**REPORT ON AUDIO CLASSIFICATION ON ALZHEIMER’S DISEASE**

**Data :**

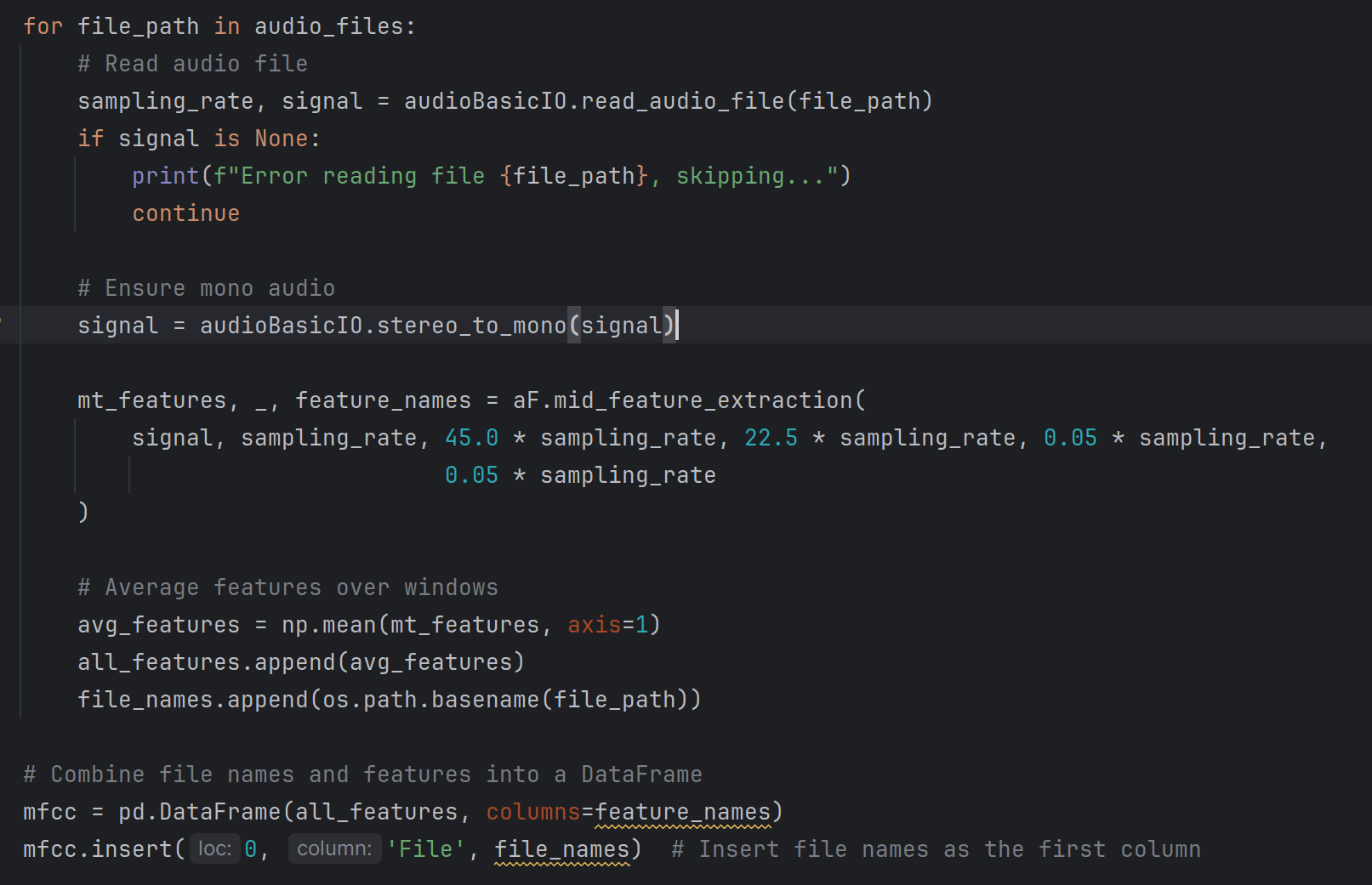
273 audios(mp3 files) από προσωπικές συνεδρίες πιθανών ασθενών με ειδικούς όπου τους παρέχεται μια εικόνα να περιγράψουν + ground truth dataset όπου εκτός των labels (ProbabaleAD, Control), περιέχει και πληροφορία σχετικά με age, gender, education και mmse scores.

**Data exploration και πρώτη αντίληψη:** Τα audio αντιστοιχίζονται στο groundtruth dataset μέσω του filename, και είναι ισομερώς μοιρασμένα ανάμεσα στις 2 κλάσεις (52% Alzheimer, 48% Control).

Εξαγωγή metadata των audio με τη βιβλιοθήκη mutagen ωστε να εξασφαλιστεί η ακεραιότητα των audio και πως δε θα υπάρχει η ίδια συνεδρία 2 φορές. Δεν εξήχθει σημαντική πληροφορία καθώς σε κάποια audio δε βρεθηκαν metadata ενω σε άλλα βρεθηκε πληροφορία σχετική μόνο με το format και το encoding του αρχειου.

Ακούγοντας τα περισσότερα audio, εγινε γρήγορα αντιληπτό πως στα audio που ήταν labeled ως probableAD ο ειδικός παρεμβένει αρκετά και καθοδηγεί. Αν ήταν μεμονομένο, σε λίγα audio θα επιχειρούσα να περικόψω τη φωνή του γιατρού, αλλά ήταν σε πάρα πολλά αν όχι σε όλα οπότε άρχισα να το αντιλαμβάνομαι ως feature.

**Feature extraction :** Εξαγωγή mfcc features χρησιμοποιώντας την pyAudioAnalysis για διαφορετικά mid-term windows ώστε να έχω μία κάποια διαφορετικότητα ανάμεσα στα test μου και να κατανοήσω βαθύτερα το task at hand. Γενικά προτιμήθηκαν mid-terms των 10, 15 ή και 30 δευτερολέπτων ώστε να κάνω capture τη ροή του λόγου καθώς στην περίπτωση του alzheimer παίζει σημαντικό ρόλο. Παράλληλα έκανα calculate το average των mid-term. Ενδεικτικά μία περίπτωση στην οποία εξάγω features με sampling rate 45 δεύτερα:



Κάθε διαφορετική εξαγωγή feature έγινε save σε csv ωστέ να χρησιμοποιηθεί κατά βούληση.

Παράλληλα, θεώρησα πως θα ήταν καλύτερο να αποκτήσω extra δεδομένα, οπότε με την assemblyai εξήγαγα text από τα audio και με την nltk έκανα 1) tokenization του text σε λέξεις, και στη συνέχεια 2) εξήγαγα τα features word variance και hapax legomena.

Η διαδικασία του feature extraction έγινε με τα python files ***extract\_MFCC*** και ***feature\_extraction\_from\_text*** που βρίσκονται στο public repo μου.

**Cleaning και περαιτέρω ανάλυση και intuition πάνω στα data:** To groundtruth dataset περιείχε null values στο education και categorical values (gender). Αυτό που αποφάσισα για να έχω μεγαλύτερη συνοχή και να παρέμβω με το μικρότερο δυνατό impact ήταν τα null values του education να αποκτήσουν τη μέση τιμή που αντιστοιχεί στην κλάση τους. Το gender μετατράπησε σε 0 και 1.

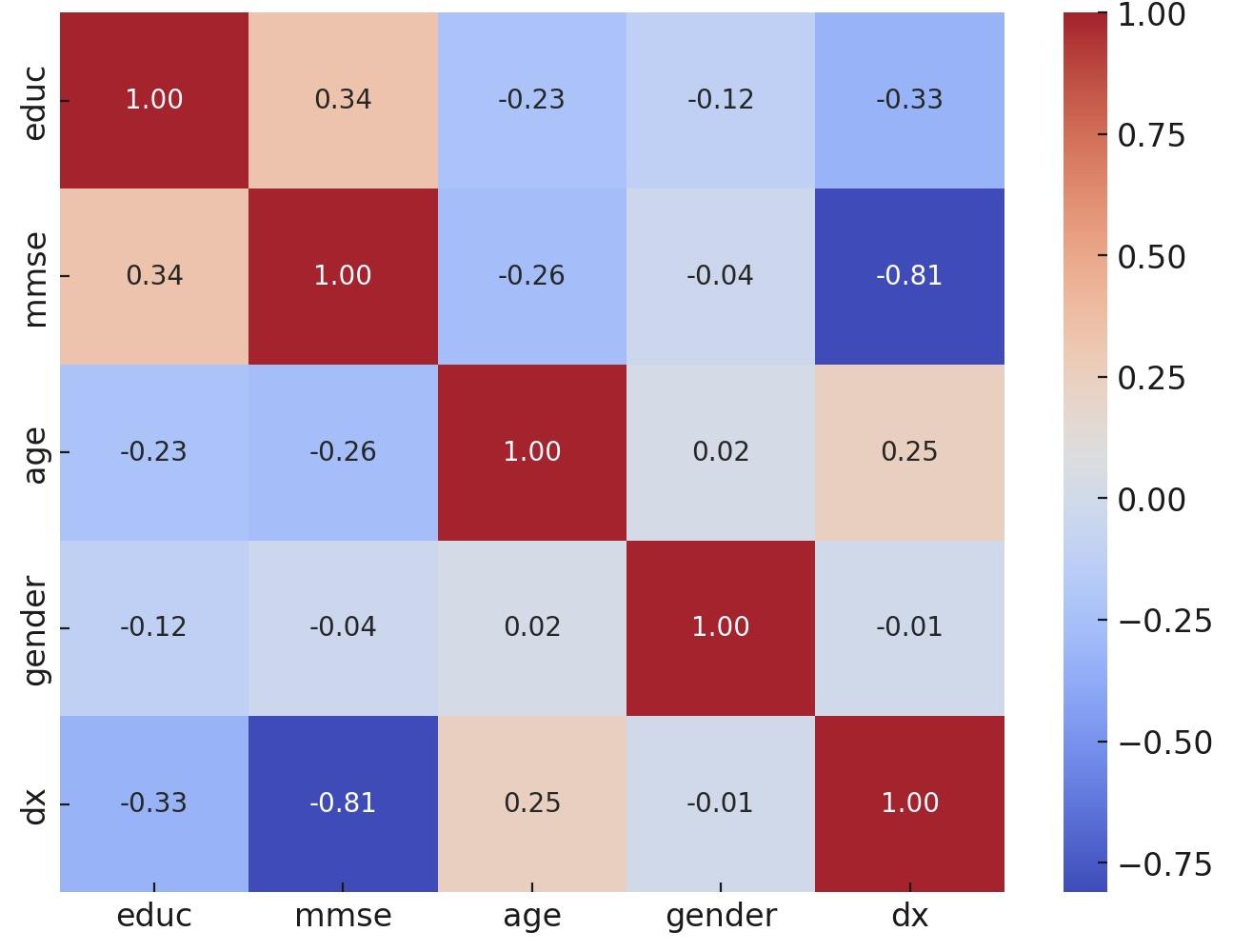
Έχοντας πλέον όλα τα δεδομένα (mfcc features, text features, groundtruth dataset), υπολόγισα correlation για καθένα από αυτά σχετικά με τις διαγνώσεις.

Στα mfcc παρατηρήθηκε μικρό correlation και αυτό μόνο γιά 5-10 εκ των 137 συνολικά. Μεγαλύτερο correlation εντοπίστηκε στα mffc\_5, delta\_mfcc\_5\_std, mfcc\_5\_mean, chroma\_2\_mean.

Αντίθετα, το groundtruth dataset παρουσίασε μεγαλύτερο correlation και συγκεκριμένα το mmse score το οποίο δεν είναι τυχαίο ότι αποτέλει και ειδική εξέταση πάνω στο alzheimer φάνηκε να είναι highly correlated με τη διάγνωση.

Τελικώς τα 2 text features έδειξαν και αυτά correlation και ειδικότερα το feature hapax legomena.

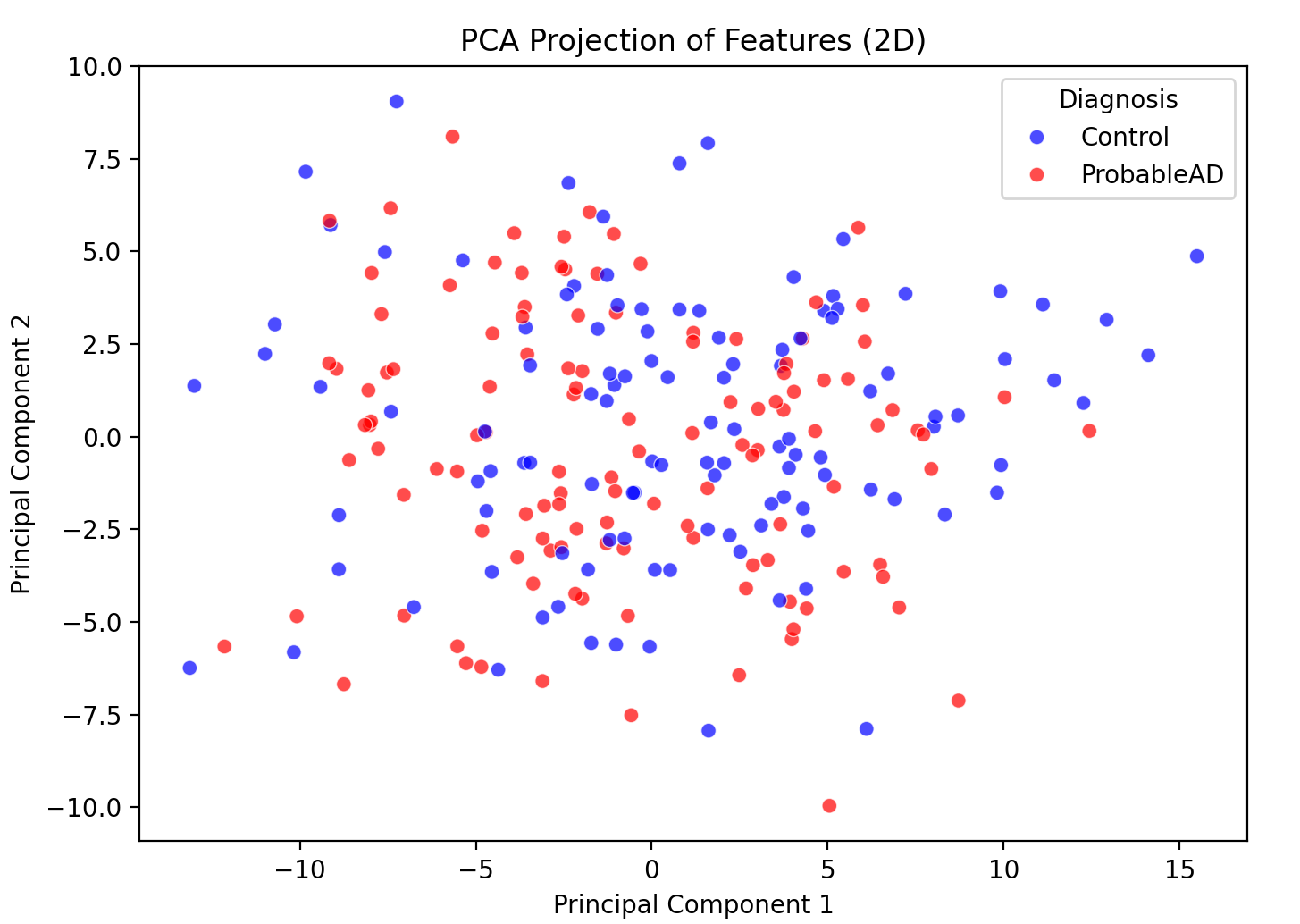
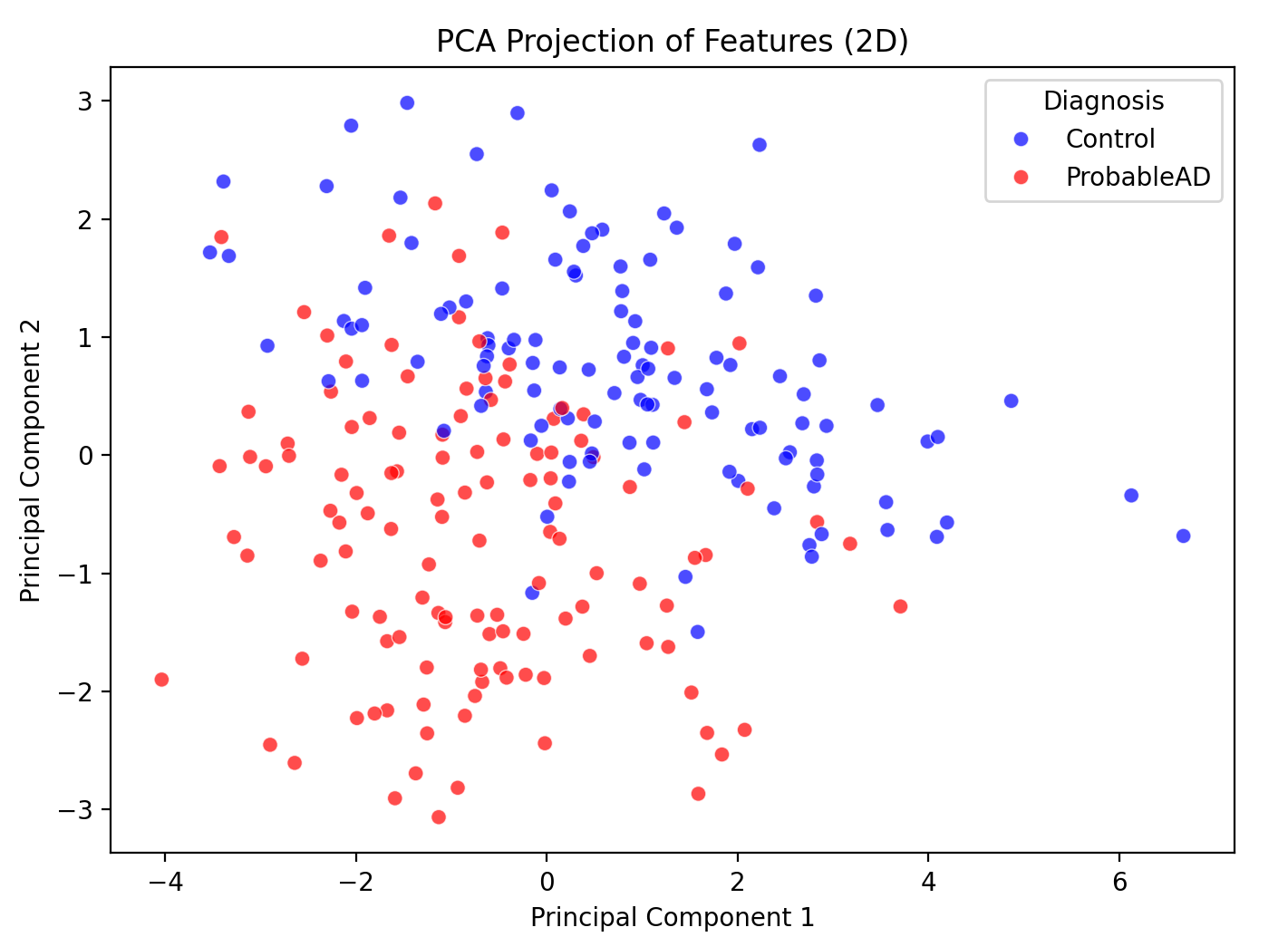
Ενδεικτικά το correlation του groundtruth dataset:



**Finalised datasets και fusions:** Με την πληροφορία που απέκτησα από την προηγούμενη ανάλυση κατέληξα να δημιουργήσω διαφορετικές εκδοχές των data μου. Κάθε διαφορετικό είδος feature παραμένει ως μοναδικό dataset για μεγαλύτερη ανάλυση και χρήση σε majority vote αργότερα. Παράλληλα, παρήγαγα 2 νέα dataset από τα mfcc features. Ένα mffc\_top\_5 και ένα mfcc\_top\_10 που αφορούν τα 5 και 10 πιό correlated features αντιστοιχα, για να μειώσω το dimensionality και τον πιθανώς αχρείαστο θόρυβο.

Τέλος στο κομμάτι του early fusion ένωσα όλα τα datasets μαζί τους αλλά με τις διαφορετικές εκδοχές των mfcc. Δηλαδή ένα combined dataset με τα top 5 (5 mfcc + 4 groundtruth + 2 lexical richness), ένα με τα top 10 mfcc (10 mfcc + 4 groundtruth + 2 lexical richness) και ένα με όλα τα features (137 mfcc + 4 groundtruth + 2 lexical richness). Python file - ***fuse\_data***

**Further analysis :** Για να επιβεβαιώσω πως έπρεπε τα dimensions των mfcc να μειωθούν έκανα calculate principal components και scatter plot πάνω στη διάγνωση. Όντως ο θόρυβος ήταν πάρα πολύς στο μεγάλο dataset που περιέχει όλα τα features ένω αντίθετα στο combined dataset με τα top 5 άρχισε να υπάρχει ένα ολοφάνερο clustering με ένα κάποιο overlaping ομως. (python file - ***PCA\_representation***)



Αυτό με έκανε να συγκεντρωθώ περισσότερο στα datasets με το λιγότερο dimensionality καθώς έδειχνα πως θα συμπεριφέρονταν καλύτερα στους classifiers και θα ήταν και κάτι computationally πιο efficient σε περίπτωση γενίκευσης σε μεγαλύτερα dataset μελλοντικά. ***Κάθε dataset έγινε standardized πριν χρησιμοποιηθεί στους classifiers.***

**Επιλογή classifiers:**

1. **SVM** γιατί με τους kernel μπορεί να κάνει handle καλύτερα το overlapping που παρουσιάζεται, είναι φύσει binary classifier, και μπορεί να δουλέψει αποδοτικά σε μικρά dataset αλλά και σε μεγάλο dimensionality.
2. **LDA** και logistic regression κυρίως λόγω της γραμμικότητας που παρουσιάζεται σε κάποια data αλλά και ως baseline στα αρχικά τεστ μου.
3. **Majority voting** με combination SVM και LDA.

**Αποτελέσματα για combined dataset (mfcc\_top\_5 + text\_features + groundtruth) :**

| **svm.SVC(kernel='rbf', gamma=0.009, C=1.2, class\_weight={'Control': 1, 'ProbableAD': 1.5})** | **TRAIN-TEST SPLIT test\_size=0.3, random\_state=16** | **Accuracy: 0.8**  **F1 Score Control: 0.758**  **F1 Score ProbableAD: 0.837** |
| --- | --- | --- |
| **svm.SVC(kernel='rbf', gamma=0.009, C=1.2, class\_weight={'Control': 1, 'ProbableAD': 1.5})** | **KFold(n\_splits=10, shuffle=True, random\_state=42)** | **Mean Accuracy: 0.801**    **F1 Score Control: 0.79**  **F1 Score ProbableAD: 0.809** |
| **LinearDiscriminantAnalysis(solver='eigen', shrinkage=0.8)** | **TRAIN-TEST SPLIT test\_size=0.3, random\_state=16** | **Accuracy: 0.79**  **F1 Score Control: 0.776**  **F1 Score ProbableAD: 0.805** |
| **LinearDiscriminantAnalysis(solver='eigen', shrinkage=0.8)** | **KFold(n\_splits=10, shuffle=True, random\_state=42)** | **Mean Accuracy: 0.805**  **F1 Score Control: 0.79**  **F1 Score ProbableAD: 0.816** |
| **Majority Vote of:**  **MFCC top 5 classifier: svm.SVC(kernel='rbf', C=0.8)**  **Training Groundtruth classifier: svm.SVC(kernel='rbf', C=0.05)**  **Text Features classifier: LinearDiscriminantAnalysis(solver='eigen', shrinkage=1)** | **TRAIN-TEST SPLIT test\_size=0.3, random\_state=16** | **Accuracy: 0.791**  **F1 Score Control: 0.754**  **F1 Score ProbableAD: 0.819** |
| **Majority Vote of:**  **MFCC top 5 classifier: svm.SVC(kernel='rbf', C=0.8)**  **Training Groundtruth classifier: svm.SVC(kernel='rbf', C=0.05)**  **Text Features classifier: LinearDiscriminantAnalysis(solver='eigen', shrinkage=1)** | **KFold(n\_splits=10, shuffle=True, random\_state=42)** | **Accuracy: 0.793**  **F1 Score Control: 0.77**  **F1 Score ProbableAD: 0.8** |

Θεωρώ ότι το SVM συμπεριφέρθηκε πολύ καλά στα data που επέλεξα (το fused dataset με τα important features - μικρότερο dimensionality ), ενώ παράλληλα μου επέτρεπε να κάνω tweaks στις παραμέτρους του ώστε να φτάσω στο στόχο μου. Ποιός ήταν αυτός; Accuracy ανάμεσα στο 75-80% με κύριο στόχο να μην γίνονται misclassify τα probableAD χωρίς όμως παράλληλα να υπάρξει overfit προς μία κλάση. Όποτε έπρεπε να έχω αρκετά ισορροπημένα αποτελέσματα. Το LDA παρόλο που και αυτό έφερε το επιθυμιτό αποτέλεσμα, μου φάνηκε κάπως πιο δυσπρόσιτο, καθώς δυσκολεύτηκα να το κάνω tune προς την κατεύθυνση που ήθελα. Τελικώς το ensemble επίσης απέδωσε όπως περίμενα. Αν και κάπως πιο κοστοβορο υπολογιστικά στο implementation μου το θεωρώ πολύτιμο εργαλείο για 2 λόγους :   
 1) Τα mffc πάσχιζαν να πετύχουν accuracy >60%, ενώ τα text features δε ξεπερνούσαν το 70%. Αυτό με έκανε να υιοθετήσω ένα weighting approach και να επιτρέψω ένα μικρό overfit στο groundtruth dataset ώστε να ακυρώνει το impact που είχαν τα άλλα 2 datasets, φτανοντας έτσι συνολικά στο 79-80%.

2) Αυτή η ελαστικότητα που μου προσέφερε το ensemble ανάμεσα σε διαφορετικά data με διαφορετική συμπεριφορά με έκανε να σκεφτώ πως θα ήταν πολύτιμο εργαλείο σε πιθανά improvements του implementation μου. πχ εξτρά data διαφορετικού τύπου. Οπως εξαγωγή spectograms από τα audio, και επεξεργασία εικόνας( των spectogram ) για εξαγωγή extra features. Νόμίζω πως εκεί το ensemble θα επιβεβαίωνε αυτο που λέω.

Ήταν συνολικά μία πολύ ενδιαφέρουσα διαδικασία όλο το pipeline και δεν αποκλείεται να προχωρήσω στο improvement με τα spectograms.

Ευχαριστώ