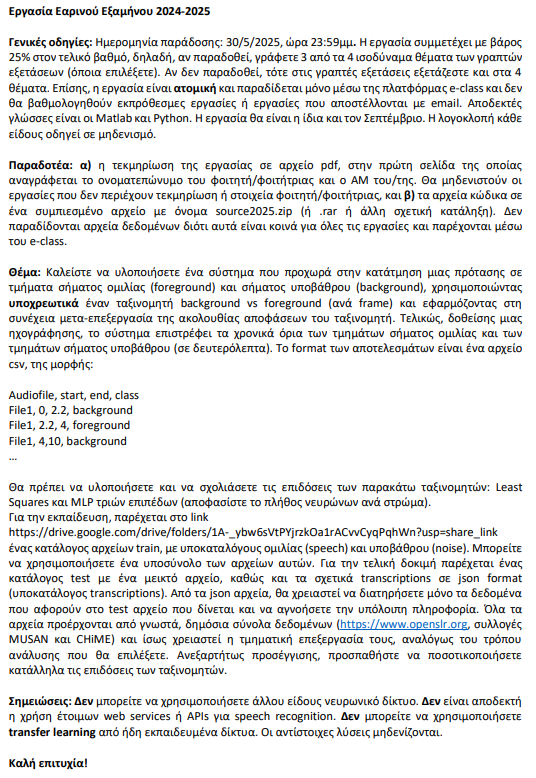
|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Επεξεργασία σημάτων φωνής & ήχου  Εργασία 25% – Εξεταστική Ιουνίου 2025 |
| |  |  |  | | --- | --- | --- | | ΓΕΩΡΓΙΟΣ ΧΡΙΣΤΟΠΟΥΛΟΣ |  | 20206 | |



ΠΕΡΙΕΧΟΜΕΝΑ

[Εισαγωγή3](#eisagogi)

[A. ΠΡΩΤΟ ΠΑΡΑΔΕΙΓΜΑ – ΔΟΚΙΜΗ ΣΕ ΕΝΑ ΑΡΧΕΙΟ4](#A)

[B. ΕΞΑΓΩΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΚΑΙ ΑΡΧΕΙΑ ΗΧΟΥ5](#B)

[Β1. Εξαγωγή MFCC από τα αρχεία ήχου5](#B1)

[Β2. Δημιουργία πίνακα δεδομένων X και ετικετών y5](#B2)

[Β3. Ενδεικτικά αποτελέσματα5](#B3)

[**Γ.ΕΚΠΑΙΔΕΥΣΗ ΜΟΝΤΕΛΩΝ 6**](#C)

[Γ1. Least Squares6](#C1)

[Γ2. Πολυεπίπεδο Νευρωνικό Δίκτυο (MLP)6](#C2)

[Γ3. Αποθήκευση μοντέλων 6](#C3)

[**Δ. ΠΡΟΒΛΕΨΗ ΣΕ test.wav ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΑΠΟΤΕΛΕΣΜΑΤΟΣ7**](#D)

[Δ1. Εφαρμογή του ταξινομητή στο test αρχείο 7](#D1)

[Δ2. Ομαδοποίηση προβλέψεων σε χρονικά διαστήματα 7](#D2)

[Δ3. Αποθήκευση αποτελέσματος σε .csv7](#D3)

**ΕΙΣΑΓΩΓΗ**

**ΣΤΟΧΟΣ:**

Να φτιάξουμε ένα μοντέλο που **μαθαίνει να ξεχωρίζει πότε υπάρχει ομιλία (speech)** και **πότε υπάρχει μόνο θόρυβος (background)** σε ένα ηχητικό αρχείο, έχοντας ως είσοδο ένα ηχητικό αρχείο.

**ΔΕΔΟΜΕΝΑ ΕΚΠΑΙΔΕΥΣΗΣ:**

Από τον φάκελο train/ θα χρειαστούμε:

* 📁 speech/us-gov/ → περιέχει .wav αρχεία ηχητικά αρχεία με καθαρή ομιλία
* 📁 noise/free-sound/ → περιέχει .wav αρχεία με ποικίλους θορύβους, χωρίς παρουσία ομιλίας.

**ΑΠΟ ΗΧΟ ΣΕ ΑΡΙΘΜΟΥΣ:**

Για να **εκπαιδεύσουμε ένα μοντέλο** (π.χ. MLP, Least Squares), πρέπει να μετατρέψουμε τον ήχο σε **αριθμούς,** άρα για κάθε ηχητικό αρχείο θα γίνει:

* ανάγνωση του σήματος με τη **librosa.load()**
* διαχωρισμό σε frames 25ms με βήμα 10ms
* από κάθε frame εξαγωγή **MFCC (Mel-Frequency Cepstral Coefficients),** αριθμητικά χαρακτηριστικά που περιγράφουν το "ηχόχρωμα" κάθε frame ήχου, δηλαδή το φάσμα συχνοτήτων όπως το αντιλαμβάνεται το ανθρώπινο αυτί.
* Οι MFCC ανά frame αποθηκεύονται σε πίνακα X

Κάθε frame χαρακτηρίζεται ως:

* 1 → frame από αρχείο ομιλίας
* 0 → frame από αρχείο θορύβου

Η **τελική έξοδος** του συστήματος δίνεται σε μορφή .csv με χρονικά διαστήματα και κλάση (foreground, background) για κάθε τμήμα.

**A. ΠΡΩΤΟ ΠΑΡΑΔΕΙΓΜΑ – ΔΟΚΙΜΗ ΣΕ ΕΝΑ ΑΡΧΕΙΟ (.wav) [**[**extract\_mfcc.py**](https://unipigr-my.sharepoint.com/personal/p20206_unipi_gr/Documents/Επιφάνεια%20εργασίας/signals2025/source2025/extract_mfcc.py)**]**

Για την αρχική επαλήθευση της μεθόδου εξαγωγής χαρακτηριστικών, εφαρμόζουμε τη διαδικασία σε ένα μόνο αρχείο .wav από τον φάκελο speech/us-gov/. Συγκεκριμένα, χρησιμοποιήθηκε το αρχείο: **speech-us-gov-0005.wav**

Παρακάτω παρατίθεται ο σχετικός κώδικας και η έξοδος:

A screenshot of a computer program

AI-generated content may be incorrect.

**ΠΑΡΑΤΗΡΗΣΕΙΣ:**

* Το αρχείο speech-us-gov-0005.wav περιείχε 2679 frames των 25ms
* Για κάθε frame εξήχθησαν **13 χαρακτηριστικά (MFCCs)**
* Μέχρι στιγμής είναι αρκετό για να ελέγξουμε ότι όλα πήγαν σωστά (π.χ. συμφωνεί το πλήθος των frames).

**ΣΥΜΠΕΡΑΣΜΑ:**

Η αρχική δοκιμή έδειξε ότι λειτουργεί ορθά. Επόμενο βήμα είναι η μαζική εφαρμογή της ίδιας διαδικασίας σε όλα τα .wav αρχεία των φακέλων εκπαίδευσης, ώστε να δημιουργηθεί πλήρες σύνολο εκπαίδευσης για το μοντέλο.

**B. ΕΞΑΓΩΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΚΑΙ ΑΡΧΕΙΑ ΗΧΟΥ (MFCCs) [mfcc\_extraction\_and\_dataset.py]**

**Β1.** **Εξαγωγή MFCC από τα αρχεία ήχου**

Για κάθε αρχείο ήχου:

* Φορτώθηκε με τη συνάρτηση **librosa.load()** με **resampling** στα 16.000 Hz.
* Χωρίστηκε σε **frames** των 25ms με βήμα 10ms.
* Από κάθε frame υπολογίστηκαν **13 MFCCs**, με χρήση της **librosa.feature.mfcc()**.
* Το αποτέλεσμα ήταν πίνακες **[n frames, 13]**, οι οποίοι αποθηκεύτηκαν σε αρχεία .**json** για ευκολία επεξεργασίας.

**Β2.** **Δημιουργία πίνακα δεδομένων X και ετικετών y**

* Τα δεδομένα εισόδου (**X**) είναι όλα τα frames από όλα τα αρχεία, ως διανύσματα 13 χαρακτηριστικών (**MFCCs**).
* Ο πίνακας y περιέχει την αντίστοιχη ετικέτα για κάθε frame:
  + 1 → αν προέρχεται από αρχείο ομιλίας (speech)
  + 0 → αν προέρχεται από αρχείο θορύβου (noise)

Με αυτόν τον τρόπο, δημιουργείται ένα **πλήρες σύνολο εκπαίδευσης**, κατάλληλο για επίβλεψη με χρήση ταξινομητών όπως Least Squares και MLP.

**Β3. Ενδεικτικά αποτελέσματα**

Με χρήση 20 αρχείων ανά κατηγορία (speech και noise), προέκυψαν τα εξής:

* X shape: (931936, 13)
* y shape: (931936,)
* Ετικέτες: frames ομιλίας(1): 907381, frames θορύβου (0): 24555

**ΠΑΡΑΤΗΡΗΣΕΙΣ:**

* Το πρόγραμμα είναι γενικό και μπορεί να εφαρμοστεί σε μεγαλύτερο σύνολο αρχείων χωρίς τροποποιήσεις.
* Οι MFCCs αποθηκεύονται εξωτερικά για επαναχρησιμοποίηση χωρίς επανυπολογισμό.
* Το τελικό X είναι έτοιμο για να χρησιμοποιηθεί ως είσοδος στα μοντέλα ταξινόμησης.

**Γ.ΕΚΠΑΙΔΕΥΣΗ ΜΟΝΤΕΛΩΝ**

Στο στάδιο αυτό, εκπαιδεύουμε δύο μοντέλα για την ταξινόμηση των frames του ήχου:

**Γ1. Least Squares**

* Χρησιμοποιήθηκε η **LinearRegression** από τη βιβλιοθήκη **sklearn**.
* Οι τιμές εξόδου μετατράπηκαν σε 0/1 βάσει κατωφλίου 0.5.
* Η μέθοδος αυτή χρησιμοποιήθηκε ως **baseline**, επειδή είναι απλό γραμμικό μοντέλο χωρίς δυνατότητα αποτύπωσης μη γραμμικών σχέσεων μεταξύ **MFCC** χαρακτηριστικών και ετικετών.

A screen shot of a computer program

AI-generated content may be incorrect.

**Γ2. Πολυεπίπεδο Νευρωνικό Δίκτυο (MLP)**

* Χρησιμοποιήθηκε MLPClassifier με 3 επίπεδα:
* 1ο: 64 νευρώνες
* 2ο: 32 νευρώνες
* 3ο: 16 νευρώνες
* Ορίστηκε max\_iter=100 και random\_state=42 για αναπαραγωγιμότητα.

A screen shot of a computer program

AI-generated content may be incorrect.

**Γ3. Αποθήκευση μοντέλων**

* Και τα δύο μοντέλα αποθηκεύτηκαν με joblib για επαναχρησιμοποίηση στο στάδιο δοκιμής.

**Αποτελέσματα εκπαίδευσης (στο ίδιο σύνολο):**

* Least Squares Accuracy: 0.84
* MLP Accuracy: 0.97

Τα αποτελέσματα δείχνουν ότι το **MLP** υπερέχει στην ταξινόμηση έναντι του **Least Squares**, όπως αναμενόταν λόγω της ικανότητάς του να μοντελοποιεί μη γραμμικές σχέσεις. Ωστόσο, σημειώνεται ότι η αξιολόγηση έγινε στο **ίδιο σύνολο δεδομένων** με αυτό της εκπαίδευσης. Για πιο ρεαλιστική αποτίμηση της απόδοσης των μοντέλων, θα ήταν σκόπιμη η χρήση ξεχωριστού συνόλου επικύρωσης ή η εφαρμογή τεχνικών όπως η διασταυρούμενη επικύρωση (cross-validation), ώστε να περιοριστεί ο κίνδυνος υπερεκπαίδευσης (overfitting).

**Δ. ΠΡΟΒΛΕΨΗ ΣΕ test.wav ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΑΠΟΤΕΛΕΣΜΑΤΟΣ**

**Δ1. Εφαρμογή του ταξινομητή στο test αρχείο**

* Φορτώθηκε το test.wav και μετατράπηκε σε MFCCs με τις ίδιες παραμέτρους (25ms frame, 10ms hop).
* Χρησιμοποιήθηκε το εκπαιδευμένο μοντέλο (MLP) για πρόβλεψη της ετικέτας του κάθε frame.

**Δ2. Ομαδοποίηση προβλέψεων σε χρονικά διαστήματα**

* Συνεχόμενα frames με ίδια πρόβλεψη ομαδοποιήθηκαν σε ενιαία segments.
* Οι χρονικές τιμές υπολογίστηκαν πολλαπλασιάζοντας τον αριθμό frame με το χρονικό βήμα (hop\_duration = 0.01 sec).
* Η κάθε πρόβλεψη μετατράπηκε σε foreground (1) ή background (0), ώστε να συμμορφώνεται με τις απαιτήσεις της εκφώνησης.

**Δ3. Αποθήκευση αποτελέσματος σε .csv**

Το τελικό αρχείο results.csv έχει μορφή:

A screenshot of a computer

AI-generated content may be incorrect.