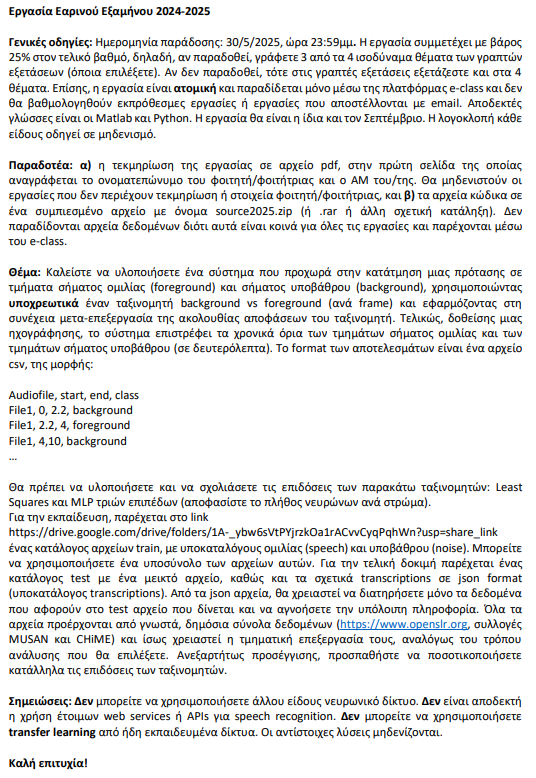
|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Επεξεργασία σημάτων φωνής & ήχου  Εργασία 25% – Εξεταστική Ιουνίου 2025 |
| |  |  |  | | --- | --- | --- | | ΓΕΩΡΓΙΟΣ ΧΡΙΣΤΟΠΟΥΛΟΣ |  | 20206 | |



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

**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)]**

Εφαρμόζουμε την παραπάνω διαδικασία σε ένα μόνο αρχείο από τον φάκελο speech/us-gov/, ώστε να επαληθευτεί η ορθότητα της μεθόδου.

Παρακάτω παρατίθεται ο σχετικός κώδικας και το αποτέλεσμα για το **speech-us-gov-0005.wav**:

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 των χαρακτηριστικών MFCC από τα αρχεία .wav

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 των πινάκων δεδομένων X και y που απαιτούνται για την εκπαίδευση του ταξινομητή.

* 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: Counter({np.int64(1): 907381, np.int64(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**