UNIVERSITE DE LAUSANNE
FACULTE DES SCIENCES SOCIALES
ET POLITIQUES
INSTITUT DE PSYCHOLOGIE

How Speech Signals Affect Own and Partner's Emotions: A Machine Learning Approach

Mémoire de Maîtrise universitaire ès Sciences en Psychologie

Présenté par Julien Paillard
Directeur : Peter Hilpert
Expert : Matthew Vowels

Session d'été 2022

An die Musik

Musik: Atem der Statuen. Vielleicht:
Stille der Bilder. Du Sprache wo Sprachen
enden. Du Zeit
die senkrecht steht auf der Richtung
vergehender Herzen.

Gefühle zu wem? O du der Gefühle
Wandlung in was? —: in hörbare Landschaft.
Du Fremde: Musik. Du uns entwachsener
Herzraum. Innigstes unser,
das, uns übersteigend, hinausdrängt, —
heiliger Abschied:
da uns das Innre umsteht
als geübteste Ferne, als andre
Seite der Luft:
rein,
riesig
nicht mehr bewohnbar.

Rainer Maria Rilke

Abstract

Couple interactions have a long history of research in psychology and strong evidence highlighted correlations with many different bio-psycho-social outcomes, such as well-being and health. The current thesis proposes to investigate emotion and speech as two central aspects in couple interactions. Based on advanced technologies allowing acoustic features to be automatically extracted, relationships between acoustic characteristics and emotion have been thoroughly described. Vocal emotion recognition, a new area of research dedicated to the prediction of emotion based on speech signals, emerged a few decades ago and shows great promise. Emotion is a broadly used concept in psychology, but no common definition has yet been agreed on. In regard with a dimensional approach of emotion, arousal and valence are the most studied parameters. If speech signals have been reported to predict arousal with high accuracy, valence is known to show lower accuracy and incoherent results. A better understanding of the link between emotions and speech on a talk-turn level would allow current theories to be developed further and help improve actual and future interventions. Using a machine learning approach, the goal of the present study is to investigate that link with the emotional valence in focus. The results show low performances with the best accuracy rate for women self-reported emotional valence of 59%, but significantly different from chance. Among the four main different categories of acoustic features, voice quality has been assumed in the literature to be a better predictor for valence than pitch, but no significant differences have been found between acoustic features in this study. Propositions for future vocal emotion recognition research based on valence are introduced in the discussion.

Résumé

Une longue histoire de recherche en psychologie sur les interactions de couple a mis en évidence de fortes corrélations en lien avec de nombreux aspects bio-psycho-sociaux, tels que le bien-être et la santé. La présente thèse propose de considérer les émotions et la dimension acoustique du discours comme deux aspects centraux des interactions de couple. En se basant sur des technologies récentes permettant d'extraire automatiquement des caractéristiques acoustiques, les relations entre ces les propriétés acoustiques du discours et les émotions ont déjà pu être décrites en détail dans la littérature. La reconnaissance vocale des émotions, un nouveau domaine de recherche dédié à la prédiction des émotions à partir de signaux vocaux, a émergé il y a quelques décennies et s'avère très prometteuse. Le concept d'émotion est largement utilisé en psychologie, mais aucune définition commune n'a encore été adoptée dans le monde scientifique. Dans le cadre d'une approche dimensionnelle des émotions, l'excitation (*arousal* en anglais) et la valence sont les paramètres les plus étudiés. Si les signaux vocaux permettent de prédire l'excitation avec une grande précision, la valence est connue pour présenter une précision moindre et des résultats incohérents. Une meilleure compréhension du lien entre les émotions et la parole permettrait de développer les théories en cours et d'améliorer les interventions actuelles et futures. En utilisant une approche basée sur l'apprentissage automatique (*machine learning* en anglais) l'objectif de la présente étude est d'examiner ce lien en se focalisant sur la valence émotionnelle. Les résultats montrent de faibles performances, mais néanmoins significativement différentes du hasard. Le meilleur taux de prédiction pour la valence émotionnelle auto-reportée est de 59 %. Parmi les quatre principales catégories des caractéristiques acoustiques, la qualité de la voix a été considérée dans la littérature comme un meilleur prédicteur de la valence que la hauteur de la voix, mais aucune différence significative n'a été trouvée dans cette étude. Des propositions concernant de futures recherches sur la reconnaissance des émotions vocales basées sur la valence sont abordées dans la discussion.

Table of Contents

1. Intro

The relationship between speech and emotion has a long history of research dating back from manuals about rhetoric in antic Greece and, after the contribution of Charles Darwin and other evolutionary theorists, got an increasing attention among scientists from the 19th century (Scherer, 2003). More recently, speech as an expression of emotion generated a considerable number of studies (Keltner et al., 2019; Scherer & Bänzinger, 2004; Scherer, 2003; Juslin & Laukka, 2003; Mauss & Robinson, 2009) and, as anticipated by Scherer (1986), the emergence of an acoustic code for basic emotions is building up. On a dimensional approach of emotion, if the important role of arousal in vocal expression is well documented (Banse & Scherer, 1996), the role played by valence is still in debate and is collecting less-conclusive support (Goudbeek & Scherer, 2010; Belyk & Brown, 2014; Liscombe, 2007). A better understanding of the link between emotions and speech would allow current theories to be developed further and help improve actual and future interventions. To better predict emotion based on speech could be of considerable value regarding the manifold applications of vocal emotion recognition in mental health. Using a predictive approach inspired by the new research area of vocal emotion recognition (for a review see Swain et al., 2018), the current paper, proposes to investigate this issue further. Therefore, this dissertation focuses on how accurate speech acoustic features can predict emotional valence during couple interactions using machine learning.

1.1 Couple Interactions

Couple interactions are a key concept in couple researches and are proven to have a strong impact on many bio-psycho-social outcomes for partners, as well-being and health (Friedlander et al., 2019; Määttä & Uusiautti, 2013; Gottman & Notarius, 2000). A dyadic relationship is built on repetitive and cumulative interactions and creates a unique culture shared by partners (Friedlander et al., 2019). That's why communication within a dyadic interaction provides a pathway to investigate relationship dysfunctions between partners or between therapists and clients (Heyman, 2001). Social expressions, as speech production, and social experiences, as emotional states, emerging during a conversation are rooted in

psycho-biological processes, which are, as proposed by Porges (2001), for a part, inherited from evolution with the function to initiate and support social interactions.

Two levels of analysis exist to observe couple interaction. The first, the macro level, is situated on the overall interaction and can provide general information about different dimensions, as for example the average level of positive affect or the average quality of communication (Friedlander et al., 2019). The second, the micro level, considers the interaction as subdividable into smaller sequences of interaction either based on the talk turn level or on a continuous temporality. Researches have shown that diverse relational outcomes can be predicted based on a small part of interaction and that talk turn level has become a golden standard for many coding systems (Friedlander et al., 2019).

Changes and processes occurring continuously during an interaction are part of a dynamic and self-organized ensemble that constitutes a relationship (Butler, 2011). In that sense, while speaking, partners are simultaneously and mutually influencing each other in a sequence of distinct behaviors. This point is leading to the conceptualization that a given behavior in an interaction limits the response options and, therefore, this response can be predicted (Friedlander et al., 2019).

Partner's behavior is a crucial component of intimate relationships and can be systematically observed as an interpersonal or intrapersonal aspect of interaction. In the current study the main question is how accurate objective interpersonal aspects of interaction (i.e., speech) can predict an intra-personal subjective aspect (i.e., emotional valence) on a talk turn level.

1.2 Interpersonal Aspects: Speech

Behavior, as speech, can be considered as an objective measure widely used to study interactions (Black et al., 2013). The complexity of dyadic interaction can be partially explained by its multidimensional nature, which can be summarize for interpersonal aspects into three dimensions of behavior: verbal (e.g., words, meaning), paraverbal (e.g., pitch, loudness) and nonverbal behaviors (e.g., facial expressions, posture) (Friedlander et al., 2019). Speech is included on many different interaction coding systems and is often associated with nonverbal behaviors as gestures or facial expressions. Facial expressions, more specifically muscle movements, are far the most studied dimension (Keltner et al., 2019),

contrasting with acoustic features who did not elicited much attention till the end of the 20th century. The development gap of technologies needed to analyze acoustic features could partially explain a part of that lag (Scherer, 1986). Another reason concerns the limitation of the human auditory perception system in discerning acoustic characteristics in a reliable, objective and detailed way (Black et al., 2010). Computers are better suited for this task allowing them to extract systematically and automatically a high number of acoustic variables.

Most common features extracted from speech can be classified into four categories: pitch (e.g., level, range and contour of the fundamental frequency), loudness (e.g., energy and amplitude perceived as intensity of the voice), voice quality (e.g., formants and spectral features) and durational measures (e.g., speech rate). Two categories can be linked directly to a specific step of sound production occurring in the body. Initially, air exhaled from the lungs provokes oscillations of the vocal cords located in the larynx (Fitch, 2000). Oscillation's frequency of vocal cords is the physical process that determines the pitch. The acoustic energy generated passes then through the vocal tract (i.e., the pharyngeal, oral and nasal cavities) which acts as a filter and shapes the voice quality creating specific formants (Fitch, 2000).

After having highlighted the importance, the complexity and the specificity of speech as an interpersonal aspect of interactions, the second variable of interest for this study will be discussed in the next section.

1.3 Intra-Personal Aspects: Emotion

Intra-personal aspects of interaction take place within individuals and as for interpersonal behaviors are of a multidimensional nature. The task of defining these aspects is somewhat more delicate and should include, on a cognitive level, subjective experience, but also changes on physiological and neurological levels. To start, let's mention that emotion should not be mixed up with other types of affective states as mood (i.e., more diffuse state), interpersonal stances (e.g., distant, cold and warm), attitudes (i.e., affectively colored beliefs) or personality traits (i.e., stable personality dispositions) (Scherer, 2003).

Even though a lack of agreement in the scientific community about how to define emotion has been reported (Barret, 2006; Cole et al., 2004), numbers of

theories got strong support in specific areas of research. In connection with speech and emotion, the component model of affective states proposed by Scherer (1986) is especially relevant and is also broadly recognized and used (Mauss & Robinson, 2009). In line with this theory, any event or signal perceived is first evaluated in terms of its significance for survival and well-being (Scherer, 1986; Cole et al., 2004). This process is called appraisal and gives rise to different emotional responses, such as subjective experiences, activation of the nervous systems and behaviors (Mauss & Robinson, 2009). The distinction between emotions and appraisal remains unclear, because emotions could be also considered as biologically prepared capabilities preceding and organizing the appraisal process (Cole et al., 2004).

Nevertheless, emotion can be considered as a rapid and fluid process that can be periodically perceived by a person and then apprehended as a subjective experience or a feeling (Cole et al., 2004). In that view, emotion is a series of interrelated adaptive changes that take place in several subsystems of the organism, with each change being simultaneously influenced by the previous one and influencing the next (Scherer, 1986). Facing a quick and continuous process, researchers must remain conscient that its nature involves variations often beyond the detection's level of chosen measures (Cole et al., 2004).

These series of interrelated adaptive changes subjectively perceived as emotional entities can be labeled by a person in different ways, the most commonly studied being discrete emotions (e.g., joy, anger, disgust, sadness, fear and surprise). This approach collected strong support, but no consensus exists on a number of questions about discrete emotions, such as how many emotions exist and if they are at all differentiable. An interesting question concerns the existence of more fundamental psychological processes that could be of a better fit to support scientific induction of emotion (Barret, 2006). Following that thought, discrete emotions could be described by a combination of different dimensions reflecting these underlying psychological processes. For this study, it is also relevant to mention that, concerning self-reported emotional states, dimensions seem to better explain the variability observed than discrete categories of emotion (Mauss & Robinson, 2009).

Three main dimensions can be of interest here and are commonly reported as important to explain emotional states (Mauss & Robinson, 2009). The first,

arousal, concerns the level of activation or intensity reported. For example, a low arousal could correspond to being quiet or bored, and a high arousal to being surprised or angry. The second, valence, often coupled with activation to describe discrete emotions, concerns the level of pleasure or displeasure reported. Valence results from the process of valuation which represent the meaning analysis of a stimulus judged as helpful or harmful (Barret, 2006). This second dimension could be considered as a basic building block of emotional life, also called a core affective state (Barret, 2006). The third, the motivational dimension of approach-avoidance, concerns the tendencies to approach (e.g., facilitated by excitement) or avoid a given stimulus (e.g., facilitated by anxiety) (Barret, 2006). Other dimensions, that are more or less related to the third one, are also reported, as power or control (Scherer, 2003), but for the purpose of this study, they will not be introduced in further details.

This section has focused on different elements of theory helping to define and understand how the concept of emotional valence is approached in the present study. In the next section the focus will be put on how speech can express emotions as it commonly appears in the literature.

1.4 Vocal Expression of Emotion

Emotions are an important dimension of interactions that initiate responses through changes in physiology, thoughts and expressive behaviors, and communicate signals that coordinate social contacts (Keltner et al., 2019). Within meaningful relationships, discussions are regulated through emotion expression, as speech, which mirror physiological and cognitive changes. These emotional expressions can be considered as multimodal and dynamic patterns of behavior, either of an interpersonal (e.g., movements of the face, eyes, body parts or vocalization) or of an intrapersonal aspect (e.g., autonomic response, scent) (Keltner et al., 2019).

Focusing on speech, a distinction needs to be made between the verbal and the paraverbal dimensions. To clarify this, speech is defined here as qualities apart from the actual verbal content (Juslin & Scherer, 2005). The second part of the well-known expression “it’s not what you said, but how you said it” gives a good example of what is referred to as speech qualities. To support this dichotomization,

researches showed that emotions can be communicated through speech without any verbal content (Cowie et al., 2001). From an evolutionary perspective, speech and mechanisms of sound production developed in a tied manner, independently from the evolution of language (Fitch, 2000).

The idea that emotions influence physiological processes and that these changes have an impact on acoustic characteristics of speech can be traced back to Darwin and can be found in Spencer's law (Juslin & Laukka, 2003). The modulation of the striated musculature activity corresponding to emotional changes is directly linked to the production of vocalization. Following the component process model of emotion (Scherer, 1986), changes in muscle tension are controlled by the somatic nervous system, the autonomic nervous system and other organismic subsystems, which are themself influenced by the process of appraisal. In other words, the muscle units mobilized for vocal expression are controlled in part by emotion through physiological changes.

Considering the physiology of vocal expression of emotions, three systems controlled by autonomic and somatic nervous systems are of a particular interest, namely respiratory (producing air exhaled from the lungs), vocal (phonatory and articulatory apparatus) and resonance systems (vocal tract shape) (Johnston et al., 2001; Scherer & Bänziger, 2004). Nevertheless, speech is not reducible to expression of emotion and changes in pitch, loudness or voice quality also serve to communicate other phonological, syntactic or meaning aspects of verbal communication (Scherer, 1995).

The brunswikian lens model of vocal communication of emotion used by Scherer (2003) proposes a way to decompose the process into an encoding phase, consisting of the vocal expression of emotion described above, and a decoding phase in which the acoustic changes are considered by the listener as cues to speaker affect. During this decoding phase, inferences of other's emotions are made based on internalized representations of the observed speech changes (Scherer, 2003). The current emotional state of the listener is affected by the evaluation process of these internalized representations. Knowing that the resulting emotional changes will affect vocal expression, this explains how partners influence each other's emotional states through speech in an interdependent manner while interacting.

Researches have shown that humans demonstrate high accuracy rates when judging emotional state from the voice alone (Scherer, 1995; Banse & Scherer,

1996). The emotional attribution is significantly associated with different acoustic parameters changes, as pitch range, intensity and speech rate (Scherer, 1995). For example, a narrow pitch, or fundamental frequency, range is often perceived as a sign of sadness, when a wide one is often perceived as a sign of high level of arousal and highly negative emotions (Scherer, 1995). A linear relationship between pitch range and the attributed level of arousal can be observed. Even though humans seemed highly suited to decode vocal expression of emotions, the subjectivity of auditory perception is highly problematic and generates low scores of inter-coder reliabilities (Scherer & Bänziger, 2004). Choosing an automatic sound processing approach to extract acoustic features seems a suited choice to overcome this lack of objectivity.

Vocal expression of emotions has been mostly studied based on discrete emotions (Cowie et al., 2001), valence and arousal being often obtained by converting discrete emotions into these two dimensions. Using an automatic sound processing approach, researches converge supporting the existence of specific acoustic profiles that could differentiate a large number of discrete emotions (Goudbeek & Scherer, 2010). If fundamental frequency and intensity measures are reported to discriminate well between levels of arousal, they often fail when valence is varying, showing the importance of a multidimensional approach (Goudbeek & Scherer, 2010). Valence could be better predicted by voice quality generally (e.g., spectral characteristics) and especially in comparison with pitch (Goudbeek & Scherer, 2010), but a lack of research on valence and incoherent results do not allow to make any strong predictions based on the actual knowledge. The current study proposes to align with the few other researches on vocal expression of emotions based on valence. But rather than convert discrete emotions into dimensional scores, self-reported valence emotional scores are directly measured using the Affect Rating Dial (Ruef & Levenson, 2007), a method described later in the corresponding section.

Vocal expression of emotional valence has been presented in link with a more global perspective on emotion expression and specific categories of acoustic features relevant to this area of research have been discussed. Directly connected to it, a new area of research that focuses on automatic processing of acoustic features emerged in the last decades. In this approach, machine learning is the most commonly used technology and is the one that was chosen for the current study to

explore the prediction accuracy of chosen acoustic features on emotional valence self-rating scores. The machine learning approach will be introduced in the next section.

1.5 Machine Learning

To open this section on machine learning, let's start with a citation that, even its rather enthusiastic form, highlight the complementarity of scientific research and technology development:

> New directions in science are launched by new tools much more than by new concepts. The effect of a concept-driven revolution is to explain old things in new ways. The effect of a tool driven revolution is to discover new things that have to be explained. (Dyson, 1998, as cited in Goldberg et al., 2020 (p. 438))

The relationship of emotional valence, specifically self-reported, and vocal expression being not well documented, machine learning could provide a tool-driven approach to explore this link in a pragmatic and predictive way. In addition to that, when dealing with big data, as for extracted acoustic features, machine learning has been recognized to be a robust technique (Shatte et al., 2019). This computational technology uses self-learning algorithms to improve their performances based on previous experiences (Goldberg et al., 2020), often to predict chosen outcomes. It is important to mention that, compared to commonly used statistical analyses, machine learning does not usually fit to model causal relationships between variables, but can inform on a given model about its predictive performances.

Successful attempts to automating the assessment of aspects of treatment have been conducted using machine learning and natural language processing (i.e., verbal and paraverbal) (Goldberg et al., 2020). For example, in mental health conversational agents, using this technology based on language processing, are already implemented in activities related to medical care (Miner et al., 2017). Domains of mental health has been reported to include speech processing technics for detection and diagnosis of mental problems as suicide ideation, schizophrenia, depression, drugs intake or at-risk patients of Alzheimer's (Shatte et al., 2019;

Kliper et al., 2015). Overall, the potential of machine learning for improving clinical aspects of treatment is collecting an increasing interest among the research community, especially because once an algorithm has been proven to be reliable and helpful, it can be deployed on a broader scale with limited additional costs or human judgment (Goldberg et al., 2020).

More specifically connected to this study, researches using acoustic features to predict emotions exist and show compelling findings (for a review see Swain et al., 2018). An interesting fact is that among the 59 studies reported by Swain (2018), only four of them focuses on emotional dimensions (Quiros-Ramirez et al., 2014; Lee & Narayanan, 2003; Lee & Narayanan, 2005; Grimm et al., 2008) and none used self-reported measures, but rather used scores rated by external trained researchers and calculated after controlling for inter-judge fidelity. The emotion recognition field using new technologies is obviously lacking in studies focusing on emotional dimensions and using self-rating scores. The current study addresses these problems focusing on emotional dimensions with the use of self-reported scores.

1.6 Current Study

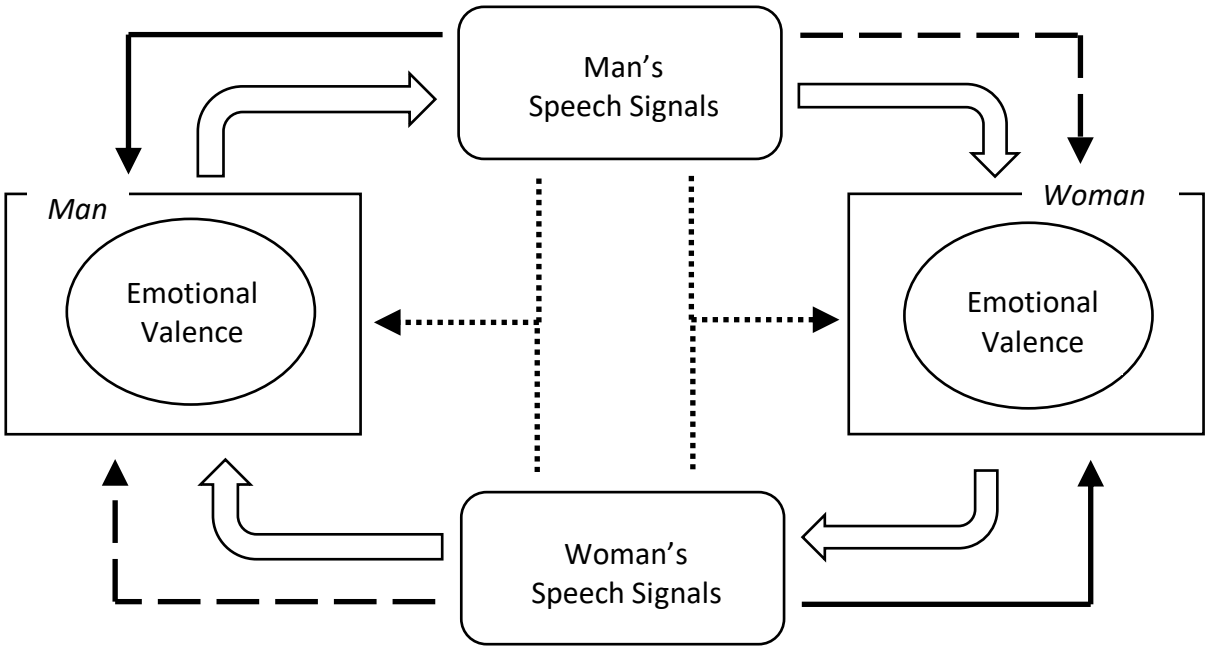
The current study proposes to explore the self-rated emotional valence scores prediction accuracy based on extracted acoustic features of couple interactions using a machine learning approach. The talk-turn level is chosen to extract acoustic features and to run the predictions as it is often used and is recognized as a golden standard. This choice allows a temporal granularity neither to narrow or to wide. Since emotional expression is of a continuous nature, focusing on the process would require extracting vocal features using the smallest measures available (i.e., few microseconds). The problem using such temporal granularity is that, even if it allows to grasp micro changes in speech, the corresponding self-reported feeling measures do not share the same rapidity of variation. That would imply introducing more acoustic different measures for the same emotional score. On the other hand, a too wide temporal granularity would not fit the rapid changing nature of emotions and would not be adequate for this study. Talk turns are by nature varying largely in length, but the ecological validity of this phenomenon

reflects a reality of couple interaction and is a solid argument for backing up this approach.

Partners simultaneously influencing each other, three different models (i.e., own, crossed or mixed predictive models) are first tested to investigate if higher predictive accuracy rates are obtained using exclusively one or both partner's acoustic features to predict either own's or partner's emotional valence scores accuracy (see Figure 1). Different acoustic feature sets are then compared in terms of their accuracy scores. Keeping in mind that this study proposes to explore relationships between speech and emotional valence with the use of machine learning rather than try to test specific assumptions, two main hypotheses can nevertheless be formulated. The set using all acoustic features should predict emotional valence with the highest accuracy performances above all, because it contains more features than all others and because the amount of data is known to often increase prediction accuracy (H1). Four other sets are focusing on the four main acoustic categories, pitch, loudness, durational characteristics and voice quality. As suggested by Goudbeek and Scherer (2010), valence should have a stronger impact on the voice quality (i.e., spectral features) than on pitch characteristics. Therefore, the voice quality set should predict emotional valence with higher accuracy than the pitch set (H2).

Figure 1

Simplified Vocal Emotion Communication System with the Three Predictive Models



Note. Basic vocal emotion communication process for couple interaction is represented with empty double arrows showing the interdependence of the two partners (a man on the left and a woman on the right). The full single arrows represent the own predictive model where own's speech signals influence one's own feelings. The dashed single arrows represent the crossed predictive model where one's speech signals influence partner's feelings. The dotted single arrows represent the mixed predictive model where both partner's speech signals influence one's feelings.

2. Method

In this section, the dataset and preprocessing and then the machine learning's process are described.

2.1 Participants and Procedure

101 Dutch speaking heterosexual couples took part in a Belgium study about Dyadic Interaction (Boateng et al., 2020). Participants were asked to discuss, first, about a negative topic (a highly annoying characteristic of their partner) and then, about a positive topic (a highly valuable one), both times for a 10-minute video and audio taped interaction. Couple were enjoined to wrap up the conversation after eight minutes and, for the first part, to end on good terms. Then, participants were isolated in separate rooms and watched the video recording of their conversation on a computer with the new task of rating their own emotion on a moment-by-moment basis using the Affect Rating Dial (Ruef & Levenson, 2007). Doing so, they had to continuously adjust a joystick from left to right (from very negative to very positive) in order to match their feelings as closely as possible, resulting in emotional valence rating scores.

Trained research assistants encoded manually the recordings in separate talk turns corresponding to different events (i.e., man or woman talking, cross-talk, pause, laughter or noise). Each talk turn's timestamp was marked based on auditory and visual perceptions. 197 files (99 for the first part of the conversation with a negative topic and 98 for the second part with a positive topic) have been created containing starting and ending timestamps for each talk turn plus a label describing the type of event. A total of 82'022 talk turns have been encoded with a talk turns average per file of 416 (for more details see Table 1). In a total of more than 32 hours of recording, one third are represented by women talking and another one third by men talking (for more details see Table 2).

200 audio files with a length of 10 minutes have been extracted from the video and converted into Waveform Audio File Format (WAV) using the python's library *audiofile*. The lowest frequency of the human voice being equal to 80 Hz, a high pass filter, suppressing low frequencies, with a threshold of 80 Hz has been applied to avoid unnecessary noises. The use of a low pass filter, to suppress higher frequencies, has not been used in order to preserve the spectral characteristics.

Table 1

Descriptive Statistics for Number of Talk Turns

Number of talk turns	Total	M	SD	Min	Max
All events	82022	416.36	78.40	153	625
Man talking	20148	102.27	22.71	43	164
Woman talking	20134	102.20	25.74	36	173

Table 2

Descriptive Statistics for Duration of Talk Turns

Duration of talk turns	Total	M	SD	Min	Max
Man talking	10.52	1.88	2.00	0.01	87.16
Woman talking	11.06	1.98	2.11	0.01	58.38

Note. Totals are given in hours. All other statistics are given in seconds.

2.2 Measures

2.2.1 Emotional Valence Rating Scores (EVRS) - Dependent Variables

EVRS were measured on a continuous scale from -1 to 1 with a sampling rate of one per second. Valence refers to how negative to positive the person feels (Boateng et al., 2020) and should not be mixed with arousal, referring to how active and engaged a person feels. Scores have been separated into positive situations (i.e., the 10-minute discussion with positive topic) and negative situations (i.e., the 10-min discussion with negative topic). 200 different files have been created, but 10 have been removed (four had missing values and six had extreme and problematic values). A total of 190 files have been kept (see Table 3). As we can see in the Table 3, EVRS are generally slightly lower in women than in men and also lower in negative than in positive situations.

Table 3

Descriptive Statistics for Emotional Valence Rating Scores (EVRS)

Duration of talk turns	M	SD	Min	Max
Positive discussions				
Men	0.37	0.20	-0.10	0.91
Women	0.35	0.20	-0.06	0.89
Negative discussions				
Men	0.14	0.23	-0.62	0.81
Women	0.11	0.22	-0.41	0.82

To solve the problem resulting from discrepancies in timestamps between talk turns and EVRS, an interpolation function has been used to predict EVRS for each talk turn. A spline function was chosen after testing it by predicting EVRS at -0.5 and at +0.5 second. Predicted values were then compared to originals at 0.0 second and reports showed strong accuracy scores (for more details, see Annex 1, and for the script, see Annex 2).

Another problem emerged related to variations of talk turns durations. The sampling rate of EVRS being of one per second, the following decision algorithm has been selected to interpolate EVRS for talk turns longer than 1 seconds and containing then potentially more than 1 corresponding EVRS. If talk turns durations are lower than two seconds, a unique timestamp, adding the starting time to the half of the talk turn's length, is chosen to interpolate the EVRS. If talk turns durations are higher than two seconds, starting and ending time as well as every additional second fitting in-between generates a list of timestamps. For each timestamp in this list, a corresponding EVRS is interpolated. The resulting EVRS for these talk turns is equal to the average of interpolated EVRS (for the script, see Annex 3, starting at end of p. 46).

EVRS being ranging from -1 to 1 by definition, if the interpolated emotion ratings exceeded 1 in absolute value, for positive and negative values respectively, the value 1 or -1 has been used instead. Then EVRS were dichotomized between 0, equal to negative scores, and 1, equal to scores higher or equal to 0, thus allowing the use of classifier algorithms.

2.2.2 Acoustic Features - Independent Variables

The open-source library for Python *openSMILE* (for open-source Speech and Music Interpretation by Large feature-space Extraction) was used for the audio extraction (Eyben et al., 2010). *openSMILE* was designed to be employed by researchers and system developers and, from the year of its public release, in 2009, is a widely used toolkit in different research fields, like in psychology (see for example Faurholt-Jepsen et al., 2021; Li et al., 2021; de Boer et al., 2021). The extraction algorithms can provide thousands of different acoustic features. Besides the impressive quantity of information that could be provided, the use of such a number of variables can be problematic and the reason is twofold. First, extracting such a high number of acoustic features has often the consequence that researchers do not use the same set of variables making comparison between studies almost impossible and then, slowing the cumulation of empirical evidence (Eyben et al., 2016). Secondly, with machine learning, the use of brute-forced (i.e., including as many variables as possible) could lead to over-adaptation of the model and limit generalization on unseen data (Eyben et al., 2016). To address these problems, a minimalistic and systematic use of acoustic features has been proposed with the

Geneva Minimalistic Acoustic Parameter Set (GeMAPs) which has been conceived specifically for psychologists (Eyben et al., 2016). GeMAPs is included in the *openSMILE* toolkit and was chosen to extract acoustic features in this study.

Having explained the choice of the extraction toolkit, the actual sound extraction process is now described. After combining available files between EVRS, audio and talk turns, a total of 168 files (86 for negative and 82 for positive discussions) have been successfully extracted (for the script, see Annex 3). 58 separate audio features, consisting of low-level descriptors, functionals and temporal features, were created for each talk turn. Arithmetic means and coefficients of variation, among other functionals, were generated for three categories of parameters (i.e., frequency, amplitude and spectral balance) and temporal features (for descriptive statistics of a feature's selection, see Table 4; for more details about all features and sets, see Annex 6, p. 54).

2.3 Statistical Analysis

Cleaning data sets is an important prior step to using machine learning and help optimizing performances. Only talk turns for men and women with a fundamental frequency (F0) higher than zero and without missing values have been kept. To avoid future problems concerning the size of test and training sets regarding the number of iterations, only files containing more than 40 talk turns for men or women were kept (for the script, see Annex 4). After data cleaning, a total of 155 files (79 for the negative discussion and 76 for the positive) still remained in the study.

In line with often used models in the literature, three models have been selected: Balanced Random Forest (BRF), Support Vector Machine with the radial basis function and K-Nearest Neighbors. After running multiple experiments on the mixed predictive model with all acoustic features, BRF performed the best overall and was selected for all future analysis. *Scikit-learn* (Pedregosa et al., 2011) and *Imbalanced-learn* (Lemaître et al., 2017) libraries for Python have been used to run machine learning models and the default values of the hyperparameters, which are known to be robust for the Random Forest Model (Probst et al., 2019), have been kept.

Dealing with 155 files and two different ones, when available, for each couple (i.e., one for positive and one for negative discussions), a customized function has been created to generate the data sets before the split for train and test sets. This function allows a given number of different talk turns (i.e., a number of iteration) to be randomly chosen among each file (for the script, see Annex 5). The minimal number of talk turns for women or men per file being of 40, no more than 30 iterations have been used to avoid having not enough talk turns or exactly the same ones each time. 10, 20 or 30 iterations have been tried for the different models. The resulting subsampled sets are then split into training and test sets with a ration of 0.25 for the test set.

EVRS having been binarized and the resulting classification being unbalanced for positive and negative emotional valence, a metric balanced accuracy has been utilized to assess performances. The formula for balanced accuracy is given bellow. The final performance results consist of average and standard deviation of 100 balanced accuracies scores based on the same model but with each time a new subsampled set.

$$balanced\ accuracy = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

At the end, all possible combinations of iterations (i.e., 10, 20 or 30), predictive models (i.e., own, crossed or mixed), discussion types (i.e., positive, negative or all), feature sets (i.e., all, durational, loudness, pitch or spectral) of sex (i.e., men or women) have been tried out and a total of 270 different average balanced accuracies have been produced and analyzed (for the script, see Annex 6).

3. Results, Discussion, Future Research

3.1 Results

A total of 270 variations using BRF has been processed and for each one, average balanced accuracy (BA) has been generated. The overall best BA is of 58.74% for predicting women's emotional valence with the mixed predictive model using 30 iterations, negative discussions and the durational feature set. Table 5 shows the best performances filtered by variables. The number of iterations increases the best performances modestly, only by 0.29%. Results using only positive discussions have the lowest BA (55.28%) and do not differ from chance, but performances are better using only negative (58.74%) or both (58.43%) discussions. All best performances for feature sets above 58%, are all significatively different from chance and do not differ between them. The mixed predictive model, which uses both partners' acoustic features, is the best one and the only one that differs significantly from chance. The partner's sex does play a role, emotional valence is better predicted for women (58.74%) than for men (54.98%).

3.2 Discussion

The goal of the current study is to investigate the link between speech signals and emotional valence using a machine learning approach. Prediction performances are reported in the literature to be inconsistent and especially low for this dimension, but specific acoustic features could work better than others for this task. Using different predictive models and acoustic feature sets, the hope of this study was to obtain strong prediction accuracies and to observe differences among acoustic features. Unfortunately, the results in this study, even still different from chance, show low prediction accuracies and no significant differences have been found between acoustic features, infirming both previous hypotheses. No benefits in performance have been gained using the presented approach.

Discussing the quality of accuracy for emotion recognition is a complicated task and there is no clear standard or procedure. In the literature, accuracy scores for valence based on speech range from bellow 50%, meaning that the use of algorithms is less effective than flipping a coin, to over 90% (Schuller et al., 2009; Swain et al., 2018). In this study, the best accuracy score being of 59%, it can be stated that the chosen models performed poorly. Nevertheless, the highlighted

results in Table 5 are still significantly different from chance and, knowing that valence is harder than other emotional dimensions to predict, the obtained predictions are considered as valuable. Compared with another study (Eyben et al., 2013), the current one presents similar accuracy scores. The differences in model variations will be now discussed per variable in more details.

Augmenting the number of iterations through each file increase the accuracy slightly, but not significantly after 20 iterations. Processing power needed to run predictions increasing by the amount of data provided, using not more iterations that is needed can save a lot of processing time. For similar conditions as this experimentation, starting the model with 20 iterations per file for sampling could be a good start. It is useless to say, that, the minimum number of talk turns for men or women should be taken into account. Having used a minimal cut-off of 40 talk turns and running the model 100 times, it could be problematic to use 40 or more iterations to run the predictions, because the exact same talk turns would be selected each time not allowing them to be randomly picked anymore.

Acoustic features extracted from positive discussions do not predict significantly EVRS better than chance. Negative discussions are better suited for this task. An explanation could be that, even the average of EVRS for both positive and negative situations is higher than 0, for negative situations, the average of EVRS is lower and nearer to zero. It does play an important role, because EVRS are classified into positive (higher than zero) and negative (lower than zero) emotional valence before running the models. It follows that the binarizing classification process of EVRS leads to include more negative deviations shifts as negative for negative situations, than for positive situations. Another explanation is based on the fact that conflict interactions could engage more partners in the conversation, resulting on wider behavioral responses. As showed in Table 3, minimal and maximal scores obtained during negative discussions demonstrate a wider range than the one obtained during positive discussions. These more extreme emotional shifts could therefore have a bigger impact on acoustic features which could explain why accuracy scores are higher for negative than positive situations.

A surprising result is that acoustic features perform all equally, which invalidate both assumptions that, first, using all acoustic features should show the highest accuracy scores (H1) and that, secondly, the voice quality set should perform better than the pitch set (H2). The results from Goudbeek & Scherer

(2010), showing that intensity, spectral and durational characteristics of speech are more correlated to emotional valence than pitch, could not have been translated and reproduced here. Even correlation and prediction are not of the same nature, it could have been rationally expected that it should be also observed using a machine learning approach. Maybe the fact that the models show overall poor results may play a role, in the sense that they do not allow variations to be good captured limiting the observation of differences among acoustic features. Cultural differences specific to Dutch culture and language could also explain that the results are not as well predictable as in other countries. The so called "pull effect" (i.e., external factors that affect emotion expression, such as "social display rules") as opposed to "push effect" (i.e., physiological changes) (Scherer, 1995), may here be helpful to explain the lack of differences between the acoustic sets. If Dutch couples would be found to regulate their feelings or to express them in a culturally specific way, that could mean that speech signals may be less correlated to partner's emotional valence. Another explanation could come from the same problem encountered for discrete emotion recognition, when different emotions are correlated to the same changes in speech signals. For example, happiness and anger, which could be categorized as having respectively positive and negative valence, have been both reported to correlate with higher pitch average, wider pitch range and higher intensity (Scherer, 1986; Cowie et al., 2001). Similar impacts on speech signals from different emotional states, could diminish the predictive accuracy of models using only one dimension.

The mixed predictive model, using both partner's speech signals to predict EVRS, works the best and is the only one having accuracy scores differing significantly from chance. These results show that using acoustic features for both partners leads to better performances than to treat them separately. The fact that using one partner speech signals to predict his or her own emotional valence (own predictive model) do not differ from predicting other's emotional valence (crossed predictive model) can be explained by the choice of the talk turn as a temporal frame for emotion. Knowing that emotional changes are of a continuous nature, it can be assumed that during a talk turn, micro emotional changes occurring within the speaker as well as within the listener are to be captured. Meaning that emotional changes either expressed through speech or being the result of the impact of the speech on both partners are included in EVRS on a talk turn level. The temporal

granularity being not precise enough to differentiate between them, this could explain why no significant differences have been observed between best performances of own and crossed predictive models.

Only predictions for women shows significant accuracy scores. In emotion recognition, women have been reported to present higher accuracy rate than men (Swain et al., 2018), meaning that their emotions could be better predicted than for men. An explanation could be that women and men differ biologically impacting the process of vocal expression, as for example having higher pitch range for women than for men. Among many other known differences in psychology between men and women, emotion regulation could be particularly relevant here. Differences in emotional response strategies between men and women (see for example Nolen-Hoeksema, 2012) will have an impact on emotion expression and therefore on speech signals. Even better understanding of this link could be of importance, these differences go beyond the reach of the current study and will not be discussed further.

After having discussed the results, more general limitations will be introduced now. It is a difficult task to compare different results in emotion recognition research area, because the procedure and the nature of emotion differ strongly between studies. For example, some studies are using emotive (i.e., produced by actors) and others emotional (i.e., spontaneously produced) expressions of emotional states (Banse & Scherer, 1996). Even more, often, dimensional emotions are being generated based on discrete emotion data sets which differ in method from directly measuring arousal or valence. Adding to the diversity of these approaches, the methodological differences between emotions having been coded by an external person and self-reported measures of emotion can also generate noise and confusion when trying to compare different studies. Therefore, all these limitations have to be taken into account for each attempt to generalize the current results.

3.3 Future Research

For further research, different elements can be changed to improve the current approach. First, it is important to remember that speech is only one modality of emotion expression among others occurring during interactions. Combining

linguistic with acoustic dimensions often shows better results (for example see Lee & Narayanan, 2005). Building a multimodal approach for emotion recognition (e.g., combining linguistic, facial and vocal information) is certainly a new challenge for future research and can help improving performances. But such an approach has to be done carefully, because performances do not improve only regarding the richness of the data (i.e., the more modalities that are included), but also its design, including setting up machine learning models (for example see Fragopanagos & Taylor, 2005).

Secondly, using a smaller temporal frame, than the talk turn level, to measure emotional states that match the EVRS sampling rate could more effectively catch smaller variations and shifts in emotional state, especially for extremely long talk turns (see the maximum duration of talk turns in Table 4). Even talk turn level is a golden standard and can be ecologically justified, this choice implies also a loss of information. Concretely, for talk turns during more than 2 seconds, an average of EVRS has been generated and, therefore, variations have been flattened. Dividing a talk turn on samples smaller or equal to two seconds could describe in a richer way the emotional process and generate better performances.

Thirdly, if the objective is to get the highest prediction accuracy and not trying to explore the relationship between speech and emotion, a more brute force approach using a larger number of acoustic features may lead to better results, but they will be more complicated to generalize.

Finally, tuning machine learning models and using more complex algorithms, such as neural networks (see for example Issa et al., 2020), may also show better performances, but with the risk of overfitting the data and not being helpful on another dataset.

4. Conclusion

Emotion recognition is a new field in psychology showing great progress and with many applications in the domains of mental health. Only few studies using a machine learning approach tried to predict emotional valance based on speech signals. In this area of research, results show often low prediction performances and incoherent results. The current study is one of the few that used self-reported emotional valence scores and had the goal to better understand the link between speech signals and emotion. Even using 270 different model variations focusing on emotional valence prediction, the best accuracy was low (59%), but significantly different from chance. Unfortunately, further analysis comparing different acoustic features were not able to highlight any significant differences among them. Vocal emotional valence prediction has still yet to be better apprehended and the key concept for future researches may well be a multimodal approach, combining speech signals, with other observations, such as linguistic information, or with other dimensions of emotion, such as arousal and motivational dimensions. The use of much larger sets of features may also be of help to get better prediction accuracies, but with the cost of failing to better understand the link between speech signals and emotion.

References

Banse, R., & Scherer, K. R. (1996). Acoustic profiles in vocal emotion expression. *Journal of Personality and Social Psychology*, *70*(3), 614–636. https://doi.org/10.1037/0022-3514.70.3.614

Barrett, L. F. (2006). Valence is a basic building block of emotional life. *Journal of Research in Personality*, *40*(1), 35–55. https://doi.org/10.1016/j.jrp.2005.08.006

Belyk, M., & Brown, S. (2014). The Acoustic Correlates of Valence Depend on Emotion Family. *Journal of Voice*, *28*(4), 523.e9-523.e18. https://doi.org/10.1016/j.jvoice.2013.12.007

Black, M., Katsamanis, A., Lee, C. C., Lammert, A. C., Baucom, B. R., Christensen, A., ... & Narayanan, S. S. (2010, September). Automatic classification of married couples' behavior using audio features. In *Eleventh annual conference of the international speech communication association* (pp. 2030-2033). ISCA.

Black, M. P., Katsamanis, A., Baucom, B. R., Lee, C. C., Lammert, A. C., Christensen, A., Georgiou, P. G., & Narayanan, S. S. (2013). Toward automating a human behavioral coding system for married couples' interactions using speech acoustic features. *Speech Communication*, *55*(1), 1–21. https://doi.org/10.1016/j.specom.2011.12.003

Boateng, G., Sels, L., Kuppens, P., Hilpert, P., & Kowatsch, T. (2020, October). Speech emotion recognition among couples using the peak-end rule and transfer learning. In *Companion Publication of the 2020 International Conference on Multimodal Interaction* (pp. 17-21). ICMI.

Butler, E. A. (2011). Temporal Interpersonal Emotion Systems. *Personality and Social Psychology Review*, *15*(4), 367–393. https://doi.org/10.1177/1088868311411164

Cole, P. M., Martin, S. E., & Dennis, T. A. (2004). Emotion Regulation as a Scientific Construct: Methodological Challenges and Directions for Child Development

Research. *Child Development*, *75*(2), 317–333. https://doi.org/10.1111/j.1467-8624.2004.00673.x

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., & Taylor, J. (2001). Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, *18*(1), 32–80. https://doi.org/10.1109/79.911197

de Boer, J. N., Voppel, A. E., Brederoo, S. G., Schnack, H. G., Truong, K. P., Wijnen, F. N. K., & Sommer, I. E. C. (2021). Acoustic speech markers for schizophrenia-spectrum disorders: a diagnostic and symptom-recognition tool. *Psychological Medicine*, 1–11. https://doi.org/10.1017/s0033291721002804

Eyben, F., Wöllmer, M., & Schuller, B. (2010, October). Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia* (pp. 1459-1462). ACM.

Eyben, F., Weninger, F., & Schuller, B. (2013). Affect recognition in real-life acoustic conditions-a new perspective on feature selection. In *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association* (pp. 2044-2048). ISCA.

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., Andre, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., & Truong, K. P. (2016). The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE Transactions on Affective Computing*, *7*(2), 190–202. https://doi.org/10.1109/taffc.2015.2457417

Faurholt-Jepsen, M., Rohani, D. A., Busk, J., Vinberg, M., Bardram, J. E., & Kessing, L. V. (2021). Voice analyses using smartphone-based data in patients with bipolar disorder, unaffected relatives and healthy control individuals, and during different affective

states. *International Journal of Bipolar Disorders*, *9*(1). https://doi.org/10.1186/s40345-021-00243-3

Fitch, W. (2000). The evolution of speech: a comparative review. *Trends in Cognitive Sciences*, *4*(7), 258–267. https://doi.org/10.1016/s1364-6613(00)01494-7

Fragopanagos, N., & Taylor, J. (2005). Emotion recognition in human–computer interaction. *Neural Networks*, *18*(4), 389–405. https://doi.org/10.1016/j.neunet.2005.03.006

Friedlander, M. L., Lee, M., & Escudero, V. (2019). What we do and do not know about the nature and analysis of couple interaction. *Couple and Family Psychology: Research and Practice*, *8*(1), 24–44. https://doi.org/10.1037/cfp0000114

Goldberg, S. B., Flemotomos, N., Martinez, V. R., Tanana, M. J., Kuo, P. B., Pace, B. T., Villatte, J. L., Georgiou, P. G., van Epps, J., Imel, Z. E., Narayanan, S. S., & Atkins, D. C. (2020). Machine learning and natural language processing in psychotherapy research: Alliance as example use case. *Journal of Counseling Psychology*, *67*(4), 438–448. https://doi.org/10.1037/cou0000382

Gottman, J. M., & Notarius, C. I. (2000). Decade Review: Observing Marital Interaction. *Journal of Marriage and Family*, *62*(4), 927–947. https://doi.org/10.1111/j.1741-3737.2000.00927.x

Goudbeek, M., & Scherer, K. (2010). Beyond arousal: Valence and potency/control cues in the vocal expression of emotion. *The Journal of the Acoustical Society of America*, *128*(3), 1322. https://doi.org/10.1121/1.3466853

Grimm, M., Kroschel, K., & Narayanan, S. (2008, June). The Vera am Mittag German audio-visual emotional speech database. In *2008 IEEE international conference on multimedia and expo* (pp. 865-868). IEEE.

Heyman, R. E. (2001). Observation of couple conflicts: Clinical assessment applications, stubborn truths, and shaky foundations. *Psychological Assessment*, *13*(1), 5–35. https://doi.org/10.1037/1040-3590.13.1.5

Issa, D., Fatih Demirci, M., & Yazici, A. (2020). Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control*, *59*, 101894. https://doi.org/10.1016/j.bspc.2020.101894

Johnstone, T., van Reekum, C. M., & Scherer, K. R. (2001). Vocal correlates of appraisal processes. In K. R. Scherer, A Schorr, & T. Johnstone (Eds.). *Appraisal processes in emotion: Theory, Methods, Research* (pp. 271-284). Oxford University Press.

Juslin, P. N., & Laukka, P. (2003). Communication of emotions in vocal expression and music performance: Different channels, same code? *Psychological Bulletin*, *129*(5), 770–814. https://doi.org/10.1037/0033-2909.129.5.770

Juslin, P. N., & Scherer, K. R. (2005). Vocal expression of affect. In Harrigan, J., Rosenthal, R., & Scherer, K. (Eds.). *New handbook of methods in nonverbal behavior research*. Oxford University Press.

Keltner, D., Sauter, D., Tracy, J., & Cowen, A. (2019). Emotional Expression: Advances in Basic Emotion Theory. *Journal of Nonverbal Behavior*, *43*(2), 133–160. https://doi.org/10.1007/s10919-019-00293-3

Kliper, R., Portuguese, S., & Weinshall, D. (2015, September). Prosodic analysis of speech and the underlying mental state. In *International Symposium on Pervasive Computing Paradigms for Mental Health* (pp. 52-62). Springer, Cham.

Lee, C. M., & Narayanan, S. (2003). Emotion recognition using a data-driven fuzzy inference system. In *Eighth European conference on speech communication and technology*. Eurospeech.

Lee, C. M., & Narayanan, S. (2005). Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, *13*(2), 293–303. https://doi.org/10.1109/tsa.2004.838534

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1-5.

Li, J., Yu, J., Ye, Z., Wong, S., Mak, M., Mak, B., ... & Meng, H. (2021, June). A comparative study of acoustic and linguistic features classification for alzheimer's disease detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6423-6427). IEEE.

Liscombe, J. J. (2007). *Prosody and speaker state: paralinguistics, pragmatics, and proficiency,* 8-9 [Doctoral dissertation, Columbia University]. ProQuest Information and Learning Company.

Määttä, K., & Uusiautti, S. (2013). Silence is Not Golden: Review of Studies of Couple Interaction. *Communication Studies*, *64*(1), 33–48. https://doi.org/10.1080/10510974.2012.731467

Mauss, I. B., & Robinson, M. D. (2009). Measures of emotion: A review. *Cognition & Emotion*, *23*(2), 209–237. https://doi.org/10.1080/02699930802204677

Miner, A. S., Milstein, A., & Hancock, J. T. (2017). Talking to Machines About Personal Mental Health Problems. *JAMA*, *318*(13), 1217. https://doi.org/10.1001/jama.2017.14151

Nolen-Hoeksema, S. (2012). Emotion Regulation and Psychopathology: The Role of Gender. *Annual Review of Clinical Psychology*, *8*(1), 161–187. https://doi.org/10.1146/annurev-clinpsy-032511-143109

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of machine Learning research*, *12*, 2825-2830.

Porges, S. W. (2001). The polyvagal theory: phylogenetic substrates of a social nervous system. *International Journal of Psychophysiology*, *42*(2), 123–146. https://doi.org/10.1016/s0167-8760(01)00162-3

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3). https://doi.org/10.1002/widm.1301

Quiros-Ramirez, M. A., Polikovsky, S., Kameda, Y., & Onisawa, T. (2014, June). A spontaneous cross-cultural emotion database: Latin-America vs. Japan. In *KEER2014. Proceedings of the 5th Kanesi Engineering and Emotion Research; International Conference* (No. 100, pp. 1127-1134). Linköping University Electronic Press.

Ruef, A. M., & Levenson, R. W. (2007). Continuous measurement of emotion. In Coan, J. A., & Allen, J. J. (Eds.). *Handbook of emotion elicitation and assessment* (pp. 286-297). Oxford University Press.

Scherer, K. R. (1986). Vocal affect expression: A review and a model for future research. *Psychological Bulletin*, *99*(2), 143–165. https://doi.org/10.1037/0033-2909.99.2.143

Scherer, K. R. (1995). Expression of emotion in voice and music. *Journal of Voice*, *9*(3), 235–248. https://doi.org/10.1016/s0892-1997(05)80231-0

Scherer, K. R. (2003). Vocal communication of emotion: A review of research paradigms. *Speech Communication*, *40*(1–2), 227–256. https://doi.org/10.1016/s0167-6393(02)00084-5

Scherer, K. R., & Bänziger, T. (2004, March). Emotional expression in prosody: a review and an agenda for future research. In *Speech prosody 2004, international conference*. ISCA.

Schuller, B., Vlasenko, B., Eyben, F., Rigoll, G., & Wendemuth, A. (2009, November). Acoustic emotion recognition: A benchmark comparison of performances. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding* (pp. 552-557). IEEE.

Shatte, A. B. R., Hutchinson, D. M., & Teague, S. J. (2019). Machine learning in mental health: a scoping review of methods and applications. *Psychological Medicine*, *49*(09), 1426–1448. https://doi.org/10.1017/s0033291719000151

Swain, M., Routray, A., & Kabisatpathy, P. (2018). Databases, features and classifiers for speech emotion recognition: a review. *International Journal of Speech Technology*, *21*(1), 93–120. https://doi.org/10.1007/s10772-018-9491-z

Table 4

Descriptive Statistics for Selected Acoustic Features

	Positive discussions		Negative discussions	
Acoustic features	Men	Women	Men	Women
F0 semitone (from 27.5 Hz)	24.35 (2.46) 112.25[a]	33.71 (2.98) 192.74 [a]	24.49 (2.5) 113.16 [a]	33.77 (3.13) 193.41 [a]
Loudness	0.26 (0.09)	0.25 (0.09)	0.26 (0.08)	0.25 (0.08)
Jitter	0.03 (0.02)	0.03 (0.02)	0.03 (0.02)	0.03 (0.02)
Shimmer	1.27 (0.32)	1.28 (0.29)	1.28 (0.33)	1.28 (0.28)
Harmonics to noise ratio	4.42 (1.28)	6.84 (1.52)	4.41 (1.30)	6.73 (1.57)
Hammarberg index	28.16 (3.81)	25.89 (4.29)	28.04 (3.83)	25.60 (4.22)
Alpha ratio	-19.28 (3.21)	-16.71 (3.50)	-19.13 (3.31)	-16.49 (3.47)
Voiced segments per second	2.61 (1.06)	2.85 (1.13)	2.66 (1.08)	2.86 (1.15)
Mean voiced segment length per second	0.16 (0.10)	0.15 (0.09)	0.15 (0.09)	0.14 (0.09)
Mean unvoiced segment length per second	0.22 (0.13)	0.20 (0.13)	0.21 (0.13)	0.20 (0.13)

Note. Only means (with standard deviations presented in parentheses) are given.

[a] Fundamental frequency (F0) given in Hz.

Table 5

Best Balanced Accuracies (BBA) filtered per Categories

Category filtered		Iterations	Discussions	Features Set	Predictive Model	Sex	BBA (%)	
							M	SD
Iterations:	10	x	Negative	All	Mixed	Women	58.45	4.04
	20	x	Negative	Pitch	Mixed	Women	58.49*	2.57
	30	x	Negative	Durational	Mixed	Women	58.74*	2.41
Discussions:	All	30	x	All	Mixed	Women	58.43*	2.03
	Negative	30	x	Durational	Mixed	Women	58.74*	2.41
	Positive	20	x	Spectral	Mixed	Women	55.28	4.53
Features set:	All	30	Negative	x	Mixed	Women	58.60*	2.05
	Durational	30	Negative	x	Mixed	Women	58.74*	2.41
	Loudness	30	Negative	x	Mixed	Women	58.31*	2.20
	Pitch	30	Negative	x	Mixed	Women	58.67*	2.22
	Spectral	30	Negative	x	Mixed	Women	58.51*	2.46

Category filtered		Iterations	Discussions	Features Set	Predictive Model	Sex	BBA (%)	
							M	SD
Predictive Model:	Crossed	30	Negative	Loudness	x	Men	54.98	2.99
	Own	30	Negative	Durational	x	Women	54.38	3.43
	Mixed	30	Negative	Durational	x	Men	58.74*	3.43
Sex:	Men	30	Negative	Loudness	Crossed	x	54.98	2.99
	Women	30	Negative	Durational	Mixed	x	58.74*	2.41

Note. To avoid redundancies on the table, x corresponds to the category filtered.

* Significative results different from chance (50%) using a one-tailed test with a $p < .01$.

Annex 1

Metrics for Accuracy of the Spline Function

Score type	Men	Women
R^2 (coefficient of determination)	0.97 (0.03)	0.98 (0.02)
Explained variance	0.97 (0.03)	0.98 (0.02)
Mean squared error	0.00 (0.00)	0.00 (0.00)

Note. All scores are first calculated separately for each file and then all combined per type to generate averages and standard deviations. Standard deviations are given in parentheses. R^2, explained variance and mean squared error are calculated using the formulas given bellow.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$Explained\ variance(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}$$

$$Mean\ squared\ error(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples-1}} (y_i - \hat{y_i})^2$$

Annex 2

```
##############################
##### SPLINE CHECK SCRIPT #####
##############################

### IMPORTING IMPORTANT LIBRARIES AND FUNCTIONS ###

from scipy.interpolate import splev, splrep
import statistics
import pandas as pd
import os
import matplotlib.pyplot as plt
from sklearn.metrics import \
    r2_score, \
    explained_variance_score, \
    mean_squared_error

# defining the path where the 190 files are situated
spline_check_path = "/ PATH TO THE FOLDER CONTAINING THE EVRS FILES /"


### CONVERT VALUES FROM A DATA FRAME TO A LIST ###

def df_values_to_list(data_frame, name_column):
    """convert the values of a given variable (name_column)
    from a given data frame (data_frame) to a list of values."""

    x = data_frame.loc[:, name_column]
    x = x.values
    x = x.tolist()
    return x


### COLLECT THE FILES NAMES FROM A FOLDER ###

def get_files_names_from_folder(folder_path):
    """collect the file names from a folder"""

    path, dirs, files = next(os.walk(folder_path))
    files = [file for file in files if file != ".DS_Store"]
    return files


### GENERATING THE PREDICTIONS AND ACCURACY METRIC SCORES ###

def spline_pred(file):
    """export the spline prediction accuracy
    for men and woman as csv files to the spline_check folder and
    return multiples accuracy metric scores for men and women
    separetly
    """

    # creating the pandas dataframe for a given file
    df_emo = pd.read_csv(f"{spline_check_path}{file}", sep=",")
```

```
### CREATING PREDICTIONS BASED ON A SPLINE FUNCTION ###

# checking for problems in file
if df_emo.isnull().values.any():
    print(f"{file} has some missing datas")
if len(df_emo) > 601 or len(df_emo) < 599:
    print(f"{file} has a length problem")
else:
    # selection of x = time, y = EVRS (h = men and f = women)
    x = df_values_to_list(df_emo, "dialnr")
    y_h = df_values_to_list(df_emo, "emo.m")
    y_f = df_values_to_list(df_emo, "emo.f")

    # creating predicted EVRS for x + 0.5 sec
    x2_plus = [i + 0.5 for i in x]
    # for men
    spl_h1 = splrep(x, y_h)
    y2_h1 = splev(x2_plus, spl_h1)
    df_emo.insert(6, "emo_h1_spline_+0.5(s)", y2_h1)
    # for women
    spl_f1 = splrep(x, y_f)
    y2_f1 = splev(x2_plus, spl_f1)
    df_emo.insert(8, "emo_f1_spline_+0.5(s)", y2_f1)

    # creating predicted EVRS for x - 0.5 sec
    x2_minus = [i - 0.5 for i in x]
    # for men
    spl_h2 = splrep(x, y_h)
    y2_h2 = splev(x2_minus, spl_h2)
    df_emo.insert(5, "emo_h2_spline_-0.5(s)", y2_h2)
    # for women
    spl_f2 = splrep(x, y_f)
    y2_f2 = splev(x2_minus, spl_f2)
    df_emo.insert(8, "emo_f2_spline_-0.5(s)", y2_f2)

    # create a csv file
    df_emo.to_csv(f"/ FOLDER PATH FOR THE FILE /{file}")

    ### CREATING ACCURACY METRIC SCORES ###

    # for men
    r_squared_h1 = \
        r2_score(y_true=y_h, y_pred=y2_h1)
    expl_var_h1 = \
        explained_variance_score(y_true=y_h, y_pred=y2_h1)
    mse_h1 = \
        mean_squared_error(y_true=y_h, y_pred=y2_h1)
    r_squared_h2 = \
        r2_score(y_true=y_h, y_pred=y2_h2)
    expl_var_h2 = \
        explained_variance_score(y_true=y_h, y_pred=y2_h2)
    mse_h2 = \
        mean_squared_error(y_true=y_h, y_pred=y2_h2)

    # for women
```

```
        r_squared_f1 = \
            r2_score(y_true=y_f, y_pred=y2_f1)
        expl_var_f1 = \
            explained_variance_score(y_true=y_f, y_pred=y2_f1)
        mse_f1 = \
            mean_squared_error(y_true=y_f, y_pred=y2_f1)
        r_squared_f2 = \
            r2_score(y_true=y_f, y_pred=y2_f2)
        expl_var_f2 = \
            explained_variance_score(y_true=y_f, y_pred=y2_f2)
        mse_f2 = \
            mean_squared_error(y_true=y_f, y_pred=y2_f2)

        ### RETURNING THE AVERAGE METRIC SCORES FOR THE FILE
        # BASED ON -0.5 and +0.5 SECONDS ###

        return (r_squared_h1 + r_squared_h2) / 2, \
               (expl_var_h1 + expl_var_h2) / 2, \
               (mse_h1 + mse_h2) / 2, \
               (r_squared_f1 + r_squared_f2) / 2, \
               (expl_var_f1 + expl_var_f2) / 2, \
               (mse_f1 + mse_f2) / 2


### GENERAL FUNCTION FRAMING THE WHOLE SPLINE CHECK REPORT ###

def get_report_emo():
    """create a printed accuracy report
    of the use of spline to EVRS"""

    # creating empty lists and other variables
    h_r_squared = []
    h_expl_var = []
    h_mse = []
    f_r_squared = []
    f_expl_var = []
    f_mse = []
    worked = 0
    failed = 0
    nan = 0

    # starting the for loop for all files
    # generating accuracy metric scores
    # and checking for problems

    for file in get_files_names_from_folder(spline_check_path):
        print(file)
        try:
            new_h_r_squared, new_h_var, new_h_mse,\
            new_f_r_squared, new_f_var, new_f_mse \
                = spline_pred(file)
            worked += 1
        except ValueError:
            failed += 1
            pass
        except TypeError:
```

```
            nan += 1
        else:
            # adding to the list the corresponding
            # accuracy metric score for each file
            h_r_squared.append(new_h_r_squared)
            h_expl_var.append(new_h_var)
            h_mse.append(new_h_mse)
            f_r_squared.append(new_f_r_squared)
            f_expl_var.append(new_f_var)
            f_mse.append(new_f_mse)

    ### CREATE THE REPORT ###

    print(f"{worked} worked // {failed} failed "
          f"// {nan} contained NaN\n")

    print("for men")
    print(f"the mean of r_squared = "
          f"{statistics.mean(h_r_squared)} "
          f"| sd = {statistics.stdev(h_r_squared)}")
    print(f"the mean of explained variance = "
          f"{statistics.mean(h_expl_var)} "
          f"| sd = {statistics.stdev(h_expl_var)}")
    print(f"the mean of mean squared error = "
          f"{statistics.mean(h_mse)} "
          f"| sd = {statistics.stdev(h_mse)}\n")

    print("for women")
    print(f"the mean of r_squared = "
          f"{statistics.mean(f_r_squared)} "
          f"| sd = {statistics.stdev(f_r_squared)}")
    print(f"the mean of explained variance = "
          f"{statistics.mean(f_expl_var)} "
          f"| sd = {statistics.stdev(f_expl_var)}")
    print(f"the mean of mean squared error = "
          f"{statistics.mean(f_mse)} "
          f"| sd = {statistics.stdev(f_mse)}")


### LUNCH THE SCRIPT ###

get_report_emo()
```

Annex 3

```
##############################
#####  OPENSMILE  SCRIPT  #####
##############################


### IMPORTING IMPORTANT LIBRARIES AND FUNCTIONS ###

import opensmile
import os
import pandas as pd
from datetime import datetime
from scipy.interpolate import splev, splrep
from statistics import mean

### SETTING PATHS ###

PATH_TT = "/ FOLDER WHERE THE TALK TURNS ARE /"
PATH_AUDIO = "/ FOLDER WHERE THE AUDIO ARE /"
PATH_EMO = " / FOLDER WHERE EVRS ARE /"
PATH_OUTPUTS = " / FOLDER WHERE THE OUTPUTS ARE GONNE BE SAVED /"


### CREATE A DATA FRAME LIST OF TALK TURNS ###

def create_tt_df(path_talk_turns):
    """create a data frame list with all the talk turns csv files"""

    path_folder = path_talk_turns
    # assign path
    path, dirs, files = next(os.walk(path_folder))
    file_count = len(files)
    # create empty list
    dataframes_list = []

    # append datasets to the list
    for i in range(file_count):
        if files[i] == ".DS_Store":
            pass
        else:
            temp_df = pd.read_csv(f"{path_folder}{files[i]}")
            temp_df.name = files[i] # way to keep track of the df
            dataframes_list.append(temp_df)
    return dataframes_list


### CREATE A LIST FROM A DATA FRAME'S COLUMN ###

def df_values_to_list(data_frame, name_column):
    """convert the values of a given variable (name_column)
    from a given data frame (data_frame) to a list of values."""

    x = data_frame.loc[:, name_column]
    x = x.values
```

```
    x = x.tolist()
    return x


### CREATE THE SPLINE PREDICTION FOR EVRS ###

def spline_time_profile_tt(path_file_emo_in, path_file_csv_outputs):
    """export the spline representation of a given x and y /
    calculating a mean of the emo ratings if the talk turn > 2 seconds"""

    # create a data frame from EVRS
    df_emo = pd.read_csv(path_file_emo_in, sep=",")
    # create a data frame from opensmile extracted outputs
    df_output = pd.read_csv(path_file_csv_outputs, sep=",")

    # creating variables with time info
    x2 = df_values_to_list(data_frame=df_output,
                           name_column="start_sec")
    duration = df_values_to_list(data_frame=df_output,
                                 name_column="duration_tt_sec")
    # creating a variables with time in the middle of the talk turn
    x2 = [x2[i] + duration[i] / 2 for i in range(len(x2))]

    ### FOR MEN ###

    # creating the spline representation
    x_h = df_values_to_list(data_frame=df_emo,
                            name_column="time")
    y_h = df_values_to_list(data_frame=df_emo,
                            name_column="emo_h")
    spl_h = splrep(x_h, y_h)

    # creating the spline prediction
    # using the time in the middle of the talk turn
    y2_h = splev(x2, spl_h)

    # checking if the predictions are > 1
    has_to_be_checked = False
    selected_index = 0
    selected_index_list = []
    for item in y2_h:
        if item > 1:
            print(f"{path_file_emo_in}: h = {item} "
                  f"| idx = {selected_index}")
            has_to_be_checked = True
            selected_index_list.append(selected_index)
        selected_index += 1

    # replacing everything > or < 1 or 1 by 1 or -1
    y2_h = [1 if val > 1 else val for val in y2_h]
    y2_h = [-1 if val < -1 else val for val in y2_h]

    # checking if it worked
    if has_to_be_checked:
        for idx in selected_index_list:
            print(f"h = ALL GOOD NOW : {y2_h[idx]} | {idx}!!!")
```

```
### FOR WOMEN ###

# creating the spline representation
x_f = df_values_to_list(data_frame=df_emo,
                        name_column="time")
y_f = df_values_to_list(data_frame=df_emo,
                        name_column="emo_f")
spl_f = splrep(x_f, y_f)

# creating the spline prediction
# using the time in the middle of the talk turn
y2_f = splev(x2, spl_f)

# checking if the predictions are > 1
has_to_be_checked = False
selected_index = 0
selected_index_list = []
for item in y2_f:
    if item > 1:
        print(f"{path_file_emo_in}: f = {item} "
              f"| idx = {selected_index}")
        has_to_be_checked = True
        selected_index_list.append(selected_index)
    selected_index += 1

# replacing everything > or < 1 or 1 by 1 or -1
y2_f = [1 if val > 1 else val for val in y2_f]
y2_f = [-1 if val < -1 else val for val in y2_f]

# checking if it worked
if has_to_be_checked:
    for idx in selected_index_list:
        print(f"f = ALL GOOD NOW : {y2_f[idx]} | {idx}!!!")

### INSERT NEW PREDICTED EVRS ###

df_output.insert(8, "emo_h", y2_h)
df_output.insert(9, "emo_f", y2_f)

### CALCULATE AVERAGE OF PREDICTED EVRS
# IF TALK TURNS > 2 SECONDS ###

# selecting rows > 2 seconds
index_long_rows = \
    df_output.index[df_output["duration_tt_sec"] > 2].tolist()

# for loop calculating average of predicted EVRS
for row in index_long_rows:
    new_row = df_output.iloc[row]
    start = new_row["start_sec"]
    duration = new_row["duration_tt_sec"]

    # expend the timestamps for talk turn
    # to calculate the average predicted EVRS
    expended_timestamps = [start + x
```

```
                                    if start + x <= start + duration
                                    else start + duration
                                    for x in range(0, round(duration + 1))]

            # average predicted EVRS per extra sec for men
            interpolation_emo_h = \
                splev(expended_timestamps, spl_h)
            interpolation_emo_h = \
                [1 if val > 1 else val for val in interpolation_emo_h]
            interpolation_emo_h = \
                [-1 if val < -1 else val for val in interpolation_emo_h]
            mean_emo_h = \
                mean(interpolation_emo_h)

            # average predicted EVRS per extra sec for women
            interpolation_emo_f = \
                splev(expended_timestamps, spl_f)
            interpolation_emo_f = \
                [1 if val > 1 else val for val in interpolation_emo_f]
            interpolation_emo_f = \
                [-1 if val < -1 else val for val in interpolation_emo_f]
            mean_emo_f = \
                mean(interpolation_emo_f)

            # replace old EVRS by the average predicted EVRS
            df_output.at[row, "emo_h"] = mean_emo_h
            df_output.at[row, "emo_f"] = mean_emo_f

    # create a csv file with the average predicted EVRS
    df_output.to_csv(f"{path_file_csv_outputs[:-4]}_mean.csv")


### OPENSMILE EXTRACTION ###

def opensmile_talk_turns(list_tt_df,
                         path_audio=PATH_AUDIO,
                         emo_path=PATH_EMO,
                         path_outputs=PATH_OUTPUTS):
    """opensmile extraction using GeMAPSv01b of a list
    of wav files corresponding to the given list
    of talk turns csv files /
    it only gives the means per feature per talk turn /
    => it creates 5 extra columns :
    1) timestamp in sec for start
    2) timestamp in sec for end
    3) duration of the all talk turn
    4) label corresponding to the event
    5) index of the talk turn
    => a file ending with : 'opensmile.csv' is created"""

    # seting up opensmile set and level
    smile = opensmile.Smile(
        feature_set=opensmile.FeatureSet.GeMAPSv01b,
        feature_level=opensmile.FeatureLevel.Functionals,
    )
```

```
    # starting the for loop to create a pandas dataframe
    for df in list_tt_df:
        current_time1 = datetime.now()
        y = pd.DataFrame()
        # removing the format .csv from the date frame name
        new_name = df.name[0:len(df.name) - 4]
        nbre_talk_turn = 1

        # creating rows for each talk turn
        for k in range(len(df)):
            row = df.iloc[k]
            start_time = row["start.tt"]
            end_time = row["end.tt"]
            duration = end_time - start_time
            label = row["label"]
            audio = f"{path_audio}{new_name}.wav"
            new_df = smile.process_file(audio,
                                        start=start_time,
                                        end=end_time)
            new_df.insert(loc=0, column="start_sec",
                          value=start_time)
            new_df.insert(loc=1, column="end_sec",
                          value=end_time)
            new_df.insert(loc=2, column="duration_tt_sec",
                          value=duration)
            new_df.insert(loc=3, column="label",
                          value=label)
            new_df.insert(loc=4, column="tt",
                          value=round(nbre_talk_turn))
            y = y.append(new_df)
            nbre_talk_turn += 1

        # creating a csv file from the data frame
        name_path_output = f"{path_outputs}{new_name}_opensmile.csv"
        y.to_csv(name_path_output)

        # replacing predicted EVRS using the spline function

spline_time_profile_tt(path_file_emo_in=f"{emo_path}{new_name}.csv",
                               path_file_csv_outputs=name_path_output)


### LUNCH SCRIPT ###

# Keeping track of processing duration
format_time = "%H:%M:%S"
current_time_start = datetime.now()
print(f"All process starts at :
{current_time_start.strftime(format_time)}\n")

# creating data frames based on the talk turns information
dfs_tt = create_tt_df(PATH_TT)

# lunch the opensmile extraction
opensmile_talk_turns(list_tt_df=dfs_tt)
```

```
# Keeping track of processing duration
current_time_end = datetime.now()
print(f"All process ends at :
{current_time_end.strftime(format_time)}\n")
total_time_process = current_time_end - current_time_start
print(f"\nAll process finished after: {total_time_process}")
```

Annex 4

```
##############################
#### DATA CLEANING SCRIPT #####
##############################

### IMPORTING IMPORTANT LIBRARIES ###

import os
import pandas as pd


### SETTING PATHS ###

IN_PATH = "/ FOLDER PATH WITH OUTPUTS /"
OUT_PATH = "/ FOLDER PATH FOR NEW OUTPUTS /"


### CREATE A DATA FRAME LIST FROM OUTPUT FILES ###

def create_csv_df(path_folder):
    # assign path
    path, dirs, files = next(os.walk(path_folder))
    file_count = len(files)
    # create empty list
    dataframes_list = []

    # append datasets from the list
    for i in range(file_count):
        if files[i] == ".DS_Store":
            pass
        else:
            temp_df = pd.read_csv(f"{path_folder}{files[i]}")
            temp_df.name = files[i]  # way to keep track of the name
            dataframes_list.append(temp_df)
    return dataframes_list


### LUNCH SCRIPT ###

# create a list of dataframes from output files
list_df = create_csv_df(IN_PATH)

# for loop for preprocessing
for df in list_df:
    name = df.name

    # drop the missing data
    df.dropna(inplace=True)

    # select only talk turns with men or women talking
    for el in ["n", "p", "c", "l", "s"]:
        df = df[df["label"] != el]

    # select only rows where F0 > 0
```

```
df = df[df["F0semitoneFrom27.5Hz_sma3nz_amean"] != 0]

# select only files if number of rows are > 40
# for men or women
if len(df[df["label"] == "m"]) < 40:
    print(f"{name} tt for men ="
          f"{len(df[df['label'] == 'm'])}")
elif len(df[df["label"] == "f"]) < 40:
    print(f"{name} tt for women ="
          f"{len(df[df['label'] == 'f'])}")

# creating a preprocessed csv file
else:
    df = df.iloc[:, 5:]
    df.to_csv(f"{OUT_PATH}{name}",
              index=False)
```

Annex 5

```
############################
##### SUBSAMPLING SCRIPT #####
############################


### IMPORTING LIBRARIES AND FUNCTIONS###

from random import randint
import pandas as pd
import os


### SETTING PATHS ###

OUTPUTS_PATH = "/ FOLDER WHERE THE OUTPUTS ARE /"


### EXTRACTING FILE NAMES FROM A FOLDER ###

def get_files_names_from_folder(folder_path):
    """collect the file names from a folder"""

    path, dirs, files = next(os.walk(folder_path))
    files = [file for file in files if file != ".DS_Store"]
    return files


### SUBSAMPLING FUNCTION ###

def subsampling_data_set(num_iterations_trough_all, columns_names_list,
                         only_pos=False, only_neg=False):
    """create a given number of set for training or testing:
    it randomly pick one observation (row) in each csv file
    and assemble them in one csv file"""

    # creating a list with all file names from the outputs folder
    outputs_names = get_files_names_from_folder(folder_path=OUTPUTS_PATH)

    # selecting only file for the selected discussion type
    if only_pos:
        outputs_names = [pos_name for pos_name in outputs_names if "pos"
in pos_name]
    if only_neg:
        outputs_names = [neg_name for neg_name in outputs_names if "neg"
in neg_name]

    # extracting the column names from one outputs file
    column_df_all =
pd.read_csv(f"{OUTPUTS_PATH}{outputs_names[0]}").columns
    column_names = ["file"]
    column_names[1:] = column_df_all

    # creating an empty data frame using the extracted column names
```

```
new_data_set = pd.DataFrame(columns=column_names)
list_of_columns = \
    list(range(0, 1)) + list(range(4, 5)) + list(range(6, 70))
new_data_set = new_data_set.iloc[:, list_of_columns]

# selecting only columns of interest
chosen_columns = \
    list(range(3, 4)) + list(range(5, 69)) # for EVRS and speech

# for loop for each file
for name in outputs_names:

    # for loop for the selected number of iterations
    for x in range(0, num_iterations_trough_all):

        # creating the new data frame for the first iteration
        if x == 0:
            new_df = pd.read_csv(f"{OUTPUTS_PATH}{name}")
            new_df = new_df.iloc[:, chosen_columns]

        # adding a randomly selected row
        # and keeping track of the already chosen rows
        i = randint(0, len(new_df.index)-1)
        new_row = new_df.loc[i]
        new_row_plus = [name]
        new_row_plus[1:] = new_row
        index = len(new_data_set.index)
        new_data_set.loc[index] = new_row_plus
        new_df.drop([i], axis=0, inplace=True)
        new_df = new_df.reset_index(drop=True)

# cleaning the data frame and renaming the column names
first_column = new_data_set.pop("file")
new_data_set.insert(0, "file", first_column)
new_data_set = new_data_set[columns_names_list]

return new_data_set
```

Annex 6

```
##################################
##### MACHINE LEARNING SCRIPT #####
##################################


### IMPORTING IMPORTANT LIBRARIES AND FUNCTIONS ###

import pandas as pd
from sklearn.model_selection import train_test_split
import statistics
from datetime import datetime


### DEFINING THE ACOUSTIC FEATURE SETS ###

ALL = \
    ['file', 'label', 'emo_h', 'emo_f',
     'F0semitoneFrom27.5Hz_sma3nz_amean',
     'F0semitoneFrom27.5Hz_sma3nz_stddevNorm',
     'F0semitoneFrom27.5Hz_sma3nz_percentile20.0',
     'F0semitoneFrom27.5Hz_sma3nz_percentile50.0',
     'F0semitoneFrom27.5Hz_sma3nz_percentile80.0',
     'F0semitoneFrom27.5Hz_sma3nz_pctlrange0-2',
     'F0semitoneFrom27.5Hz_sma3nz_meanRisingSlope',
     'F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope',
     'F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope',
     'F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope',
     'loudness_sma3_amean',
     'loudness_sma3_stddevNorm', 'loudness_sma3_percentile20.0',
     'loudness_sma3_percentile50.0', 'loudness_sma3_percentile80.0',
     'loudness_sma3_pctlrange0-2', 'loudness_sma3_meanRisingSlope',
     'loudness_sma3_stddevRisingSlope', 'loudness_sma3_meanFallingSlope',
     'loudness_sma3_stddevFallingSlope', 'jitterLocal_sma3nz_amean',
     'jitterLocal_sma3nz_stddevNorm', 'shimmerLocaldB_sma3nz_amean',
     'shimmerLocaldB_sma3nz_stddevNorm', 'HNRdBACF_sma3nz_amean',
     'HNRdBACF_sma3nz_stddevNorm', 'F1frequency_sma3nz_amean',
     'F1frequency_sma3nz_stddevNorm', 'F1bandwidth_sma3nz_amean',
     'F1bandwidth_sma3nz_stddevNorm', 'F1amplitudeLogRelF0_sma3nz_amean',
     'F1amplitudeLogRelF0_sma3nz_stddevNorm', 'F2frequency_sma3nz_amean',
     'F2frequency_sma3nz_stddevNorm', 'F2amplitudeLogRelF0_sma3nz_amean',
     'F2amplitudeLogRelF0_sma3nz_stddevNorm', 'F3frequency_sma3nz_amean',
     'F3frequency_sma3nz_stddevNorm', 'F3amplitudeLogRelF0_sma3nz_amean',
     'F3amplitudeLogRelF0_sma3nz_stddevNorm', 'alphaRatioV_sma3nz_amean',
     'alphaRatioV_sma3nz_stddevNorm', 'hammarbergIndexV_sma3nz_amean',
     'hammarbergIndexV_sma3nz_stddevNorm', 'slopeV0-500_sma3nz_amean',
     'slopeV0-500_sma3nz_stddevNorm', 'slopeV500-1500_sma3nz_amean',
     'slopeV500-1500_sma3nz_stddevNorm', 'alphaRatioUV_sma3nz_amean',
     'hammarbergIndexUV_sma3nz_amean', 'slopeUV0-500_sma3nz_amean',
     'slopeUV500-1500_sma3nz_amean', 'loudnessPeaksPerSec',
     'VoicedSegmentsPerSec', 'MeanVoicedSegmentLengthSec',
     'StddevVoicedSegmentLengthSec', 'MeanUnvoicedSegmentLength',
     'StddevUnvoicedSegmentLength']
```

```
SPECTRAL = \
    ['file', 'label', 'emo_h', 'emo_f',
     'F1frequency_sma3nz_amean', 'F1frequency_sma3nz_stddevNorm',
     'F1bandwidth_sma3nz_amean', 'F1bandwidth_sma3nz_stddevNorm',
     'F1amplitudeLogRelF0_sma3nz_amean',
     'F1amplitudeLogRelF0_sma3nz_stddevNorm',
     'F2frequency_sma3nz_amean', 'F2frequency_sma3nz_stddevNorm',
     'F2amplitudeLogRelF0_sma3nz_amean',
     'F2amplitudeLogRelF0_sma3nz_stddevNorm',
     'F3frequency_sma3nz_amean', 'F3frequency_sma3nz_stddevNorm',
     'F3amplitudeLogRelF0_sma3nz_amean',
     'F3amplitudeLogRelF0_sma3nz_stddevNorm',
     'alphaRatioV_sma3nz_amean', 'alphaRatioV_sma3nz_stddevNorm',
     'hammarbergIndexV_sma3nz_amean',
'hammarbergIndexV_sma3nz_stddevNorm',
     'slopeV0-500_sma3nz_amean', 'slopeV0-500_sma3nz_stddevNorm',
     'slopeV500-1500_sma3nz_amean','slopeV500-1500_sma3nz_stddevNorm',
     'alphaRatioUV_sma3nz_amean', 'hammarbergIndexUV_sma3nz_amean',
     'slopeUV0-500_sma3nz_amean', 'slopeUV500-1500_sma3nz_amean']

PITCH = \
    ['file', 'label', 'emo_h', 'emo_f',
     'F0semitoneFrom27.5Hz_sma3nz_amean',
     'F0semitoneFrom27.5Hz_sma3nz_stddevNorm',
     'F0semitoneFrom27.5Hz_sma3nz_percentile20.0',
     'F0semitoneFrom27.5Hz_sma3nz_percentile50.0',
     'F0semitoneFrom27.5Hz_sma3nz_percentile80.0',
     'F0semitoneFrom27.5Hz_sma3nz_pctlrange0-2',
     'F0semitoneFrom27.5Hz_sma3nz_meanRisingSlope',
     'F0semitoneFrom27.5Hz_sma3nz_stddevRisingSlope',
     'F0semitoneFrom27.5Hz_sma3nz_meanFallingSlope',
     'F0semitoneFrom27.5Hz_sma3nz_stddevFallingSlope',
     'jitterLocal_sma3nz_amean', 'jitterLocal_sma3nz_stddevNorm']

LOUDNESS = \
    ['file', 'label', 'emo_h', 'emo_f',
     'loudness_sma3_amean', 'loudness_sma3_stddevNorm',
     'loudness_sma3_percentile20.0', 'loudness_sma3_percentile50.0',
     'loudness_sma3_percentile80.0', 'loudness_sma3_pctlrange0-2',
     'loudness_sma3_meanRisingSlope', 'loudness_sma3_stddevRisingSlope',
     'loudness_sma3_meanFallingSlope',
'loudness_sma3_stddevFallingSlope',
     'shimmerLocaldB_sma3nz_amean', 'shimmerLocaldB_sma3nz_stddevNorm',
     'HNRdBACF_sma3nz_amean', 'HNRdBACF_sma3nz_stddevNorm']

DURATIONAL = \
    ['file', 'label', 'emo_h', 'emo_f',
     'loudnessPeaksPerSec', 'VoicedSegmentsPerSec',
     'MeanVoicedSegmentLengthSec', 'StddevVoicedSegmentLengthSec',
     'MeanUnvoicedSegmentLength', 'StddevUnvoicedSegmentLength']


### BINARIZE EVRS INTO 0 AND 1 ###

def binarize(values):
    """binarize the values > 0 = 1 and < 0 = 0"""
```

```
    binarized = (values > 0).astype('int')
    return binarized


### CREATE DATA SET FOR THE MIXED PREDICTIVE MODEL ###

def create_x_yf_ym_mixed(df, binarizing=True):
    """create X, y_f and y_m from a df /
    return => X, y_f, y_m"""

    columns_index = list(range(1, 2)) + \
                    list(range(4, len(df.columns)))

    X = df.iloc[:, columns_index]
    X.loc[X["label"] == "m", "label"] = 0
    X.loc[X["label"] == "f", "label"] = 1
    y_m = df.iloc[:, 2]
    y_f = df.iloc[:, 3]

    # binarize EVRS
    if binarizing:
        y_m = binarize(y_m)
        y_f = binarize(y_f)
    return X, y_f, y_m


### CREATE DATA SET FOR THE OWN PREDICTIVE MODEL ###

def create_own_pred_xm_xf_ym_yf(df, binarizing=True):
    """create X, y_f and y_m from a df /
    return => X, y_f, y_m"""

    columns_index = list(range(1, 2)) + \
                    list(range(4, len(df.columns)))

    # for m
    X_m = df.loc[df["label"] == "m"]
    y_m = X_m.iloc[:, 2]
    X_m = X_m.iloc[:, columns_index]
    X_m = X_m.iloc[:, 1:]

    # for f
    X_f = df.loc[df["label"] == "f"]
    y_f = X_f.iloc[:, 3]
    X_f = X_f.iloc[:, columns_index]
    X_f = X_f.iloc[:, 1:]

    # binarize EVRS
    if binarizing:
        y_m = binarize(y_m)
        y_f = binarize(y_f)
    return X_m, X_f, y_m, y_f


### CREATE DATA SET FOR THE CROSSED PREDICTIVE MODEL ###
```

```
def create_cross_pred_xm_xf_ymxf_yfxm(df, binarizing=True):
    """create X, y_f and y_m from a df /
    return => X, y_f, y_m"""

    columns_index = list(range(1, 2)) + \
                    list(range(4, len(df.columns)))

    # for m
    X_m = df.loc[df["label"] == "m"]
    yf_xm = X_m.iloc[:, 3]
    X_m = X_m.iloc[:, columns_index]
    X_m = X_m.iloc[:, 1:]

    # for f
    X_f = df.loc[df["label"] == "f"]
    ym_xf = X_f.iloc[:, 2]
    X_f = X_f.iloc[:, columns_index]
    X_f = X_f.iloc[:, 1:]

    # binarize EVRS
    if binarizing:
        ym_xf = binarize(ym_xf)
        yf_xm = binarize(yf_xm)
    return X_m, X_f, ym_xf, yf_xm


### MACHINE LEARNING MODELS ###

# Balanced Random Forest Classifier
def BRFC(xtrain_m, xtrain_f, ytrain_m, ytrain_f,
         xtest_m, xtest_f, ytest_m, ytest_f):
    """Balanced Random Forest Classificatier /
    return : bal_accuracy_m, cm_m, bal_accuracy_f, cm_f"""

    # importing the library
    from imblearn.ensemble import BalancedRandomForestClassifier

    # creating and training the model
    clf_m = BalancedRandomForestClassifier()
    clf_f = BalancedRandomForestClassifier()
    clf_m.fit(xtrain_m, ytrain_m)
    clf_f.fit(xtrain_f, ytrain_f)

    # Accuracy using : balanced_accuracy_score and confusion_matrix
    from sklearn.metrics import balanced_accuracy_score, confusion_matrix

    # for men
    y_pred_m = clf_m.predict(xtest_m)
    y_pred_m = [i for i in y_pred_m]
    y_test_m = ytest_m.tolist()
    bal_accuracy_m = \
        balanced_accuracy_score(y_true=y_test_m, y_pred=y_pred_m)
    tn_m, fp_m, fn_m, tp_m = \
        confusion_matrix(y_true=y_test_m, y_pred=y_pred_m).ravel()

    # for women
```

```
    y_pred_f = clf_m.predict(xtest_f)
    y_pred_f = [i for i in y_pred_f]
    y_test_f = ytest_f.tolist()
    bal_accuracy_f = \
        balanced_accuracy_score(y_true=y_test_f, y_pred=y_pred_f)
    tn_f, fp_f, fn_f, tp_f = \
        confusion_matrix(y_true=y_test_f, y_pred=y_pred_f).ravel()

    return bal_accuracy_m, tn_m, fp_m, fn_m, tp_m, \
           bal_accuracy_f, tn_f, fp_f, fn_f, tp_f


# Support Vector Machine Classifier
def SVC_KERNEL(xtrain_m, xtrain_f, ytrain_m, ytrain_f,
               xtest_m, xtest_f, ytest_m, ytest_f):
    """Support Vector Machine Classifier /
    return : bal_accuracy_m, cm_m, bal_accuracy_f, cm_f"""

    # importing the library
    from sklearn.svm import SVC

    # creating and training the model
    clf_m = SVC(kernel='rbf')
    clf_f = SVC(kernel='rbf')
    clf_m.fit(xtrain_m, ytrain_m)
    clf_f.fit(xtrain_f, ytrain_f)

    # Accuracy using : balanced_accuracy_score and confusion_matrix
    from sklearn.metrics import balanced_accuracy_score, confusion_matrix

    # for men
    y_pred_m = clf_m.predict(xtest_m)
    y_pred_m = [i for i in y_pred_m]
    y_test_m = ytest_m.tolist()
    bal_accuracy_m = \
        balanced_accuracy_score(y_true=y_test_m, y_pred=y_pred_m)
    tn_m, fp_m, fn_m, tp_m = \
        confusion_matrix(y_true=y_test_m, y_pred=y_pred_m).ravel()

    # for women
    y_pred_f = clf_m.predict(xtest_f)
    y_pred_f = [i for i in y_pred_f]
    y_test_f = ytest_f.tolist()
    bal_accuracy_f = \
        balanced_accuracy_score(y_true=y_test_f, y_pred=y_pred_f)
    tn_f, fp_f, fn_f, tp_f = \
        confusion_matrix(y_true=y_test_f, y_pred=y_pred_f).ravel()

    return bal_accuracy_m, tn_m, fp_m, fn_m, tp_m, \
           bal_accuracy_f, tn_f, fp_f, fn_f, tp_f


# K-Nearest Neighbors Classifier
def KNC(xtrain_m, xtrain_f, ytrain_m, ytrain_f,
        xtest_m, xtest_f, ytest_m, ytest_f):
    """K-Nearest Neighbors Classifier /
```

```
    return : bal_accuracy_m, cm_m, bal_accuracy_f, cm_f"""

    # importing the library
    from sklearn.neighbors import KNeighborsClassifier

    # creating and training the model
    clf_m = KNeighborsClassifier()
    clf_f = KNeighborsClassifier()
    clf_m.fit(xtrain_m, ytrain_m)
    clf_f.fit(xtrain_f, ytrain_f)

    # Accuracy using : balanced_accuracy_score and confusion_matrix
    from sklearn.metrics import balanced_accuracy_score, confusion_matrix

    # for m
    y_pred_m = clf_m.predict(xtest_m)
    y_pred_m = [i for i in y_pred_m]
    y_test_m = ytest_m.tolist()
    bal_accuracy_m = \
        balanced_accuracy_score(y_true=y_test_m, y_pred=y_pred_m)
    tn_m, fp_m, fn_m, tp_m = \
        confusion_matrix(y_true=y_test_m, y_pred=y_pred_m).ravel()

    # for f
    y_pred_f = clf_m.predict(xtest_f)
    y_pred_f = [i for i in y_pred_f]
    y_test_f = ytest_f.tolist()
    bal_accuracy_f = \
        balanced_accuracy_score(y_true=y_test_f, y_pred=y_pred_f)
    tn_f, fp_f, fn_f, tp_f = \
        confusion_matrix(y_true=y_test_f, y_pred=y_pred_f).ravel()

    return bal_accuracy_m, tn_m, fp_m, fn_m, tp_m, \
           bal_accuracy_f, tn_f, fp_f, fn_f, tp_f


### SUBSAMPLING ###
def subsampling_data_set():
    # see Annex 5


### FUNCTION FRAMING THE USE OF MACHINE LEARNING MODELS

def draw():

    # select the feature set
    for col in [ALL, PITCH, SPECTRAL, DURATIONAL, LOUDNESS]:

        # print the selected feature set
        if col == PITCH:
            print("Features = PITCH")
        elif col == DURATIONAL:
            print("Features = DURATIONAL")
        elif col == LOUDNESS:
            print("Features = LOUDNESS")
        elif col == SPECTRAL:
```

```
    print("Features = SPECTRAL")
elif col == ALL:
    print("Features = ALL")
else:
    print("OOPS!! AN ERROR OCCURED... hmmmm....")
print("\n")

# selecting a predictive model
for prediction in ["MIXED", "OWN", "CROSSED"]:
    PREDICTIONS = prediction

    # print the selected predictive model
    print(f"{PREDICTIONS} predictions\n")

    # reset lists for balanced accuracy
    # and confusion matrix
    bal_accuracy_mean_m = []
    tn_m_total = 0
    fp_m_total = 0
    fn_m_total = 0
    tp_m_total = 0

    bal_accuracy_mean_f = []
    tn_f_total = 0
    fp_f_total = 0
    fn_f_total = 0
    tp_f_total = 0

    # keeping track of the time processing
    new_time_start = datetime.now()

    # for loop for draws
    for i in range(NUM_DRAWS):

# subsampling : taking n talk turns randomly from every csv files
        subsample = \
   subsampling_data_set(num_iterations_trough_all=NUM_ITERATIONS,
                                columns_names_list=col,
                                only_neg=ONLY_NEG,
                                only_pos=ONLY_POS)

        if PREDICTIONS == "MIXED":
            # creating X and y
            # mixed prediction example for men
            # --> xm + ym --> ym or yf
            # and transforming m => 0 and f => 1
            X, y_f, y_m = create_x_yf_ym_mixed(df=subsample,
                                               binarizing=True)
            # splitting the data_set -> mixed
            X_f_train, X_f_test, y_f_train, y_f_test = \
                train_test_split(X, y_f,test_size=RATIO_TEST_SET)
            X_m_train, X_m_test, y_m_train, y_m_test = \
                train_test_split(X, y_m,test_size=RATIO_TEST_SET)

        elif PREDICTIONS == "OWN":
            # creating X and y
```

```
        # own prediction example for men : xm --> ym
        X_m, X_f, y_m, y_f = \
            create_own_pred_xm_xf_ym_yf(df=subsample,
                                        binarizing=True)
        # splitting the data_set -> own
        X_f_train, X_f_test, y_f_train, y_f_test = \
           train_test_split(X_f,y_f,test_size=RATIO_TEST_SET)
        X_m_train, X_m_test, y_m_train, y_m_test = \
           train_test_split(X_m,y_m,test_size=RATIO_TEST_SET)

    elif PREDICTIONS == "CROSSED":
        # creating X and y
        # crossed prediction example : xm --> yf
        X_m, X_f, ym_xf, yf_xm = \
            create_cross_pred_xm_xf_ymxf_yfxm(df=subsample,
                                              binarizing=True)
        # splitting the data_set
        # -> y_m = ym_xf and y_f = yf_xm
        X_f_train, X_f_test, y_f_train, y_f_test = \
         train_test_split(X_f,ym_xf,test_size=RATIO_TEST_SET)
        X_m_train, X_m_test, y_m_train, y_m_test = \
         train_test_split(X_m,yf_xm,test_size=RATIO_TEST_SET)

    # creating and training the model
    bal_accuracy_m, tn_m, fp_m, fn_m, tp_m, \
    bal_accuracy_f, tn_f, fp_f, fn_f, tp_f = \
        MACHINE LEARNING MODEL(
            xtrain_m=X_m_train,
            xtrain_f=X_f_train,
            ytrain_m=y_m_train,
            ytrain_f=y_f_train,
            xtest_m=X_m_test,
            xtest_f=X_f_test,
            ytest_m=y_m_test,
            ytest_f=y_f_test
    )

    # adding the balanced accuracy to the list
    # and actualizing the confusion matrix
    bal_accuracy_mean_m.append(bal_accuracy_m)
    tn_m_total += tn_m
    fp_m_total += fp_m
    fn_m_total += fn_m
    tp_m_total += tp_m
    bal_accuracy_mean_f.append(bal_accuracy_f)
    tn_f_total += tn_f
    fp_f_total += fp_f
    fn_f_total += fn_f
    tp_f_total += tp_f

    # printing a report for each draw (optional)
    print(f"Accuracies for the draw n° {i + 1}/{NUM_DRAWS}")
    print(f"for m : bal_accuracy = {bal_accuracy_m} \n"
          f"and confusion_matrix (tn, fp, fn, tp) = "
          f"{tn_m, fp_m, fn_m, tp_m}")
    print(f"for f : bal_accuracy = {bal_accuracy_f} \n"
```

```
                    f"and confusion_matrix (tn, fp, fn, tp) = "
                    f"{tn_f, fp_f, fn_f, tp_f}\n")

            # keeping track on all final balanced accuracies scores
            print("balanced accuracies for m")
            print(bal_accuracy_mean_m)
            print("balanced accuracies for f")
            print(bal_accuracy_mean_f)

            # final results for the selected model after 100 draws
            # => mean, standard deviations and confusion matrix
            print("\nFINAL RESULTS")
            print("for m")
            print(f"mean of bal_accuracies = "
                  f"{statistics.mean(bal_accuracy_mean_m)}")
            print(f"standard deviation of bal_accuracies = "
                  f"{statistics.stdev(bal_accuracy_mean_m)}")
            print(f"total confusion_matrix (tn, fp, fn, tp) = "
                  f"{tn_m_total, fp_m_total, fn_m_total, tp_m_total}")
            print("for f")
            print(f"mean of bal_accuracies = "
                  f"{statistics.mean(bal_accuracy_mean_f)}")
            print(f"standard deviation of bal_accuracies = "
                  f"{statistics.stdev(bal_accuracy_mean_f)}")
            print(f"total confusion_matrix (tn, fp, fn, tp) = "
                  f"{tn_f_total, fp_f_total, fn_f_total, tp_f_total}")
            print("\n")


### SETTING UP PARAMETERS ###

# number of draws
NUM_DRAWS = 100

# list of number of iterations randomly chosen from each file
LIST_NUM_ITERATIONS = [10, 20, 30]

# test / training sets ratio
RATIO_TEST_SET = 0.25


### LUNCH SCRIPT ###

#To keep track of the time
format_time = "%H:%M:%S"
current_time_start = datetime.now()
print(f"All process starts at :
{current_time_start.strftime(format_time)}\n")
######

print(f"NAME OF THE MACHINE LEARNING MODEL\n")
print(f"n° of draws = {NUM_DRAWS}")
print(f"Test set ratio = {RATIO_TEST_SET}")

# for loop for each number of iterations
for a in LIST_NUM_ITERATIONS:
```

```
    print("++++++++++++++++++++++++++++++\n")
    NUM_ITERATIONS = a

    # printing the number of iterations selected
    print(f"n° of talk turns taken from each file = "
          f"{NUM_ITERATIONS}\n")

    # for loop for each type of discussion
    # (positive, negative or all)
    for discussion in [1, 2, 3]:
       if discussion == 1:
           ONLY_NEG = False
           ONLY_POS = True
           print("++++++++++++++++++++++++++++++\n")
           print("Only with positive discussions")
       elif discussion == 2:
           ONLY_NEG = True
           ONLY_POS = False
           print("++++++++++++++++++++++++++++++\n")
           print("Only with negative discussions")
       elif discussion == 3:
           ONLY_NEG = False
           ONLY_POS = False
           print("++++++++++++++++++++++++++++++\n")
           print("All discussions")

       # lunch the 100 draws for the selected model
       draw()


# to keep track on time
current_time_end = datetime.now()
print(f"All process ends at :
{current_time_end.strftime(format_time)}\n")
total_time_process = current_time_end - current_time_start
print(f"\nAll process finished after: {total_time_process}")
```