**Rapport projet PAF**

**PROSODIE**

**Choix des features ( Donner la documentation )**

Je travaille ( seule ) sur la prosodie, c’est-à-dire :

La voix est très importante quand à la détermination de la confiance.

Comme j’avais un vecteur trop grand, je l’ai réduit en enlevant les min, et les max.

Voici les features que j’ai retenues et leurs utilité :

-"F0semitoneFrom27.5Hz\_sma3nz\_mean": Il s'agit de la moyenne des valeurs de la fréquence fondamentale (F0) des segments de l'audio, convertie en demi-tons par rapport à une fréquence de référence de 27,5 Hz.

-"F0semitoneFrom27.5Hz\_sma3nz\_stddev": Il s'agit de l'écart-type des valeurs de la fréquence fondamentale (F0) des segments de l'audio, convertie en demi-tons par rapport à une fréquence de référence de 27,5 Hz.

-"F0semitoneFrom27.5Hz\_sma3nz\_range": Il s'agit de la plage (différence) entre la valeur maximale et minimale de la fréquence fondamentale (F0) des segments de l'audio, convertie en demi-tons par rapport à une fréquence de référence de 27,5 Hz.

-"voiceProb\_sma3nz\_mean": Il s'agit de la moyenne des valeurs de probabilité de voix (voice probability) des segments de l'audio. Cette fonctionnalité indique la probabilité qu'un segment donné soit une voix.

-"pcm\_LOGenergy\_sma3nz\_mean": Il s'agit de la moyenne de l'énergie de la piste d'origine du signal audio.

-"pcm\_fftMag\_fband250-650\_sma3\_stddev": Il s'agit de l'écart-type des magnitudes (spectre en fréquence) des coefficients de transformation de Fourier à court terme (FFT) dans la bande de fréquences entre 250 Hz et 650 Hz du signal audio.

Toutes ces features sont intéressantes pour l’analyse de la confiance, et assez diversifiées. Pour les choisir j’ai utilisé la documentation données dans les documents paf, et aussi des ressources sur openSmile trouvées sur internet.

**Méthodologie extraction de features**

Pour l’extraction des features, j’ai travaillé avec OpenSmile.

OpenSmile est assez pratique, car il permet de travailler facilement avec des données continues.

Pour ce qui est de la segmentation j’ai travaillé avec AudioSegment ( pydub ).

Au début, j’ai eu des problèmes pour utiliser OpenSmile car je devais travailler avec une version de python que je n’avais pas sur mon ordinateur ( je travaille originellement avec Spyder et j’ai donc du travailler avec Visual Studio Code )

Au départ, je crée des fichiers cvs, mais mes camarades ont voulu uniformiser nos jeux de données, et donc j’ai du créer des fichiers npy à la place.

J’ai rencontré d’autres problèmes, par rapport à la création de fichier npy avec des chemins d’accès variables. J’ignorai que ça ne crée que des fichiers et pas des dossiers.

J’ai donc créer ces dossiers et sous dossiers à la main.

Mon extraction se fait en plusieurs étapes :

1-Decoupe des audios en segments et enregistrements de ces sous audios

2-On récupère les features pour chacun des sous audio

3-On les stocke dans un fichier dédier et des sous fichiers

**Présentation des scores du modèle de Machine Learning ( Analyse des résultats )**

Pour le machine learning je suis partie sur une regression logistique.

J’ai commencé par concaténer toutes les features des segments pour chaque intéraction, et j’ai ensuite concaténer les segments associés à chaque intéraction d’apprentissage.

J’ai mis ces vecteurs en X.

Et le résultat ( c’est-à-dire le niveau de confiance ) en Y.

J’ai changé l’intéraction test, pour que chaque interaction parmi celles qu’on avait puisse être une interaction test.

Voici mes résultats :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Analyse du modèle / Intéraction Test | **9** | 10 | 12 | 15 | 18 | 19 | 24 | 26 | 27 | 30 |
| Accurancy | 0.5904862085086489 | 0.6094736842105263 | 0.5458333333333333 | 0.5760361876709447 | 0.5650536503261099 | 0.558447296444351 | 0.5659574468085106 | 0.6042105263157895 | 0.5483146067415732 | 0.565614261168385 |
| Recall | 0.38585607940446653 | 0.3873626373626374 | 0.4775707384403036 | 0.4666666666666666 | 0.39999999999999997 | 0.4595238095238095 | 0.5013262599469496 | 0.3967325721711686 | 0.5664603595638078 | 0.4327122153209109 |
| F1 | 0.3763880771478667 | 0.3585585585585585 | 0.44708994708994704 | 0.3945141065830722 | 0.3922120572366878 | 0.43018419489007725 | 0.4927536231884058 | 0.34549656388736855 | 0.5122655122655123 | 0.39627039627039623 |
| Cross Validation Score | 0.5904862085086489 | 0.6094736842105263 | 0.5458333333333333 | 0.5760361876709447 | 0.5650536503261099 | 0.558447296444351 | 0.5659574468085106 | 0.6042105263157895 | 0.5483146067415732 | 0.565614261168385 |

On voit qu’on obtient les meilleurs résultats pour l’Accurancy en utilisant l’interaction 10 en interaction test.

Cela signifie que en moyenne, l’interaction 10 est celle qui ressemble le plus aux autres interactions.