η εξής αναπαράσταση για τις 9 κλάσεις (ψηφία 0-9) βάσει των δύο πρώτων διαστάσεων:



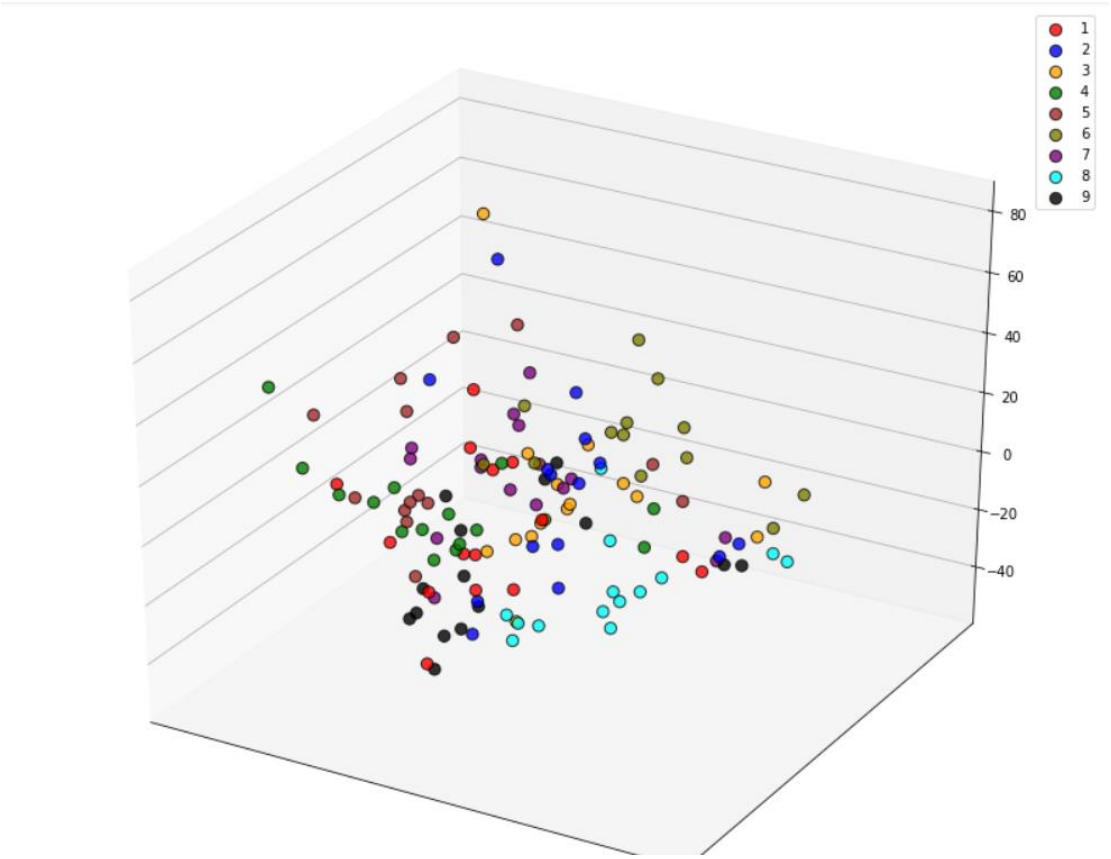
Από την παραπάνω απεικόνιση δεν μπορεί να γίνει εύκολα η διάκριση των 9 κλάσεων μας καθώς υπάρχει μεγάλη επικάλυψη των δειγμάτων. Όπως είχαμε επισημάνει και στο βήμα 4, οι δύο πρώτοι MFCCs που αποτελούν εδώ τις δύο διαστάσεις βάσει των οποίων ταξινομούνται τα δείγματά μας, δεν είναι αντιπροσωπευτικές για την διάκριση των διαφορετικών ψηφίων καθώς οι κατανομές που εμφανίζονται είναι παρόμοιες μεταξύ τους. Χρησιμοποιώντας όλες τις διαστάσεις (και τις 78) αναμένουμε σαφώς πιο ευδιάκριτη αναπαράσταση των δειγμάτων μας καθώς έτσι θα έχουμε περισσότερη πληροφορία που θα ξεχωρίζει το κάθε δείγμα από τα υπόλοιπα.

Βήμα 6

Θέλουμε να απεικονίσουμε τα 78-διάστατα δείγματά μας σε δύο διαστάσεις χωρίς όμως να χάνουμε το βασικό περιεχόμενο της πληροφορίας που περιέχουν. Για τον σκοπό αυτό θα χρησιμοποιήσουμε την Principal Component Analysis (PCA). Το αποτέλεσμα που προκύπτει τώρα είναι το εξής:



Επαναλαμβάνουμε και για 3-διαστάσεις την διαδικασία και έχουμε το τρισδιάστατο scatter plot που φαίνεται παρακάτω:



Όπως μπορούμε να δούμε η απεικόνισή μας είναι σαφώς καλύτερη σε σχέση με αυτήν του βήματος 5 καθώς τώρα αξιοποιούμε όλο το διαφορετικό περιεχόμενο πληροφορίας που προκύπτει από τις 78 διαστάσεις κάθε δείγματος. Στο τρισδιάστατο scatter plot η απεικόνιση είναι ακόμα πιο ευδιάκριτη σε σχέση με το δισδιάστατο.

Για τις διασπορές των κυρίων συνιστωσών και των αρχικών αρχικών 78 χαρακτηριστικών ισχύει οτι ισχύει η σχέση:

$\sigma_{px1}^2 + \sigma_{px2}^2 + \ldots + \sigma_{px78}^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_{78}^2$

Μπορούμε λοιπόν να συμπεράνουμε ότι διατηρώντας τις 2 ή 3 πρώτες κύριες συνιστώσες, που έχουν και τη μεγαλύτερη τιμή διασποράς θα έχουμε μια αρκετά καλή εικόνα για την αρχική διασπορά αφού αυτές αποτελούν και το μεγαλύτερο ποσοστό της.

Βλέπουμε ότι το ποσοστό που μειώνεται η αρχική διασπορά χρησιμοποιώντας τις 2 πρώτες κύριες συνιστώσες είναι περίπου 31%

```
Percentage of 2d reduction over the inital variance is 31.946334587099766%
```

Επομένως περίπου το 69% της αρχικής συνολικής διασποράς διατηρείται χρησιμοποιώντας μόνο αυτούς τους δύο πρώτους όρου και άρα μπορούμε να χαρακτηρίσουμε την PCA που εφαρμόσαμε επιτυχή καθώς διατηρείται το μεγαλύτερο ποσοστό της αρχικής διασποράς.

Βήμα 7

Αρχίζουμε να υλοποιούμε από εδώ και έπειτα το στάδιο της ταξινόμησης. Αρχικά κάνουμε split τα δεδομένα μας σε train και test με αναλογία 70%-30%.

Πλέον έχουμε στη διάθεσή μας ένα dataset. Αυτό, αποτελείται απο attributes (features) και classes (labels). Ουσιαστικά τα attributes του συνόλου δεδομένων μας είναι οι μέσες τιμές και οι αποκλίσεις των MFCCs, delta και delta2 που βρήκαμε προηγουμένως ενώ οι κλάσεις μας είναι τα 9 ψηφία.

Ορίζουμε λοιπόν έναν πίνακα X στον οποίο τοποθετούμε τα attributes και έχουμε επίσης την λίστα digit που δημιουργήσαμε στον data_parser, η οποία περιέχει τα labels κάθε sample.

Ορίζουμε επίσης τον εκτιμητή Bayes που υλοποιήσαμε στο πρώτο εργαστήριο

Ταξινομούμε το dataset χρησιμοποιώντας τους εξής ταξινομητές:

1. Custom NaiveBayes
2. Sklearn NB
3. SVM
4. KNN
5. Logistic Regression (LR)

Επίσης κανονικοποιούμε τα δεδομένα μας πριν την ταξινόμηση.

Τα αποτελέσματα που λαμβάνουμε για τις διαφορετικές αυτές ταξινομήσεις είναι:

```
Custom NB Classifier score is 0.55
Scikit NB Classifier score is 0.55
SVM Classifier score is 0.25
KNN Classifier with 3 neighbors score is 0.5
Logistic Regression Classifier score is 0.25
```

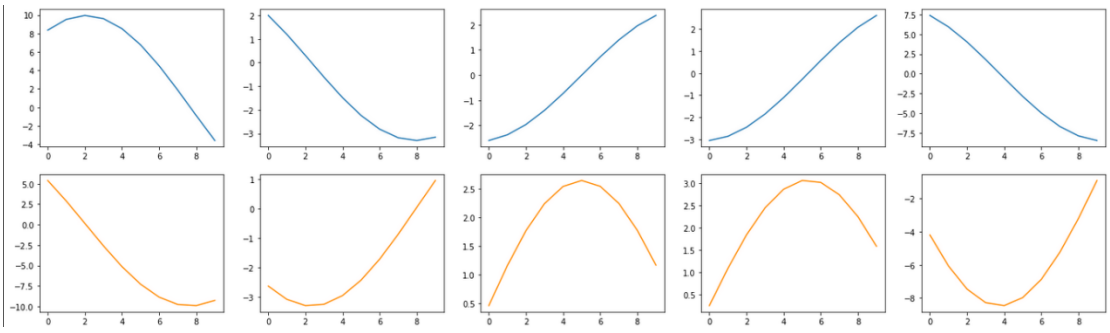
Παρατηρούμε ότι η ακρίβεια των "παραδοσιακών" ταξινομητών είναι αρκετά χαμηλη και μάλιστα ο KNN που όπως έιχαμε δει στα προηγούμενο εργαστήριο

επιτυγχάνει τις καλύτερες επιδόσεις τώρα έχει χαμηλότερη επίδοση ακόμη και από τους NB ταξινομητές.

Επίσης για την υλοποίηση του Logistic Regression χρησιμοποιώντας τις μη κανονικοποιημένες τιμές των δεδομένων επιτυγχάνουμε καλύτερο ποσοστό στο score της τάξης του 80%, κάτι το οποίο έχει να κάνει με την υλοποίηση του ίδιου του ταξινομητή.

Βήμα 8

Στο βήμα αυτό δημιουργούμε αρχικά ακολουθίες 10 σημείων ενός ημιτόνου και ενός συνημιτόνου με συχνότητα f = 40 Hz. Το αποτέλεσμα που παίρνουμε είναι το εξής:



Βήμα 9

Από εδώ και έπειτα θα δουλέψουμε χρησιμοποιώντας το Free-Spoken-Digit Dataset (FSDD). Αφού κατεβάσουμε το βοηθητικό υλικό που μας δίνεται, χωρίζουμε το dataset μας (που περιέχεται στον φάκελο recordings) σε train και test set βάσει του parser.py κώδικα που μας υποδεικνύεται. Έπειτα χωρίζουμε τα train δεδομένα μας σε training και validation set με ποσοστό 80%-20%. Ο διαχωρισμός έγινε χρησιμοποιώντας την συνάρτηση stratified split με την οποία διασφαλίζουμε ότι σε κάθε set περιέχεται το ίδιο πλήθος από κάθε διαφορετικό ψηφίο.

Τυπώνουμε ενδεικτικά τα μεγέθη των set που προκύπτουν μετά τον διαχωρισμό:

```
Number of train samples after parsing: 2700
Number of test samples after parsing: 300
Number of train samples after splitting training set: 2160
Number of validation samples after splitting training set: 540
```

Βήμα 10

Για την αναγνώριση των ψηφίων θα χρησιμοποιήσουμε GMM-HMM (Gaussian Mixtures Models-Hiddeon Markov Models).

Για κάθε ψηφίο (0,1,…,9) αρχικοποιώ ένα GMM-HMM, επομένως θα έχω συνολικά 10 μοντέλα, καθένα από τα οποία θα είναι υπεύθυνο για την αναγνώριση ενός συγκεκριμένου ψηφίου.

Ορίζω το κάθε μοντέλο μου ως εξής:

- Είναι **left-right** δηλαδή μπορώ να μεταβώ από μια κατάσταση μόνο προς κάποια επόμενη κατάσταση που βρίσκεται δεξιά της. Στον πίνακα μεταβάσεων αυτό αναπαρίσταται θέτοντας μηδενικά όλα τα α_{ij} στοιχεία για τα οποία ισχύει i<j
- Συγκεκριμένα **μπορώ να μεταβώ μόνο σε διαδοχική κατάσταση**, δηλαδή μπορώ να κάνω μετάβαση μόνο από την i στην i+1. Αυτό αναπαρίσταται στον πίνακα μεταβάσεων θέτωντας $\alpha_{ij} = 0$ για j>i+1
- Στον πίνακα μεταβάσεων θέτω αρχική πιθανότητα των στοιχείων $\pi_i = 0$ για $i \neq 1$ και $\pi_i = 1$ για $i = 1$

Για κάθε ψηφίο που εκφωνείται, εξάγουμε τόσα χαρακτηριστικά όσα είναι και τα MFCCs που προκύπτουν από όλα τα windowed frames στα οποία χωρίζουμε το αρχικό μας σήμα (όπως είδαμε και στην προπαρασκευή). Σε κάθε window εξάγονται από 6 MFCCs, όμως ο συνολικός αριθμός των windows που "χωράνε" σε κάθε φωνητικό αρχείο είναι διαφορετικός διότι κάθε εκφώνηση έχει διαφορετική χρονική διάρκεια (άλλοι εκφωνητές προφέρουν πιο γρήγορα ή πιο αργά το ίδιο ψηφίο). Έτσι και το διάνυσμα ακουστικών χαρακτηριστικών για κάθε εκφώνηση έχει διαφορετικό μέγεθος και είναι της μορφής [αριθμός windows, 6 MFCCs σε κάθε window]. Κάθε τέτοιο διάνυσμα θεωρείται ως μια πιθανή παρατήρηση σε κάποια κατάσταση. Όπως γίνεται αντιληπτό οι τιμές που μπορεί να έχουν αυτά τα διανύσματα είναι συνεχείς και για τον λόγο αυτό θα χρησιμοποιήσουμε GMM για να μοντελοποιήσουμε τις πιθανότητές τους.

Βήμα 11

Θέλουμε να εκπαιδεύσουμε καθένα από τα δέκα μοντέλα που κατασκευάσαμε παραπάνω χρησιμοποιώντας τον αλγόριθμο Expectation Maximization (EM).

Ορίζουμε 10 επαναλήψεις για τον αλγόριθμο. Για την εκπαίδευση κάθε μοντέλου χρησιμοποιούμε όλα τα διαθέσιμα δεδομένα για το ψηφίο αυτό, δηλαδή όλα τα αρχεία που εκφωνούν το ψηφίο αυτό.

Βήμα 12

Αφού έχει πραγματοποιηθεί η εκπαίδευση του βήματος 11, έχω καταλήξει στις τελικές καλύτερες παραμέτρους για καθένα από τα 10 μοντέλα.

Υπολογίζω έπειτα στο validation set την log likelihood (λογαριθμική πιθανοφάνεια) για όλες τις εκφωνήσεις χρησιμοποιώντας όλες τις παραμέτρους που προέκυψαν. Δηλαδή υπολογίζω 10 λογαριθμικές πιθανοφάνειες για κάθε εκφώνηση, μια για κάθε set παραμέτρων. Εκεί όπου εμφανίζεται η μεγαλύτερη πιθανοφάνεια θεωρώ πως είναι η σωστή κατηγορία στην οποία κατατάσσω την εκφώνηση μου. Αφού ολοκληρώσω την διαδικασία αυτή υπολογίζω το accuracy που προκύπτει.

Για κάθε ένα από τα μοντέλα μου μεταβάλλω λίγο τις παραμέτρους ώστε να αυξήσω την πιθανοφάνεια και κρατάω τελικά τις καλύτερες παραμέτρους .

Συγκεκριμένα δοκιμάζω τιμές για τα states από 1 εως 5, για τα Gaussian mixture από 1 έως 6 και για αριθμό επαναλήψεων (iterations) από 5 εως 30 με βήμα 5. Το αντίστοιχο μοντέλο που προκύπτει για αυτές τις παραμέτρους το χρησιμοποιώ τώρα για το test set.

Με την παραπάνω διαδικασία καταφέρνω να αυξήσω τις επιμέρους λογαριθμικές πιθανοφάνειες και τελικά την συνολική log likelihood. Αυτό θα έχει ως αποτέλεσμα να επιτευχθεί καλύτερο accuracy. Πέραν αυτού, με την διαδικασία αυτή έχουμε αξιοποιήσει τελικά όλα τα διαθέσιμα δεδομένα ώστε τελικά εκπαιδεύουμε καλύτερα το μοντέλο μας αποφεύγοντας το ενδεχόμενο του overfitting

Βήμα 13

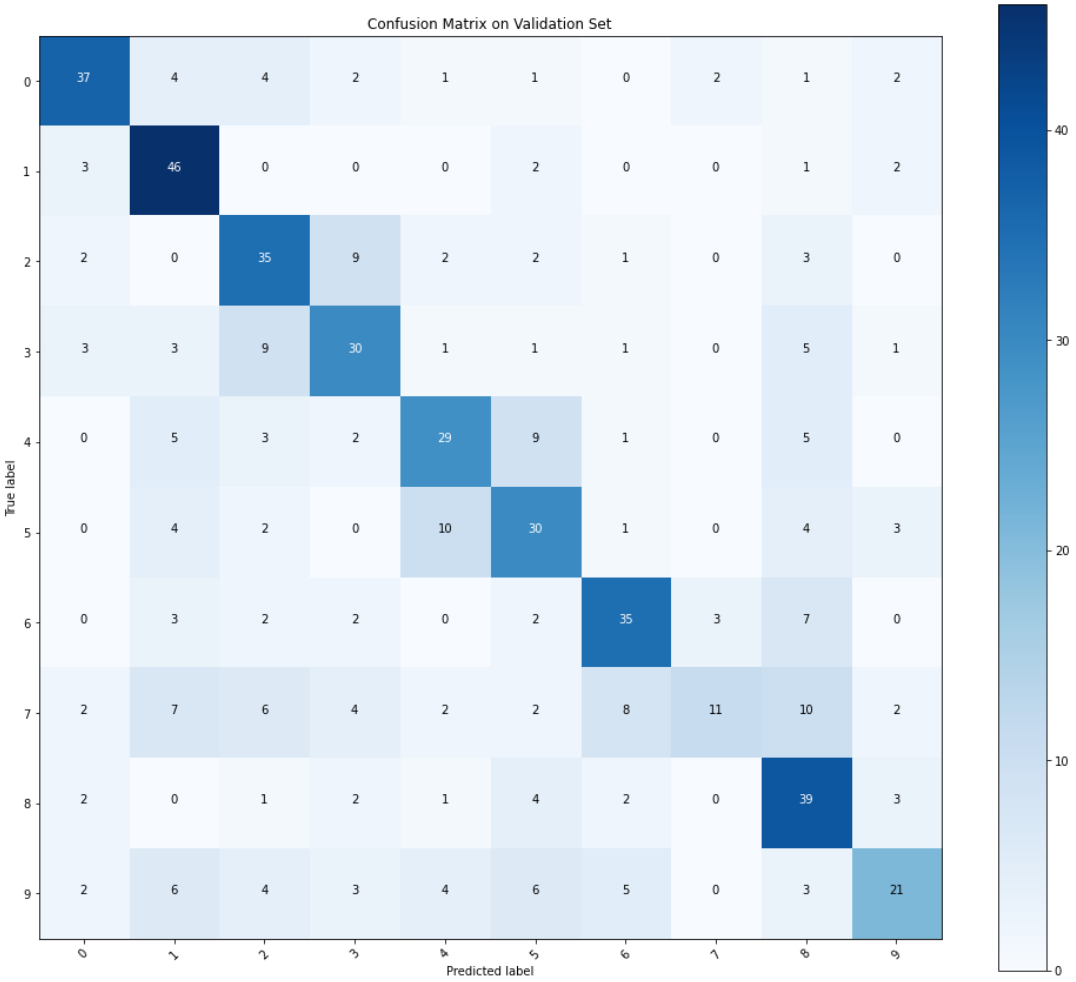
Δημιουργώ έναν Confusion Matrix για καθένα από τα validation και test set διαστάσεων 10x10 στον οποίο αποθηκεύω τον αριθμό των προβλέψεων για κάθε ψηφίο δηλαδή:

- Η κάθε γραμμή οριζόντια αναπαριστά τον αριθμό των ψηφίων που κατατάχθηκαν σε κάθε κατηγορία
- Η κάθε στήλη αναπαριστά την κατηγορία στην οποία κατατάχθηκαν τα δείγματα.

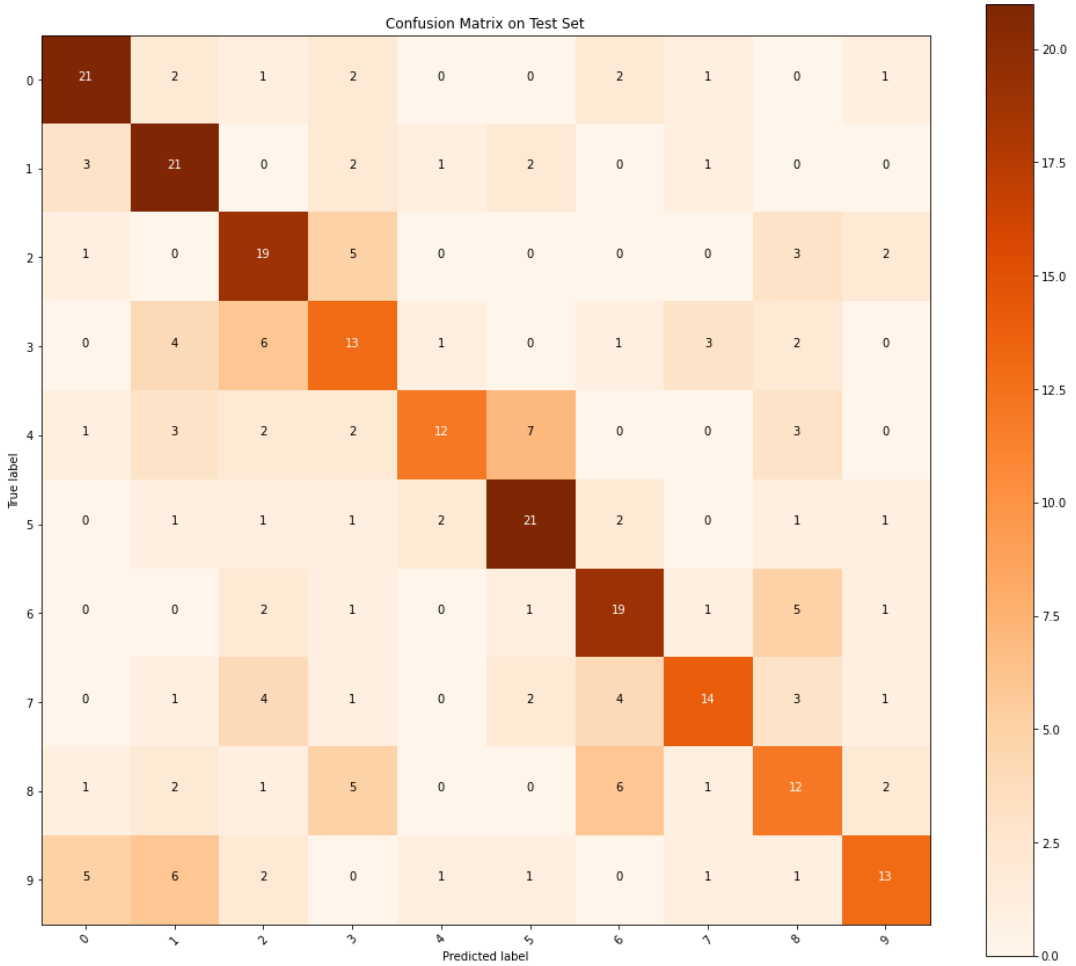
Έτσι λοιπόν αναμένουμε στα διαγώνια στοιχεία του πίνακα αυτού να έχουμε τις μεγαλύτερες συγκεντρώσεις προβλέψεων, διότι αυτές μας δείχνουν το πλήθος των σωστών προβλέψεων.

Πράγματι έχουμε τις εξής εικόνες να προκύπτουν:

- Για το validation set



- Για το test set:



Το συνολικό accuracy προκύπτει από τα άθροισμα των διαγώνιων στοιχείων α_{ii} του πίνακα προς το άθροισμα όλων των στοιχείων του πίνακα.