**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

# Υποστήριξη Ψηφιακής Επεξεργασίας Ομιλίας και Μουσικής από την Python

**Δημήτριος Ν. Αναστασόπουλος**

**Επιβλέπων:** **Γεώργιος Κουρουπέτρογλου,** Καθηγητής

**ΑΘΗΝΑ**

**ΣΕΠΤΕΜΒΡΙΟΣ 2020**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

# Υποστήριξη Ψηφιακής Επεξεργασίας Ομιλίας και Μουσικής από την Python

**Δημήτριος Ν. Αναστασόπουλος**
**Α.Μ.:** 1115201100019

**ΕΠΙΒΛΕΠΟΝΤΕΣ:** **Γεώργιος Κουρουπέτρογλου,** Καθηγητής

# ΠΕΡΙΛΗΨΗ

Η εργασία αυτή έχει ως θέμα την υποστήριξη ψηφιακής επεξεργασίας ομιλίας και μουσικής από την προγραμματιστική γλώσσα Python. Σκοπός της εργασίας είναι να αναλυθούν και να παρουσιαστούν όλοι οι διαθέσιμοι τρόποι που παρέχονται από την συγκεκριμένη γλώσσα, για την επίτευξη της επεξεργασίας ήχου σε αυτούς τους δυο τομείς.

Το αντικείμενο της μελέτης χωρίζεται σε δύο μέρη, ένα για την κάθε ανεξάρτητη περιοχή ανάλυσης. Στο πρώτο μέρος παρουσιάζονται οι βιβλιοθήκες που αφορούν την επεξεργασία ομιλίας. Πιο συγκεκριμένα αναλύονται λογισμικά της python για εφαρμογές αναγνώρισης φωνής, ανάλυσης φάσματος συχνοτήτων της ομιλίας και μετατροπή γραπτού λόγου σε ψηφιακή φωνή.

Στο δεύτερο μέρος παρουσιάζονται οι βιβλιοθήκες που αφορούν την επεξεργασία μουσικής σε τομείς όπως η σύνθεση, η ανάλυση καθώς και η παραγωγή διαφόρων ψηφιακών τύπων μουσικής.

Η μεθοδολογία που ακολουθείται περιλαμβάνει την δοκιμή κάθε δωρεάν διαθέσιμου λογισμικού, όσων αφορά τις δυνατότητες, τα προαπαιτούμενα και την τεκμηρίωση του, την παρουσίαση αυτών των ιδιοτήτων και στην συνέχεια την σύγκριση των αποτελεσμάτων για την εξαγωγή του συμπεράσματος. Στα πλαίσια της σύγκρισης διερευνάται ποια από τα επικείμενα λογισμικά / βιβλιοθήκες παρέχουν μεγαλύτερη ευκολία και δυνατότητες στον χρήστη για την επίτευξη της ψηφιακής επεξεργασίας ομιλίας και μουσικής.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ**: Επεξεργασία Ήχου και Μουσικής

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**: Συχνότητα, σήμα, αναγνώριση, ψηφιακή, υπολογιστής, πρόγραμμα

# ABSTRACT

This assignment is about programming language Python and its support on digital sound processing, more specifically on music and speech. The purpose of this task is to analyze and present all available ways provided by that language to achieve audio processing in these two areas.

The subject of the study is divided into two parts, one for each independent analysis area. The first part presents the libraries for speech processing. Python's software is analyzed in voice recognition applications, speech frequency spectrum analysis, and text-to-speech applications.

The second part presents libraries related to music processing in areas such as composition, analysis and production of various digital formats.

The methodology followed includes the downloading and testing of all free available software, in terms of capabilities, prerequisites and documentation, presenting these properties and then comparing the results to draw the conclusion. The comparison derives which of the tested software / libraries provide the user with greater convenience and features to achieve digital speech and music processing.



**SUBJECT AREA**: Sound and Music processing

**KEYWORDS**: Frequency, signal, recognition, digital, computer, program

# ΠΕΡΙΕΧΟΜΕΝΑ

# ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

# ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

# ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

# ΠΡΟΛΟΓΟΣ

Η εργασία αυτή πραγματοποιήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών του τμήματος Πληροφορικής και Τηλεπικοινωνιών στο Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών.

Αθήνα, Σεπτέμβριος 2020

# 1. ΕΙΣΑΓΩΓΗ

## 1.1 Εισαγωγή στην Python

Η Python είναι μια γλώσσα προγραμματισμού που δημιουργήθηκε από τον Guido van Rossum το 1989. Πρωτίστως, χρησιμοποιήθηκε για το λειτουργικό του κατανεμημένου συστήματος Amoeba. Η έκδοση 2.0 κυκλοφόρησε το 2000, ενώ η τρίτη έκδοση το 2008. Η Python 3.0 είναι προς τα πίσω συμβατή και αυτό την καθιστά την πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις. Σήμερα, η Python είναι ευρέως γνωστή και εδραιωμένη σε τομείς όπως ο διαδικτυακός προγραμματισμός, οι επιστημονικές εφαρμογές αλλά και στον τομέα της εκπαίδευσης λόγω της απλότητας της.

Η Python είναι διερμηνευόμενη (interpreted) γλώσσα το οποίο σημαίνει ότι χρησιμοποιεί διερμηνέα για την εκτέλεση του κώδικα της. Η διαφορά με τις γλώσσες προγραμματισμού που πρώτα μεταγλωττίζονται σε γλώσσα μηχανής είναι ότι ο κώδικας μιας διερμηνευόμενης γλώσσας εκτελείται γραμμή-γραμμή μέσω του διερμηνέα. Αυτό έχει ως αποτέλεσμα την πιο αργή εκτέλεση προγραμμάτων σε σχέση με τις γλώσσες που χρησιμοποιούν μεταγλωττιστές, αλλά παρέχει την δυνατότητα φορητότητας του κώδικα ανεξαρτήτως πλατφόρμας εκτέλεσης. Επίσης, καθώς ο διερμηνέας μπορεί να καλέσει τις ίδιες βιβλιοθήκες με έναν μεταγλωττιστή, οι χρονοβόροι αριθμητικοί αλγόριθμοι δεν εφαρμόζονται στην ίδια την γλώσσα, αλλά καλούνται από μεταγλωττισμένες βιβλιοθήκες. Αυτό συνεπάγεται στην ευέλικτη, γρήγορη και αποδοτική ανάπτυξη προγραμμάτων με την χρήση των κατάλληλων βιβλιοθηκών.

Όσον αφορά την σύνταξη του κώδικα, η Python διαθέτει απλή και ευανάγνωστη γραφή, απαλλαγμένη από τα συντακτικά σύμβολα και τις σύνθετες ιδιομορφίες που άλλες γλώσσες απαιτούν. Αυτό την καθιστά εύχρηστη, εύκολη και ευχάριστη στην μάθηση αλλά ταυτόχρονα παρέχει στους προγραμματιστές που την χρησιμοποιούν ταχύτητα και τουτέστιν προάγει την παραγωγικότητα κώδικα για την ανάπτυξη μεγάλων εφαρμογών.

Ένα σημαντικό χαρακτηριστικό της Python που σχετίζεται έμμεσα με την επικείμενη επιστημονική μελέτη, είναι ότι αποτελεί γλώσσα ανοικτού κώδικα (open source). Ο κώδικάς της δηλαδή είναι διαθέσιμος χωρίς κόστος στο ευρύ κοινό. Το λογισμικό ανοικτού κώδικα μπορεί ο καθένας ελεύθερα να το χρησιμοποιεί, να το αντιγράφει, να το διανέμει και να το τροποποιεί ανάλογα με τις ανάγκες του. Το ίδιο ισχύει και για τις βιβλιοθήκες που παρέχονται για την Python, οι οποίες είναι ανοικτού κώδικα και αναπτύσσονται από την κοινότητα που έχει διαμορφωθεί γύρω από την γλώσσα. Λόγω της μη κερδοσκοπικής φύσης της γλώσσας αλλά και πιο γενικευμένα της εξάπλωσης των open source projects η Python είναι πλέον μια από τις πιο δημοφιλείς και μαζικά υποστηριζόμενες γλώσσες προγραμματισμού. Οι βιβλιοθήκες που θα παρουσιαστούν στα πλαίσια της εργασίας ανήκουν σε αυτό το φάσμα ανοικτού κώδικα και διανέμονται χωρίς κόστος από την κοινότητα της Python.

## 1.2 Python και Ψηφιακή Επεξεργασία Σήματος

Στον τομέα της ψηφιακής επεξεργασίας σημάτων, το κύριο προτέρημα της Python σε σχέση με άλλες γλώσσες προγραμματισμού, είναι ότι παρέχει χωρίς κόστος, πληθώρα βιβλιοθηκών για επιστημονικούς υπολογισμούς. Το πιο γνωστό εργαλείο σε αυτόν τον τομέα είναι η MATLAB, η οποία επιτρέπει στον χρήστη να επεξεργαστεί αριθμητικά δεδομένα στην μορφή πινάκων και περιλαμβάνει διάφορες υλοποιήσεις για την ΨΕΣ. Ωστόσο, η MATLAB δεν αποτελεί δωρεάν λογισμικό, κάτι το οποίο περιορίζει την ανοικτή συνεργασία μεταξύ ερευνητών της επιστημονικής κοινότητας.

Σε αντίθεση, για την ανάπτυξη και σχεδίαση εφαρμογών επεξεργασίας σημάτων, η Python αποτελεί μια ευρέως αφομοιωμένη από την κοινότητα γλώσσα, που διαθέτει συνεχή υποστήριξη και ευελιξία ανάμεσα στα υπάρχοντα λειτουργικά συστήματα. Λόγω του διερμηνέα της Python, οι εντολές εκτελούνται σε runtime και τα αποτελέσματα των υπολογισμών προβάλλονται στον χρήστη πρόωρα, προάγοντας έτσι την ταχύρρυθμη παραγωγή κώδικα.

Τα πιο διαδεδομένα εργαλεία της Python για την ΨΕΣ είναι το **NumPy**, το **Matplotlib** και το **SciPy**.

- Το NumPy είναι μια βιβλιοθήκη που εισάγει ένα αντικείμενο πολυδιάστατου πίνακα και διάφορες συναρτήσεις για αριθμητικούς υπολογισμούς πάνω σε πίνακες. Είναι γραμμένο σε C και περιλαμβάνει εργαλεία για ενσωμάτωση προγραμμάτων Fortran και C++.

- Το Matplotlib είναι μια βιβλιοθήκη για την δισδιάστατη απεικόνιση γραφημάτων, γραφικών παραστάσεων και σχημάτων στην Python. Συνεργάζεται με το NumPy καθώς μπορεί να δεχτεί ως είσοδο τους πίνακες που αυτό παράγει.

- Το SciPy αποτελεί ένα οικοσύστημα βιβλιοθηκών για μαθηματικά και φυσικές επιστήμες. Βασίζεται στους πίνακες που παρέχει το NumPy και διαθέτει συναρτήσεις για την επεξεργασία σήματος, όπως μετασχηματισμό Fourier, υλοποιήσεις φίλτρων FIR / IIR και συνέλιξη. Επίσης, συμπεριλαμβάνει μεθόδους για την E/E αρχείων, δίνοντας την δυνατότητα για επεξεργασία ηχογραφημένων ηχητικών και για την εξαγωγή ηχητικών αρχείων σε διάφορα format.

Με την χρήση αυτών των εργαλείων της Python επιτυγχάνεται η ανάπτυξη εφαρμογών για την ψηφιακή επεξεργασία σημάτων, με ευκολία, επαρκή τεκμηρίωση και ταχύτητα.

Στο διαδίκτυο διατίθενται πολλές ακόμα βιβλιοθήκες με το ίδιο αντικείμενο αλλά και πιο εξειδικευμένη χρησιμότητα.

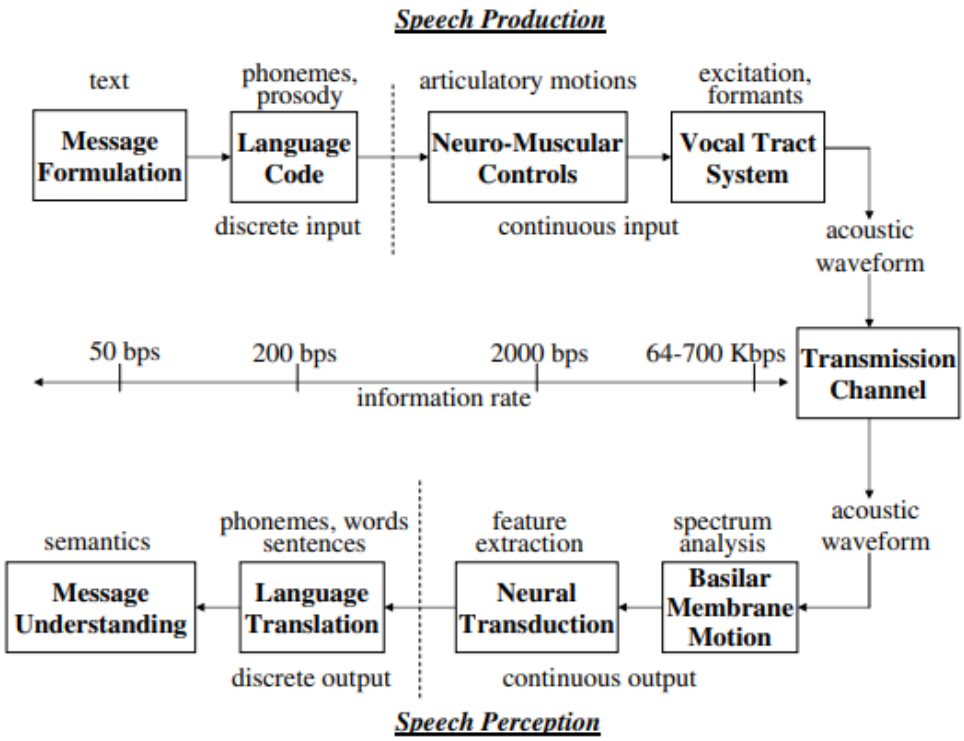
Στην παρούσα εργασία παρουσιάζονται οι βιβλιοθήκες στους συγκεκριμένους τομείς της ΨΕΣ δηλαδή στην επεξεργασία **ομιλίας** και **μουσικής**.

# 2. ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΟΜΙΛΙΑΣ

## 2.1 Εισαγωγή

Η επεξεργασία ομιλίας είναι ένας από τους πιο διαδεδομένους κλάδους της επεξεργασίας ψηφιακού σήματος. Αφορά την μετατροπή της ακουστικής κυματομορφής της ανθρώπινης ομιλίας σε μια ψηφιακή ακολουθία αριθμών με σκοπό την περαιτέρω επεξεργασία της. Βασικές εφαρμογές της ψηφιακής επεξεργασίας ομιλίας είναι η ψηφιακή **κωδικοποίηση** και **συμπίεση** ενός φωνητικού μηνύματος, η **αναγνώριση** ομιλίας καθώς και η **μετατροπή** γραπτού κειμένου σε φωνητικό (text-to-speech).

Η δυσκολία στην επεξεργασία του σήματος της ομιλίας έγκειται στην πολύπλοκη και σύνθετη δομή του. Επίσης, τα φωνητικά σήματα διαφέρουν τόσο σε θέμα συχνότητας όσο και σε θέμα χρονικής δειγματοληψίας. Αυτό σημαίνει ότι από άνθρωπο σε άνθρωπο η δομή του σήματος αλλάζει ακόμα και αν το σήμα αφορά την ίδια πληροφορία. Αυτό μπορεί να οφείλεται στην διαφορετική χροιά μεταξύ των διαφόρων φωνών αλλά και στην ταχύτητα αποτύπωσης του μηνύματος. Καθώς ένα φωνητικό σήμα είναι μια συνεχής κυματομορφή, εκτός από το φασματικό περιεχόμενο συχνοτήτων, το σήμα εξαρτάται σε μεγάλο βαθμό και από την χρονική του υπόσταση.




**Σχήμα 1: Διάγραμμα διαδικασίας παραγωγής και αντίληψης ομιλίας**

Οι πιο συνηθισμένοι μετατροπείς αναλογικού σήματος της φωνής σε ψηφιακό λειτουργούν ως εξής. Δειγματοληπτούν το σήμα της φωνής με μεγάλη συχνότητα δειγματοληψίας, εφαρμόζουν ένα κατωδιαβατό φίλτρο για να περιορίσουν το εύρος συχνοτήτων του σήματος εξόδου και στη συνέχεια μειώνουν τη συχνότητα δειγματοληψίας στην επιθυμητή, η οποία μπορεί να είναι τόσο χαμηλή όσο η διπλή συχνότητα αποκοπής του ψηφιακού φίλτρου (Nyquist).

## 2.2 Υποστήριξη Αναγνώρισης Ομιλίας από την Python

Ένας από τους βασικούς κλάδους της επεξεργασίας ομιλίας είναι η δυνατότητα αναγνώρισης πληροφοριών στον προφορικό λόγο. Η αναγνώριση ομιλίας περιλαμβάνει εφαρμογές όπως την αποτύπωση του συνεχούς φωνητικού λόγου σε γραπτό (π.χ. μεταφραστής), την αναγνώριση και κατ' επέκταση την ταυτοποίηση ενός ομιλητή, καθώς και την εύρεση συγκεκριμένων πληροφοριών (λέξεων) μέσα σε ένα φωνητικό μήνυμα.

Το κυριότερο πλεονέκτημα της αναγνώρισης ομιλίας είναι η ευκολία που παρέχει στον ομιλητή καθώς δεν απαιτεί την εξειδικευμένη ικανότητα για δακτυλογράφηση του μηνύματος. Αυτό το πλεονέκτημα προβάλει σημαντική βοήθεια σε άτομα που χρήζουν προσβασιμότητας και για αυτό οι σύγχρονες συσκευές συμπεριλαμβάνουν την αναγνώριση φωνής σε τέτοιου είδους εφαρμογές.

Ένα δεύτερο πλεονέκτημα της αναγνώρισης ομιλίας είναι η ταχύτητα αποτύπωσης του προφορικού λόγου. Η εισαγωγή πληροφοριών είναι 3 – 4 φορές πιο γρήγορη με την χρήση φωνητικής αναγνώρισης. Σε εφαρμογές όπως ένας ψηφιακός μεταφραστής ή στην αποθήκευση ενός λόγου ως έγγραφο κειμένου αυτό παίζει καθοριστικό ρόλο καθώς επιτρέπει την συνεχή ροή της ομιλίας. Η εισαγωγή πληροφοριών με ομιλία δύναται να λειτουργήσει και σε καταστάσεις θορύβου ή όταν ο χρήστης μετακινείται, προάγοντας την παραγωγικότητα του ανθρώπου ταυτόχρονα με άλλες δραστηριότητες.

Η αναγνώριση ομιλίας παρουσιάζει τις γενικευμένες δυσκολίες της επεξεργασίας της φωνής, λόγω της σύνθετης δομής του ηχητικού σήματος, αλλά εισάγει και νέες δυσκολίες ως προς την υλοποίηση της. Αυτές οι δυσκολίες αφορούν κυρίως τις γλωσσολογικές και λεξιλογικές ιδιότητες στην αποτύπωση μηνυμάτων.

Είναι απαραίτητο να προσδιοριστεί ποια γλώσσα χρησιμοποιείται από τον ομιλητή. Αφ' ενός παρουσιάζονται ομοιότητες μεταξύ λέξεων με διαφορετική σημασία σε διάφορες γλώσσες και αφ' ετέρου η βάση δεδομένων περιορίζεται σε πολύ μεγάλο βαθμό όταν η γλώσσα αναζήτησης είναι προκαθορισμένη. Επιπροσθέτως στο λεξιλόγιο μιας γλώσσας προστίθενται συνεχώς καινούργιες λέξεις. Σε κάποιες γλώσσες ο τονισμός επηρεάζει την σημαντικότητα των λέξεων και το είδος των προτάσεων. Για παράδειγμα μια ερώτηση στα ελληνικά, εφόσον ο προφορικός λόγος δεν περιλαμβάνει ρητά σημεία στίξης, εξαρτάται κυρίως από τον τονισμό της πρότασης. Αυτό εισάγει επιπρόσθετη πολυπλοκότητα στην ψηφιακή αποτύπωση του μηνύματος.

Συμπερασματικά, στα κύρια μειονεκτήματα της αναγνώρισης ομιλίας συμπεριλαμβάνονται, η δυσκολία ανίχνευσης των γλωσσολογικών ιδιαιτεροτήτων όσον αφορά την στίξη, η αφαίρεση του θορύβου καθώς και προβλήματα διαφοροποίησης μεταξύ γλωσσών.

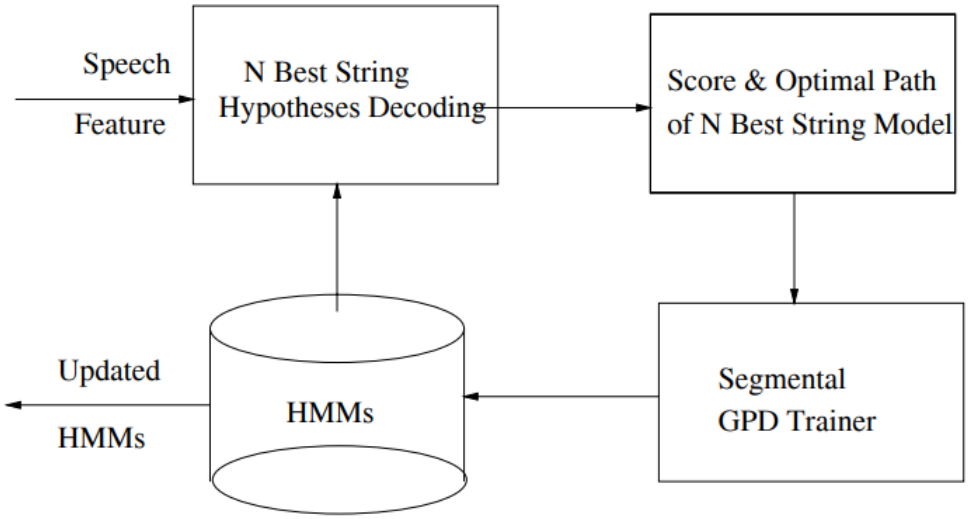
Η διαδικασία για την επίτευξη της αναγνώρισης ομιλίας περιλαμβάνει αρχικά την μετατροπή του αναλογικού σήματος σε ψηφιακό. Στην συνέχεια απαιτείται η κατάτμηση του δειγματοληπτημένου σήματος και η κατηγοριοποίηση του σε ένα σύνολο από διανύσματα παραμέτρων. Οι παράμετροι του διανύσματος έχουν να κάνουν με τις συχνοτικές αλλά και τις χρονικές ιδιότητες του παραγόμενου σήματος. Εφόσον το σήμα κατηγοριοποιηθεί σε διανύσματα πρέπει να εφαρμοστεί κάποιος αλγόριθμος για το ταίριασμα του δείγματος με τις κατάλληλες λέξεις. Οι βασικές μέθοδοι που χρησιμοποιούνται συνήθως στις εφαρμογές αναγνώρισης ομιλίας είναι είτε ντετερμινιστικές η στοχαστικές.

Στην πρώτη περίπτωση συχνά χρησιμοποιείται Δυναμική Χρονική Στρέβλωση του δειγματοληπτημένου σήματος. Στην πράξη, εφαρμόζονται αλγόριθμοι εύρεσης πλησιέστερου γείτονα μεταξύ των χρονικών ακολουθιών διανυσμάτων του επεξεργασμένου σήματος εισόδου και άλλων αποθηκευμένων συνόλων διανυσμάτων, με

στόχο την σύγκριση και ταυτοποίηση του μηνύματος. Η έξοδος του συστήματος περιλαμβάνει τις λέξεις που ταιριάζουν περισσότερο με τα δεδομένα της βάσης.

Στην δεύτερη περίπτωση της στοχαστικής μεθόδου, χρησιμοποιούνται μέθοδοι όπως κρυφές αλυσίδες Markov και τεχνητά νευρωνικά δίκτυα. Σε αυτή την περίπτωση η μοντελοποίηση του σήματος αποτελείται από έναν αριθμό καταστάσεων που αναπαριστούν την χρονική πρόοδο μιας λέξης, από την αρχική κατάσταση, και μέσω όλων των φωνημάτων, στην τελική κατάσταση. Αυτή η μέθοδος δεν έχει μια συγκεκριμένη έξοδο αλλά πιθανοτικά πολλές εξόδους για κάθε κατάσταση.

Επειδή ο αριθμός των ακολουθιών καταστάσεων είναι πολύ μεγάλος για μια πρόταση λέξεων, η στοχαστική μέθοδος προτάσσει αρκετά μεγάλη υπολογιστική πολυπλοκότητα και απαιτεί βελτιστοποίηση σε επίπεδο προγραμματισμού. Παρ' όλα αυτά, είναι η μέθοδος που έχει καθιερωθεί στις σύγχρονες εφαρμογές αναγνώρισης ομιλίας λόγω των επιτυχημένων αποτελεσμάτων της.




**Εικόνα 1: Μοντέλο εξαγωγής ιδιοτήτων με αλυσίδες Markov**

### 2.2.1 SpeechRecognition

Περιγραφή:

Το **SpeechRecognition** είναι μια βιβλιοθήκη αναγνώρισης ομιλίας στην Python. Αποτελεί την πιο διαδεδομένη βιβλιοθήκη στον συγκεκριμένο τομέα και υποστηρίζει offline λειτουργικότητα αλλά και την χρήση διαφόρων online API.

Για την offline λειτουργία απαιτεί την χρήση του πακέτου CMU Sphinx.

Οι βασικότερες online μηχανές / διεπαφές αναγνώρισης ομιλίας που υποστηρίζονται από το SpeechRecognition είναι το Google Cloud Speech, το Microsoft Azure Speech και το IBM Speech to Text.

Δίνει την δυνατότητα αναγνώρισης ομιλίας από συνεχόμενη ροή ήχου μέσω του μικρόφωνου αλλά και την εισαγωγή αρχείων ήχου από τον χώρο αποθήκευσης.

Προαπαιτούμενα:

Για την εκτέλεση των μεθόδων της βιβλιοθήκης που χρησιμοποιούν το μικρόφωνο του υπολογιστή απαιτείται η εγκατάσταση του **pyAudio**. Το pyAudio μετά την ολοκλήρωση του εγκαθιστά το πακέτο Visual C++ Build Tools.

URL πακέτου: https://github.com/Uberi/speech_recognition#readme

Παράδειγμα αναγνώρισης ομιλίας από ηχογραφημένο αρχείο .wav

```
# Εισαγωγή της βιβλιοθήκης
import speech_recognition as sr
# Ορισμός του αντικειμένου αναγνώρισης
r = sr.Recognizer()
# Ορισμός αρχείου ήχου
audio_file = sr.AudioFile('geia.wav')
# Αναγνώριση ομιλίας
with audio_file as source:
  r.adjust_for_ambient_noise(source)
  audio = r.record(source)
result = r.recognize_google(audio, language='el-GR')
# Εξαγωγή αποτελέσματος
with open('output.txt',mode ='w') as file:
  file.write("Recognized text:")
  file.write(result)
  print("ready!")
```

Ο παραπάνω κώδικας δέχεται σαν είσοδο ένα αρχείο ήχου .wav το οποίο περιέχει την φράση «Γειά σου τι κάνεις;». Στο τέλος της εκτέλεσης παράγεται ένα αρχείο κειμένου .txt με τα αποτελέσματα της αναγνώρισης ομιλίας.

### 2.2.2 Wit

Περιγραφή:

Το **pywit** είναι το SDK της Python για το Wit.ai. Το wit.ai είναι μια διεπαφή για εφαρμογές αναγνώρισης φυσικής ομιλίας και τεχνητής νοημοσύνης. Δημιουργήθηκε το 2013 από το Facebook και είναι δωρεάν. Δίνει την δυνατότητα σε προγραμματιστές να δημιουργήσουν bots που επικοινωνούν είτε με γραπτό η προφορικό λόγο. Το API που παρέχεται μέσω της διαδικτυακής πλατφόρμας παρέχει πληροφορίες και στατιστικά σχετικά με τις προτάσεις / μηνύματα που έχουν αναγνωριστεί από την εφαρμογή. Επίσης δίνεται η δυνατότητα στον χρήστη να εκπαιδεύσει την εφαρμογή. Μπορεί να αντιστοιχίσει πιθανές λέξεις η προτάσεις με διαφορετική έννοια και έτσι να εισάγει τεχνητή νοημοσύνη στις απαντήσεις του bot.

Οι βασικές ιδιότητες της διεπαφής είναι η αναγνώριση ομιλίας σε διάφορες γλώσσες, η κατανόηση του λόγου με χρήση τεχνητής νοημοσύνης, η προβολή στατιστικών σχετικά με τις ομιλίες που καταγράφηκαν καθώς και η μηχανική εκμάθηση του συστήματος.

Προαπαιτούμενα:

Για την χρήση του SDK απαιτείται η δημιουργία λογαριασμού στην διαδικτυακή πλατφόρμα του wit.ai και η απόκτηση ενός κλειδιού πρόσβασης ( access token) για την εφαρμογή.

URL πακέτου: https://github.com/wit-ai/pywit

URL διεπαφής: https://wit.ai/

Παράδειγμα χρήσης της βιβλιοθήκης για την αναγνώριση ηχητικού αρχείου:from wit import Wit

```
from wit import Wit
resp = None
client = Wit("YTZDFQRQCLNABC2ASGZXRQEXVCBYV4PF")
with open('test-en.wav.wav', 'rb') as f:
  resp = client.speech(f, {'Content-Type': 'audio/wav'})
print('Yay, got Wit.ai response: ' + str(resp))
```

Αποτέλεσμα:

```
Yay, got Wit.ai response:
 {'entities': {}, 'intents': [], 'text': 'hello how are you doing', 'traits': {}}
```

### 2.2.3 Pocketsphinx

Περιγραφή:

Το Pocketsphinx είναι μια διεπαφή της Python για την αναγνώριση ομιλίας. Είναι μέρος του εργαλείου CMU Sphinx και χρησιμοποιεί την βάση δεδομένων του.

Υποστηρίζει την αναγνώριση πολλαπλών γλωσσών με την λήψη του αντίστοιχου λεξικού. Τα λεξικά είναι αρχεία της μορφής .dict και υπάρχουν διαθέσιμα στην ιστοσελίδα του cmusphinx.

Οι βασικές λειτουργίες που υποστηρίζει είναι:

- Η αναγνώριση συνεχούς ομιλίας από αρχείο αλλά και το μικρόφωνο
- Η αναζήτηση keyword στην ομιλία
- Η δημιουργία χρονικών μικρογραφιών για κάθε λέξη σε ένα αρχείο ομιλίας

Προαπαιτούμενα:

Το pocketsphinx έχει ως προαπαιτούμενη βιβλιοθήκη το wheel, το setup-tools και το swig.

URL πακέτου: https://github.com/bambocher/pocketsphinx-python

Παράδειγμα αναγνώρισης ομιλίας από το μικρόφωνο με το pocketsphinx:

```
from pocketsphinx import LiveSpeech
for phrase in LiveSpeech(): print(phrase)
```

Παράδειγμα εύρεσης keyword:

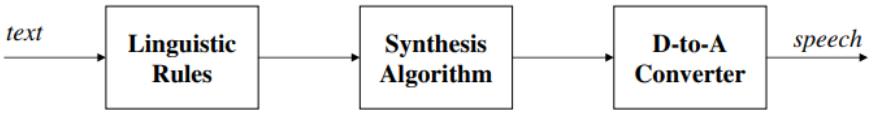
```
from pocketsphinx import LiveSpeech
speech = LiveSpeech(lm=False, keyphrase='test', kws_threshold=1e-20)
for phrase in speech:
    print(phrase.segments(detailed=True))
```

Αποτέλεσμα:

```
[('test', -1930, 162348, 162363)]
[('test', -1876, 868643, 868727)]
[('test', -1646, 1180561, 1180644)]
[('test', -1658, 5411033, 5411067)]
```

## 2.3 Υποστήριξη μετατροπής κειμένου σε ομιλία από την Python

Η σύνθεση ομιλίας (text-to-speech) περιγράφει την διαδικασία μετατροπής μιας σειράς από σύμβολα σε ένα ακουστικό σήμα ομιλίας. Προαπαιτούμενο για την σύνθεση φωνής είναι να είναι αυτοματοποιημένη και κατανοητή από τον άνθρωπο. Κάποιες από τις πιο συνήθεις εφαρμογές της σύνθεσης ομιλίας στα σύγχρονα συστήματα επικοινωνίας ανθρώπου – μηχανής είναι η φωνητική εκφώνηση μηνυμάτων στα smartphone, η φωνητική καθοδήγηση των GPS συστημάτων στα αυτοκίνητα, η αποτύπωση λέξεων από ηλεκτρονικούς μεταφραστές ή λεξικά καθώς και οι αυτοματοποιημένοι διάλογοι σε τηλεφωνικές γραμμές τεχνικής υποστήριξης. Όπως και η αναγνώριση ομιλίας, η TTS τεχνολογία βρίσκει εφαρμογή και σε τομείς προσβασιμότητας (π.χ. σε αυτόματα μηχανήματα ATM – για άτομα με περιορισμένη ορατότητα).




**Σχήμα 2: Διαδικασία μετατροπής κειμένου σε ομιλία**

### Σύνθεση με Κανόνες (Formant)

Η διαδικασία της σύνθεσης ομιλίας με χρήση κανόνων γίνεται σε δυο βήματα. Το πρώτο περιλαμβάνει την δημιουργία κάποιων λεκτικών κανόνων για την σύνθεση του ηχητικού σήματος. Οι κανόνες βασίζονται στον μηχανισμό παραγωγής της ανθρώπινης ομιλίας. Στην ουσία αποτελούν ακολουθίες φωνημάτων και παράγονται με την χρήση φίλτρων. Για την δημιουργία τους, ηχογραφείται ένα μεγάλο σύνολο από ακολουθίες φωνηέντων-συμφώνων από τις οποίες εξάγονται οι παράμετροι για τον αλγόριθμο διαμόρφωσης των κανόνων. Μέσω αυτής της ανάλυσης παράγεται μια συμβολοσειρά με τα φωνήματα, η οποία περιλαμβάνει την προσωδία και την τονικότητα του φωνητικού μηνύματος. Στο δεύτερο βήμα της συνολικής διαδικασίας, το παραγόμενο λεκτικό εισάγεται ως παράμετρος σε έναν συνθέτη ο οποίος μετατρέπει το παραμετρικό σήμα σε ψηφιακό σήμα ομιλίας.

### Σύνθεση με Συρραφή (Concatenation)

Στην περίπτωση της σύνθεσης με συρραφή χρησιμοποιείται μια βάση δεδομένων η οποία περιέχει κομμάτια από την κατάτμηση ηχογραφημένης ομιλίας. Τα τμήματα μπορεί να είναι φωνήματα, δίφωνα, συλλαβές ή και λέξεις. Η βάση χρησιμοποιείται σε συνδυασμό με έναν αλγόριθμο επιλογής τμημάτων ο οποίος διαλέγει τα κατάλληλα φωνήματα για την συρραφή του τελικού σήματος. Αυτή η μέθοδος παράγει πιο φυσική ομιλία από την σύνθεση με κανόνες καθώς δεν εφαρμόζει μεγάλο βαθμό επεξεργασίας στα ηχογραφημένα κομμάτια. Η μόνη ουσιαστική επεξεργασία γίνεται στα σημεία ένωσης των φωνημάτων με σκοπό την ομαλή ροή του παραγόμενου μηνύματος.

### 2.3.1 Pyttsx3

Η pyttsx3 της Python είναι μια βιβλιοθήκη που μετατρέπει κείμενο σε ομιλία.

Περιγραφή:

Τα βασικά της χαρακτηριστικά είναι τα εξής. Αρχικά και πιο σημαντικό από όλα είναι ότι δουλεύει χωρίς σύνδεση στο internet, χρησιμοποιώντας τα προεγκατεστημένα API του εκάστοτε λειτουργικού συστήματος. Για παράδειγμα, στα Windows χρησιμοποιεί το SAPI5 (Speech Application Programming Interface). Πέρα από αυτό, διαθέτει διάφορες **γλώσσες** και **φωνές**, παρέχει την δυνατότητα να αλλάξει η **ταχύτητα** και ο **ρυθμός** της ομιλίας, την προσαρμογή της **έντασης** καθώς και την **αποθήκευση** της ομιλίας σε αρχείο.

URL πακέτου: https://pyttsx3.readthedocs.io/en/latest/engine.html

Στο ακόλουθο παράδειγμα μπορούμε να δούμε τις βασικές εντολές παραμετροποίησης ήχου και ρυθμού και τέλος εναλλαγή της φωνής από θηλυκό σε αρσενικό και “εκτύπωση” σε ομιλία “Hello World” με την κάθε φωνή.

```
import pyttsx3
engine = pyttsx3.init()

""" RATE"""
rate = engine.getProperty('rate')   # παίρνουμε την τιμή του ρυθμού.
print (rate)                        # εκτύπωση τιμής ρυθμού
engine.setProperty('rate', 125)     # αλλαγή ρυθμού σε 125
"""VOLUME"""
volume = engine.getProperty('volume')    # παίρνουμε την τιμή της έντασης (min=0, max=1)
print (volume)                          # εκτύπωση τιμής έντασης
engine.setProperty('volume',1.0)    # αλλαγή έντασης σε 1.0
"""VOICE"""
voices = engine.getProperty('voices')       # παίρνουμε την τιμή της φωνής
engine.setProperty('voice', voices[1].id)   # αλλαγή φωνής σε γυναικεία
engine.say("Hello World!")
engine.runAndWait()
engine.setProperty('voice', voices[0].id)  # αλλαγή φωνής σε αντρική
engine.say("Hello World!")
engine.runAndWait()
engine.stop()

"""Αποθήκευση αρχείου"""
engine.save_to_file('Hello World', 'test.mp3')
engine.runAndWait()
```

### 2.3.2 gTTS – Google Text-to-Speech

Περιγραφή:

Το πακέτο gTTS είναι μια σχετικά απλή βιβλιοθήκη. Χρειάζεται σύνδεση στο διαδίκτυο αλλά ταυτόχρονα δεν χρησιμοποιεί κάποιο επί πληρωμή API. Αξιοποιεί το δωρεάν API της google που χρησιμοποιείται για τις μεταφράσεις του google translate.

Μια από τις βασικές δυνατότητες του είναι η χρήση **οποιασδήποτε γλώσσας** με σωστή προφορά και τονισμό καθώς και η αποθήκευση της φωνής σε αρχείο.

URL πακέτου: https://gtts.readthedocs.io/en/latest/index.html

Στο ακόλουθο παράδειγμα αποθηκεύουμε δύο αρχεία, ένα στα Αγγλικά που λέει “hello” και ένα στα Ελληνικά που λέει “γειά”.

```
from gtts import gTTS
tts = gTTS('hello')
tts.save('hello.mp3')

tts = gTTS('Γειά!', lang='el')
tts.save('geia.mp3')
```

## 2.4 Υποστήριξη Φασματικής Ανάλυσης Ομιλίας από την Python

### 2.4.1 pyAudioAnalysis

Περιγραφή:

Το pyAudioAnalysis είναι μια βιβλιοθήκη ανάλυσης και επεξεργασίας ήχου. Παρέχει τις εξής δυνατότητες:

- Εξαγωγή ιδιοτήτων ήχου και οπτικών αναπαραστάσεων (π.χ. mfccs, φασματογράφημα, χρωματογράφημα)
- Ταξινόμηση αγνώστων ήχων
- Εντοπισμός ηχητικών συμβάντων, όπως για παράδειγμα η ύπαρξη περιόδων σιωπής σε ηχητικά κομμάτια
- Κατάτμηση ηχητικών κομματιών και εξαγωγή μικρογραφιών
- Μοντέλα εκπαίδευσης ήχου. Για παράδειγμα ανίχνευση συναισθήματος
- Μετατροπή αρχείων (π.χ. .mp3 σε .wav)

Προαπαιτούμενα:

Για την ορθή λειτουργία όλων των παραπάνω μεθόδων το pyAudioAnalysis περιλαμβάνει ένα σύνολο προαπαιτούμενων βιβλιοθηκών:

Matplotlib, simplejson, scipy, numpy, hmmlearn, eyeD3, pydub, scikit_learn, tqdm, plotly

Ο κώδικας της βιβλιοθήκης pyAudioAnalysis είναι οργανωμένος σε 6 βασικά αρχεία τα οποία εξυπηρετούν τις παραπάνω δυνατότητες.

1) audioAnalysis.py: Περιλαμβάνει τις βασικές λειτουργίες της γραμμής εντολών

2) ShortTertFeatures.py: Εξαγωγή πληροφοριών ήχου

3) MidTermFeatures.py: Εξαγωγή στατιστικών για τις παραπάνω πληροφορίες

4) audioTrainTest.py: Διεργασίες Κατηγοριοποίησης

5) audioBasicIO.py: Συναρτήσεις Εισόδου/Εξόδου αρχείων και μετατροπές

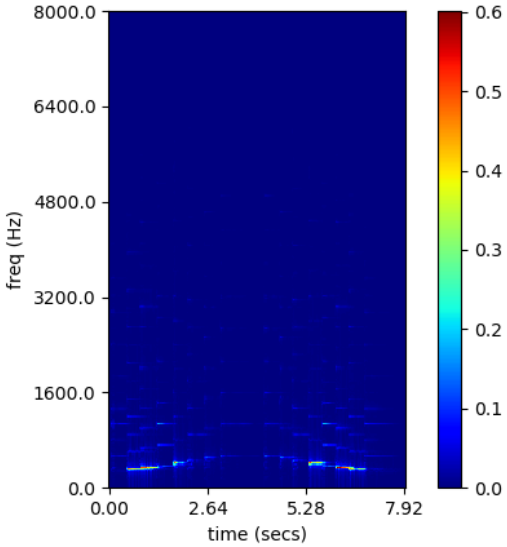
6) audioVisualization.py: Συναρτήσεις απεικόνισης γραφημάτων

URL πακέτου: https://github.com/tyiannak/pyAudioAnalysis

Παραδείγματα χρήσης της βιβλιοθήκης pyAudioAnalysis:

Φασματογράφημα:

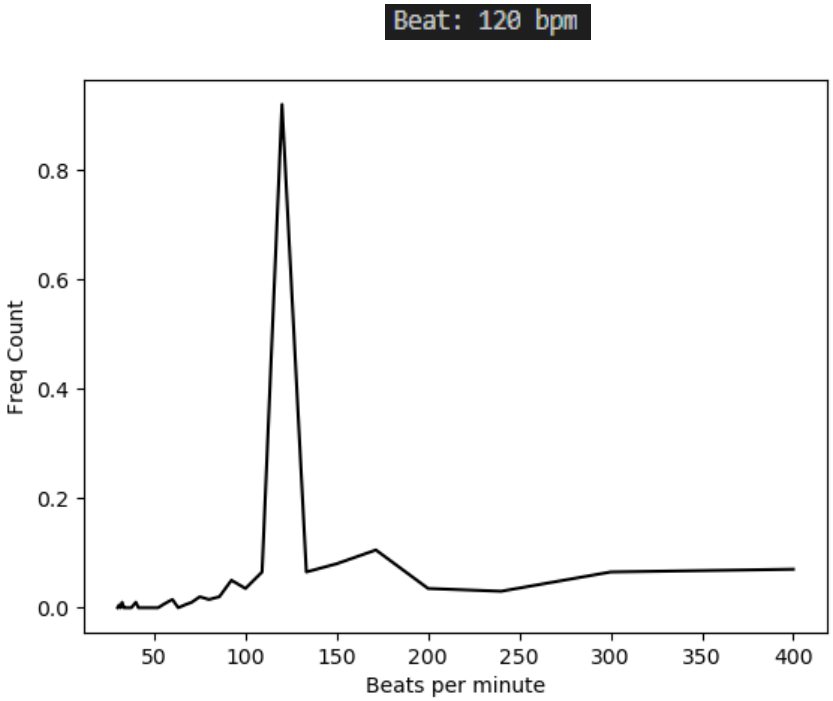
Εντολή: python audioAnalysis.py fileSpectrogram -i data/doremi.wav




**Σχήμα 3: Απεικόνιση φάσματος για το αρχείο doremi.wav**

Εύρεση ρυθμού μουσικού κομματιού:

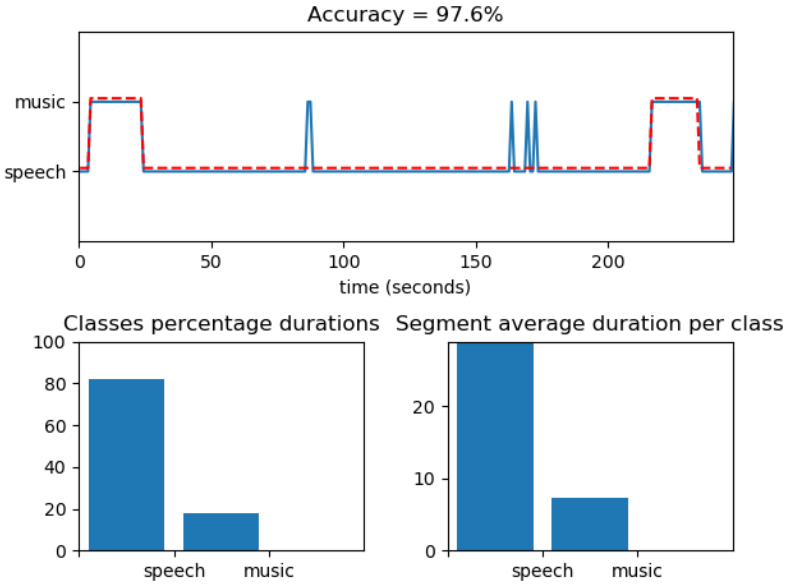
Εντολή: python audioAnalysis.py beatExtraction -i data/beat/small.wav –plot




**Σχήμα 4: Διάγραμμα συχνότητας του κάθε ρυθμού μέσα στο κομμάτι**

Κατηγοριοποίηση – Ταξινόμηση αρχείου ήχου:

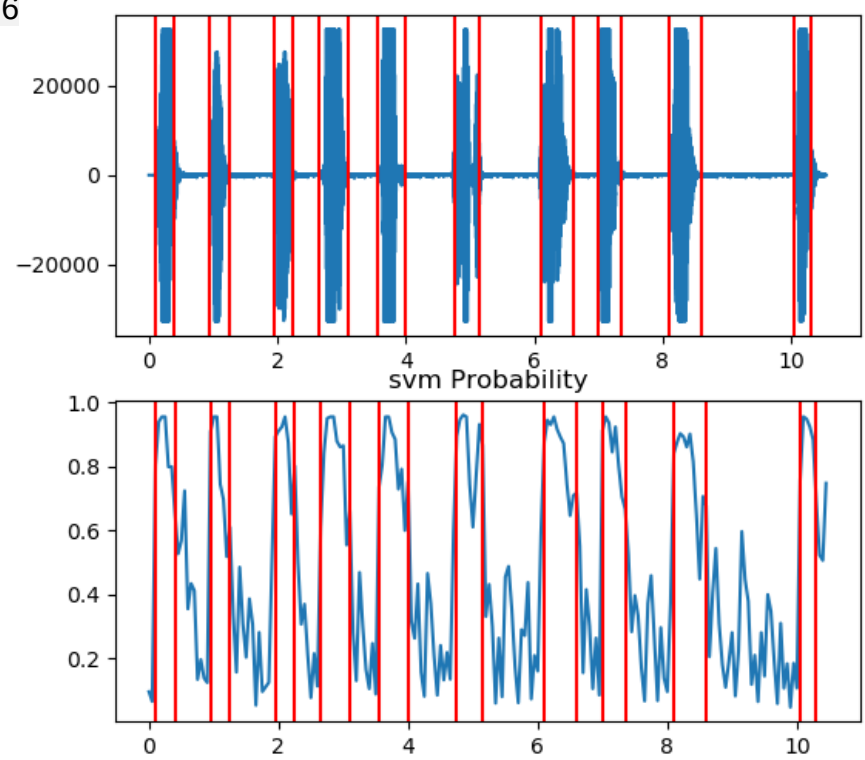
Εντολή: `python audioAnalysis.py segmentClassifyFile -i data/scottish.wav --model svm --modelName data/models/svm_rbf_sm`




**Σχήμα 5: Εύρεση ποσοστού ομιλίας και μουσικής σε αρχείο ήχου**

Εντοπισμός και αφαίρεση σιωπής από ηχητικό κομμάτι:

Εντολή: `python audioAnalysis.py silenceRemoval -i data/count2.wav --smoothing 0.1 --weight 0.6`




**Σχήμα 6: Εντοπισμός περιόδων σιωπής σε ηχητικό αρχείο**

### 2.4.2 Python Speech Features

Περιγραφή:

Το Python Speech Features είναι μια βιβλιοθήκη της Python για την εξαγωγή ιδιοτήτων από σήματα ομιλίας. Μπορεί να χρησιμοποιηθεί σε συνεργασία με άλλες βιβλιοθήκες για την δημιουργία συστημάτων αυτόματης αναγνώρισης ομιλίας (ASR).

Καθώς η ανθρώπινη ομιλία εμπεριέχει στοιχεία θορύβου, για τον προσδιορισμό των φωνημάτων που παράγονται, απαιτείται η εξαγωγή ιδιοτήτων. Οι πλέον καθιερωμένες ιδιότητες για την αναγνώριση ομιλίας ονομάζονται MFCCs (Mel Frequency Cepstral Coefficients) και πρόκειται για σταθερές που αναπαριστούν το φάσμα ισχύος μικρής χρονικής διάρκειας. Πρόκειται για το αποτέλεσμα του μετασχηματισμού Fourier, φιλτραρισμένο με χρονικά παράθυρα που ακολουθούν την κλίμακα Mel. Η κλίμακα Mel είναι μια αντιστοιχία της αντιληπτής συχνότητας/τονικότητας ενός τόνου με την πραγματική μετρούμενη συχνότητα.

Προαπαιτούμενα:

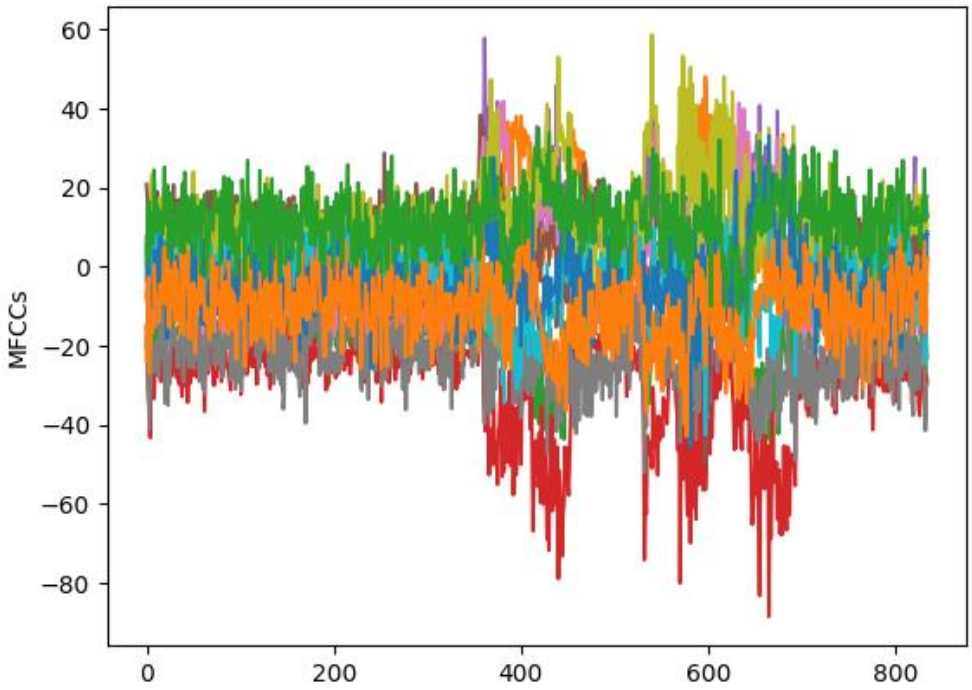
Για την χρήση του Python Speech Features είναι απαραίτητη η εγκατάσταση των βιβλιοθηκών numpy, scipy και matplotlib, για την επεξεργασία και αναπαράσταση των αριθμητικών δεδομένων.

URL πακέτου: https://github.com/jameslyons/python_speech_features

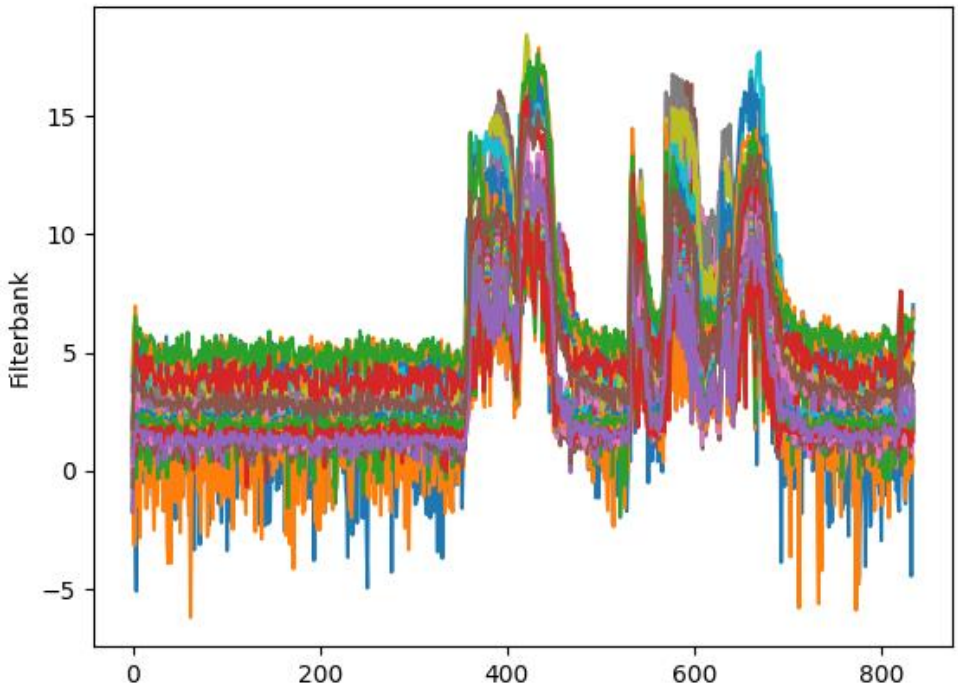
Παράδειγμα εξαγωγής ιδιοτήτων από ηχητικό αρχείο ηχογραφημένης ομιλίας:

```
from python_speech_features import mfcc
from python_speech_features import delta
from python_speech_features import logfbank
import scipy.io.wavfile as wav
import matplotlib.pyplot as plt
(rate,sig) = wav.read("test.wav")
mfcc_feat = mfcc(sig,rate)
d_mfcc_feat = delta(mfcc_feat, 2)
fbank_feat = logfbank(sig,rate)
print(fbank_feat[1:3,:])
plt.plot(fbank_feat)
plt.ylabel('MFCC')
plt.show()
```

Αποτελέσματα:




**Σχήμα 7: Διάγραμμα συντελεστών Mel Frequency Cepstral**




**Σχήμα 8: Διάγραμμα αναπαράστασης Filterbank**

## 2.5 Παρατηρήσεις – Συγκρίσεις

Στον τομέα της αναγνώρισης ομιλίας παρουσιάζονται 3 βιβλιοθήκες. Στον παρακάτω πίνακα παρατηρούνται οι διαφορές τους:

**Πίνακας 1: Σύγκριση βιβλιοθηκών Αναγνώρισης Ομιλίας**

| | SpeechRecognition | Wit | Pocketsphinx |
|---|---|---|---|
| Επιλογή Γλώσσας | Διαθέτει επιλογή γλώσσας χωρίς προαπαιτούμενο. | Διαθέτει μέσω της πλατφόρμας. | Διαθέτει με την λήψη του κατάλληλου αρχείου βάσης. |
| Πηγή εισόδου | Μικρόφωνο (συνεχής) / Από αρχείο ήχου. | Από αρχείο. | Μικρόφωνο (συνεχής) / Από αρχείο ήχου. |
| Λειτουργία online/ offline | Παρέχει λειτουργία offline με την βιβλιοθήκη CMUSphinx αλλά και online APIs. | Δεν διαθέτει. Απαιτεί σύνδεση στο διαδίκτυο. | Παρέχει λειτουργία offline. |
| Ακρίβεια ΑΟ | Είχε μεγάλη ακρίβεια στα Αγγλικά και στα Ελληνικά. | Είχε ακρίβεια στα αποτελέσματα αναγνώρισης. | Δεν είχε ακρίβεια μέσω του μικροφώνου. Αναγνώριζε άλλες λέξεις. |
| Ευκολία Χρήσης / Τεκμηρίωση | Διαθέτει επαρκή τεκμηρίωση, εύκολο στην χρήση. | Η δυσκολία έγκειται στην σύνδεση με την διαδικτυακή πλατφόρμα. Η τεκμηρίωση του εργαλείου είναι επαρκής. | Διαθέτει επαρκή τεκμηρίωση. Δύσκολο στην εγκατάσταση. |

Από τις τρεις βιβλιοθήκες που αναφέρονται η βιβλιοθήκη SpeechRecognition είναι η πιο διαδεδομένη και αποτελεί ένα ολοκληρωμένο εργαλείο για την αναγνώριση ομιλίας μέσω της Python.

Η διεπαφή Wit προσφέρει προχωρημένες δυνατότητες για την ανάπτυξη εφαρμογών τεχνητής νοημοσύνης. Η βιβλιοθήκη Pocketsphinx απαιτεί την συνεργασία με άλλα προγράμματα και λήψη βάσεων δεδομένων για την αποδοτική λειτουργία της.

Μετατροπή κειμένου σε ομιλία:

**Πίνακας 2: Σύγκριση βιβλιοθηκών TTS**

| | Pyttsx3 | gTTS |
|---|---|---|
| Επιλογή Γλώσσας | Διαθέτει επιλογή γλώσσας. | Διαθέτει όλες τις γλώσσες του Google Translate. |
| Φωνές | Διαθέτει διαφορετικές φωνές και αλλαγή ρυθμού / ταχύτητας. | Δεν διαθέτει αλλαγή φωνής. |
| Τύπος εξόδου | Αρχείο ήχου. | Αρχείο ήχου. |
| Λειτουργία Online / Offline | Λειτουργεί offline. | Απαιτεί σύνδεση με το online api της Google. |
| Ευκολία Χρήσης/Τεκμηρίωση | Διαθέτει αναλυτική τεκμηρίωση και είναι απλό στην χρήση. | Είναι εύκολο και διαθέτει την βασική τεκμηρίωση. |

Ανάμεσα στις δύο βιβλιοθήκες το **Pyttsx3** παρέχει περισσότερες δυνατότητες, πληρέστερη τεκμηρίωση και λειτουργεί χωρίς σύνδεση στο διαδίκτυο.

Το **gTTS** είναι μια διεπαφή του Google Translate API για την Python και τουτέστιν ο παραγόμενος λόγος έχει σωστή προφορά και τονικότητα.

Φασματική ανάλυση σήματος ομιλίας:

**Πίνακας 3: Σύγκριση βιβλιοθηκών ανάλυσης ψηφιακών σημάτων ομιλίας**

| | pyAudioAnalysis | Python Speech Features |
|---|---|---|
| Δυνατότητες | Πολλές δυνατότητες ανάλυσης και επεξεργασίας σήματος. Απεικόνιση φάσματος, κατάτμηση, ταξινόμηση και εκπαίδευση μοντέλων ήχου. | Συγκεκριμένες δυνατότητες για την εξαγωγή ιδιοτήτων ομιλίας – Mfccs / Filterbank |
| Ευκολία Χρήσης/Τεκμηρίωση | Παρέχει αναλυτική τεκμηρίωση. Παρέχει χρήσιμα εργαλεία με εύκολη διεπαφή γραμμής εντολών (cli). | Παρέχει επαρκή τεκμηρίωση. Απαιτεί γνώσεις στην επεξεργασία σημάτων ομιλίας. |

# 3. ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΜΟΥΣΙΚΗΣ

## 3.1 Υποστήριξη σύνθεσης μουσικής από την Python

### 3.1.1 Music21

Περιγραφή:

Το music21 είναι μια βιβλιοθήκη ανοιχτού λογισμικού για την μελέτη και σύνθεση μουσικής. Πρόκειται για μια συλλογή εργαλείων μουσικολογίας και επιτυγχάνει την μελέτη μεγάλων dataset μουσικής, την σύνθεση μουσικών κομματιών καθώς και την εκπαίδευση μουσικής θεωρίας. Σκοπός της βιβλιοθήκης είναι η γρήγορη ανάλυση και σύνθεση σε προγραμματιστικό περιβάλλον.

Η βιβλιοθήκη music21 αποτελεί ένα σύνθετο εργαλείο που απαιτεί γνώσεις μουσικής.

Στις βασικές δυνατότητες περιλαμβάνονται:

- Η σύνθεση μουσικών κομματιών
  - Δημιουργία νοτών, συγχορδιών, κλιμάκων
  - Αλλαγή ρυθμού
  - Προσθήκη αρμονίας
  - Παραγωγή παρτιτούρας
- Η ανάγνωση μουσικής παρτιτούρας από αρχεία .xml και μετατροπή σε επεξεργάσιμο αντικείμενο της python
- Η ανάλυση μουσικής και εξαγωγή στατιστικών για κάθε μουσικό κομμάτι
- Δημιουργία MIDI αρχείων
- Μετατροπή μουσικών κομματιών σε πίνακες για αριθμητικές πράξεις

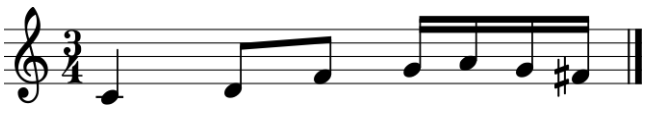
Προαπαιτούμενα:

Η βιβλιοθήκη music21 χρησιμοποιεί τις βιβλιοθήκες matplotlib, numpy, webcolors, chardet, joblib και jsonpickle για την αναπαράσταση στατιστικών γραφημάτων και την προβολή του γραφικού περιβάλλοντος.

URL πακέτου: https://github.com/cuthbertLab/music21

Παράδειγμα δημιουργίας μελωδίας:

```
from music21 import *

littleMelody = converter.parse("tinynotation: 3/4 c4 d8 f g16 a g f#")

littleMelody.show()
```



**Σχήμα 9: Αναπαράσταση μελωδίας από την βιβλιοθήκη music21**

### 3.1.2 Pyo

Περιγραφή:

Το pyo είναι ένα module της Python γραμμένο σε C, που δίνει την δυνατότητα ψηφιακής επεξεργασίας σημάτων. Παρέχει ένα σύνολο από έτοιμες κλάσεις για την επεξεργασία ήχου, παραγωγή αλγοριθμικής μουσικής και δημιουργία ηχητικών λογισμικών.

Βασικό χαρακτηριστικό του pyo και η διαφορά του με άλλες προαναφερθείσες βιβλιοθήκες είναι ότι διαθέτει γραφικό περιβάλλον.

Η βιβλιοθήκη του pyo συμπεριλαμβάνει δυνατότητες όπως:

- Επεξεργασία και ανάλυση ιδιοτήτων σήματος
- Αναπαραγωγή αρχείων ήχου
- Ηχητικά εφέ
- Φίλτρα
- MIDI

Προαπαιτούμενα:

Κατά την εγκατάσταση του pyo με τον package manager της Python δεν απαιτείται κάποιο προαπαιτούμενο. Παρ' όλα αυτά η βιβλιοθήκη εγκαθιστά τις απαραίτητες εξαρτήσεις οι οποίες είναι:

- WxPython – Γραφικό περιβάλλον GUI
- Portaudio – Είσοδος/Έξοδος αρχείων ήχου I/O
- Portmidi – Υποστήριξη αρχείων MIDI
- Libsndfile – .wav αρχεία
- Liblo - .osc αρχεία

URL πακέτου: https://github.com/belangeo/pyo

Στο ακόλουθο παράδειγμα αναπαράγουμε τον ήχο ενός ημιτόνου σε γραφικό περιβάλλον:

```
from pyo import *
s = Server().boot()
s.start()
a = Sine(mul=0.01).out()
s.gui(locals())
```

## 3.2 Υποστήριξη επεξεργασίας μουσικής από την Python

### 3.2.1 pydub

Περιγραφή:

Το pydub είναι μια διεπαφή για την επεξεργασία ήχου και μουσικής στην Python. Δίνει την δυνατότητα μέσω απλών συναρτήσεων για την επεξεργασία μουσικών κομματιών και την προσθήκη ηχητικών εφέ. Οι βασικές ιδιότητες του pydub είναι:

- Αναπαραγωγή ηχητικών αρχείων
- Κατάτμηση αρχείων σε χρονικά μέρη
- Αυξομείωση έντασης σε συγκεκριμένα χρονικά σημεία του κομματιού
- Ανάλυση πληροφοριών κομματιού σχετικά με την ένταση (Amplitude)
- Συρραφή μουσικών αρχείων (concatenation)
- Μεταβολή διάρκειας κομματιού
- Αντιστροφή αρχείου ήχου
- Επανάληψη αρχείου ήχου
- Προσθήκη εφέ όπως fade in, fade out
- Εξαγωγή σε .mp3

Εξαγωγή αρχείου “aieg.mp3” που είναι αντίστροφο του “geia.mp3” από προηγούμενο παράδειγμα.

```
from pydub import AudioSegment
song = AudioSegment.from_file("geia.wav")
backwards = song.reverse()
backwards.export('aieg.mp3',format='mp3')
play(backwards)
```

Προαπαιτούμενα:

Για την εισαγωγή αρχείων ήχου .mp3 απαιτείται μια από τις βιβλιοθήκες ffmpeg / libav. Για την αναπαραγωγή ήχου χρειάζεται το pyAudio ή εναλλακτικά το simpleaudio.

URL πακέτου: https://github.com/jiaaro/pydub

## 3.3 Υποστήριξη ανάλυσης μουσικής από την Python

### 3.3.1 Librosa

Περιγραφή:

Το **Librosa** της Python είναι ένα πακέτο για ανάλυση μουσικής και ήχου. Παρέχει τα βασικά εργαλεία για άντληση πληροφοριών και για ανάλυση.

Βασικές κατηγορίες λειτουργιών βάσει του documentation, είναι η είσοδος/έξοδος ήχου και η ψηφιακή επεξεργασία σήματος, η αναπαράσταση του ήχου/δεδομένων, η εξαγωγή χαρακτηριστικών, αναγνώριση περιόδων, beats per minute και tempo, αποσύνθεση φασματογραφήματος, εφέ, χρονική κατάτμηση και διαδοχική μοντελοποίηση.

Δεν χρειάζεται κάποιο προαπαιτούμενο πακέτο για να δουλέψει, μονάχα σε περιπτώσεις αναπαράστασης κάποιου γραφήματος χρειάζεται το πακέτο Matplotlib.

Στο ακόλουθο παράδειγμα, εφόσον έχουμε εγκαταστήσει το πακέτο Librosa, θα “φορτώσουμε” ένα αρχείο ήχου, θα εντοπίσουμε το τέμπο - beats per minute και θα εμφανίσουμε τα φασματογραφήματα της γραμμικής και εκθετικής συχνότητας.

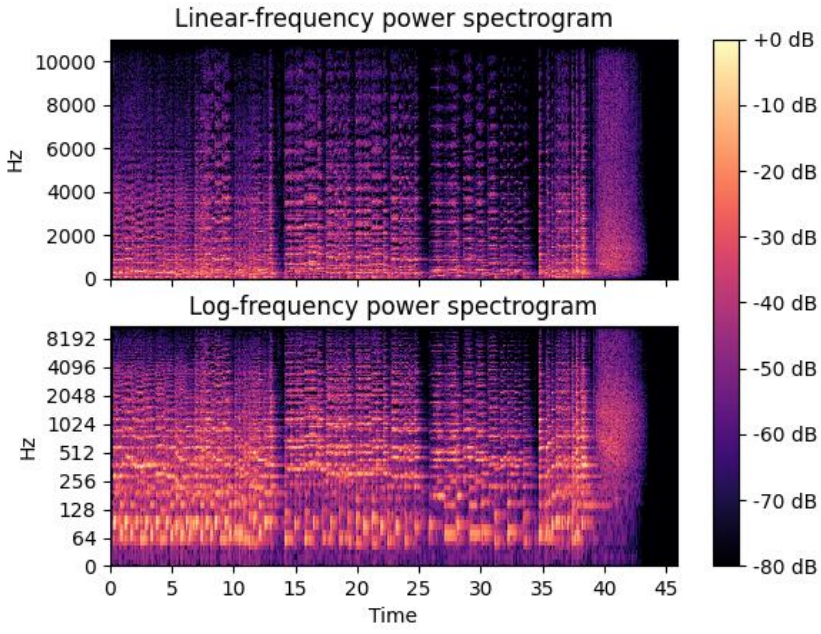
```
import librosa.display
import matplotlib.pyplot as plt
import numpy as np

# 1. Filepath αρχείου μουσικής
filename = 'hung.ogg'
# 2. Φόρτωση ήχου ως κυματομορφή 'y'as a waveform `y`
#    Αποθήκευση του ρυθμού δειγματοληψίας ως `sr`
y, sr = librosa.load(filename)
# 3. Τρέχουμε το beat tracker για το αρχείο μας
tempo, beat_frames = librosa.beat.beat_track(y=y, sr=sr)
# 4. Εκτυπώνουμε το tempo
print('Estimated tempo: {:.2f} beats per minute'.format(tempo))
# 5. Μετατρέπουμε τις εκδηλώσεις beat σε χρονικά διαστήματα
beat_times = librosa.frames_to_time(beat_frames, sr=sr)
# 6. Δημιουργία 2 subplot στο matplotlib
fig, ax = plt.subplots(nrows=2, ncols=1, sharex=True)

# 7. Μετατροπή του φασματογραφήματος πλάτους σε κλιμάκωσης dB
D = librosa.amplitude_to_db(np.abs(librosa.stft(y)), ref=np.max)
# 8. Δημιουργία γραμμικού φασματογραφήματος
img = librosa.display.specshow(D, y_axis='linear', x_axis='time',
```

```
                                sr=sr, ax=ax[0])
ax[0].set(title='Linear-frequency power spectrogram')
ax[0].label_outer()
hop_length = 1024
D = librosa.amplitude_to_db(np.abs(librosa.stft(y, hop_length=hop_length)),
                            ref=np.max)
# 8. Δημιουργία εκθετικού φασματογραφήματος
librosa.display.specshow(D, y_axis='log', sr=sr, hop_length=hop_length,
                         x_axis='time', ax=ax[1])
ax[1].set(title='Log-frequency power spectrogram')
ax[1].label_outer()
fig.colorbar(img, ax=ax, format="%+2.f dB")
plt.savefig('figure.png')
```

URL πακέτου: [https://librosa.org/doc/latest/index.html](https://librosa.org/doc/latest/index.html)

Απαιτείται η αλλαγή - φόρτωση διαφορετικού αρχείου στην γραμμή 7, με το path του.

Αποτελέσματα - έξοδος:




```
Estimated tempo: 152.00 beats per minute
```

**Σχήμα 10: Φάσμα συχνοτήτων και ρυθμός μουσικού κομματιού**

## 3.4 Παρατηρήσεις – Συγκρίσεις

Στους παρακάτω πίνακες συγκρίνονται οι βιβλιοθήκες της Python για την ψηφιακή ανάλυση, επεξεργασία και σύνθεση μουσικής:

**Πίνακας 4: Σύγκριση βιβλιοθηκών για την ανάλυση μουσικής**

| | |
|---|---|
| Librosa | Παρέχει δυνατότητες ανάλυσης. (Φασματογράφημα, ρυθμός κομματιού, εξαγωγή ιδιοτήτων) |
| Pyo | Δεν παρέχει πολλές δυνατότητες για ανάλυση μουσικής. |
| Music21 | Παρέχει δυνατότητες ανάλυσης μουσικής. (Αναγνώριση παρτιτούρας, εξαγωγή στατιστικών) |
| PyDub | Παρέχει κάποιες δυνατότητες ανάλυσης ψηφιακού σήματος ήχου. (Amplitude) |

**Πίνακας 5: Σύγκριση βιβλιοθηκών για την επεξεργασία μουσικής**

| | |
|---|---|
| Librosa | Παρέχει γενικές μεθόδους επεξεργασίας σήματος. Δεν παρέχει εξειδικευμένες δυνατότητες επεξεργασίας μουσικής. |
| Pyo | Παρέχει πολλές δυνατότητες επεξεργασίας ήχου και μουσικής σε επαγγελματικό επίπεδο. |
| Music21 | Δυνατότητες επεξεργασίας σε επίπεδο μουσικής. (Αλλαγή tempo, τονικότητας, κλίμακας) |
| PyDub | Αποτελεί εργαλείο για την επεξεργασία ήχου και μουσικής. Παρέχει πολλές δυνατότητες επεξεργασίας. (Αυξομείωση έντασης, αντιστροφή, εφέ, συρραφή) |

**Πίνακας 6: Σύγκριση βιβλιοθηκών για την σύνθεση μουσικής**

| | |
|---|---|
| Librosa | Δεν παρέχει δυνατότητες σύνθεσης. |
| Pyo | Παρέχει μεθόδους για την σύνθεση αλγοριθμικής μουσικής. |
| Music21 | Παρέχει μεθόδους για την σύνθεση μουσικής και την αναπαράσταση μουσικών κειμένων. |
| PyDub | Δεν αποτελεί εργαλείο σύνθεσης. |

# 4. ΣΥΜΠΕΡΑΣΜΑΤΑ

Μέσω των βιβλιοθηκών που εξετάστηκαν φαίνεται ότι η Python αποτελεί ένα δυνατό και εύχρηστο εργαλείο για την επεξεργασία ομιλίας και μουσικής. Στην γενική περιοχή της επεξεργασίας σήματος συμπεραίνεται ότι με την χρήση των κατάλληλων εργαλείων, μπορεί να ανταγωνιστεί software όπως το MATLAB, χωρίς κόστος.

Η διανομή ανοικτού λογισμικού δίνει την δυνατότητα για ποικιλία προγραμμάτων που αφορούν τον ίδιο τομέα. Παρατηρούμε ότι σε πολλές από τις αναφερθείσες βιβλιοθήκες η τεκμηρίωση έχει διαμορφωθεί από την κοινότητα του ανοικτού λογισμικού και πολλές λύσεις προβλημάτων παρέχονται από αυτόνομους προγραμματιστές.

Επίσης, λόγω της διερμηνευόμενης φύσης της γλώσσας η ανάπτυξη προγραμμάτων και script είναι γρήγορη και αποτελεσματική. Αυτό φάνηκε στην δοκιμή των βιβλιοθηκών, καθώς με την χρήση του package manager της Python (pip), όλα τα πακέτα εγκαταστάθηκαν άμεσα και στην συνέχεια η εκτέλεση τους δεν εμφάνισε προβλήματα. Όταν υπήρχε κάποιο λάθος στον κώδικα του εκτελέσιμου, η επεξήγηση ήταν κατατοπιστική.

Στην επεξεργασία ομιλίας αναφέρονται βιβλιοθήκες που καλύπτουν το ευρύ φάσμα περιοχών μελέτης. Οι βιβλιοθήκες/software που αναφέρονται μπορούν να χρησιμοποιηθούν για έρευνα και για την επαγγελματική ανάπτυξη εφαρμογών.

Στην επεξεργασία μουσικής παρουσιάζονται βιβλιοθήκες που εστιάζουν στην ανάλυση, την επεξεργασία και την σύνθεση μουσικής, ενώ ταυτόχρονα παρέχουν και γενικές μεθόδους επεξεργασίας σήματος. Οι περισσότερες βιβλιοθήκες που εξετάστηκαν παρέχουν γραφικό περιβάλλον, ευκολία στην χρήση και προβάλλουν εκπαιδευτικές δυνατότητες.

Συμπερασματικά, η υποστήριξη ψηφιακής επεξεργασίας ομιλίας και μουσικής είναι εφικτή στην γλώσσα Python μέσω των βιβλιοθηκών ανοικτού λογισμικού που παρέχονται δωρεάν στο διαδίκτυο.

# ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

| **Ξενόγλωσσος όρος** | **Ελληνικός Όρος** |
|---|---|
| Format | Τύπος |
| Digital | Ψηφιακή |
| Library | Βιβλιοθήκη |
| Frequency | Συχνότητα |
| Synthesis | Σύνθεση |
| Analysis | Ανάλυση |
| Software | Λογισμικό |

# ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

| | |
|---|---|
| TTS | Text – to – Speech |
| ΨΕΣ | Ψηφιακή Επεξεργασία Σήματος |
| FIR / IIR | Finite Impulse Response / Infinite Impulse Response |
| URL | Uniform Resource Locator |
| MFCCS | Mel-frequency cepstral coefficients |
| API | Application Programming Interface |
| ASR | Automatic Speech Recognition |
| MFCCs | Mel Frequency Cepstral Coefficients |
| ΑΟ | Αναγνώριση Ομιλίας |

# ΑΝΑΦΟΡΕΣ

[1] Jose Unpingco, *Python for Signal Processing*, San Diego, CA, Springer 2014.
[2] Sean A. Fulop, *Speech Spectrum Analysis*, Springer, 2011.
[3] Eric Matthes, *Python Crash Course*, San Francisco, no starch press, 2016.
[4] Barry. P, *Head First Python*. Beijing: O'Reilly, 2017.
[5] J. Glover, V. Lazzarini, J. Timoney, *Python for audio signal Processing,* National University of Ireland, Ireland, 2014
[6] Manaris, Bill & Brown, Andrew, *Making Music with Computers: Creative Programming in Python*, 2014.
[7] DOWNEY. A. B, *THINK DSP: Digital signal processing in python*. SHROFF & DISTR, 2016.
[8] L. R. Rabiner, R. Schafer, *Introduction to Digital Speech Processing,* CA, USA, 2007
[9] Macquarie University, Spectral Analysis, Προσπελάστηκε: Σεπ. 26, 2020. [Online]. Διαθέσιμο: https://www.mq.edu.au/about/about-the-university/faculties-and-departments/medicine-and-health-sciences/departments-and-centres/department-of-linguistics/our-research/phonetics-and-phonology/speech/acoustics/acoustic-analysis-of-sound/spectral-analysis