



TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA

Sistema de detección inteligente de emociones mediante dispositivos wearables

Aplicación de técnicas de machine learning y utilización de
dispositivos wereables para la detección de emociones

Autor

Yacine Brek Prieto

Directores

Francisco Manuel García Moreno

María Bermúdez Edo



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, septiembre de 2021

Sistema de detección inteligente de emociones mediante dispositivos wearables: Aplicación de técnicas de machine learning y utilización de dispositivos *wearables* para la detección de emociones

Yacine Brek Prieto

Palabras clave: Detección de emociones, machine learning, dispositivos *wearables*, señales fisiológicas, vídeos

Resumen

En este proyecto se ha abarcado la detección de emociones mediante la utilización de dispositivos *wearables*. Concretamente, se busca predecir el estado neutro y las seis emociones básicas: tristeza, alegría, sorpresa, miedo, asco e ira. Para dicho propósito se han utilizado técnicas de *machine learning*. Asimismo, para evocar las emociones de los participantes se ha hecho uso de vídeos referentes a cada una de las distintas emociones y al estado neutro.

Los *wearables* utilizados en esta prueba son una diadema y una pulsera. Para obtener la muestra de datos se ha contado con la participación de voluntarios que vieron los vídeos mientras usaban los dispositivos.

Por tanto, se puede decir que el experimento se constituye de las siguientes partes:

- Selección de participantes.
- Recolección de datos.
- Tratamiento de datos.
- Realización de los modelos.

El principal motivo por el que se ha realizado este trabajo es porque se quiere contribuir al ámbito de la medicina, pudiendo ayudar así a personas que sufren trastornos mentales, parálisis cerebral o autismo, entre otras enfermedades.

Smart emotion detection system using wearable devices: Application of machine learning techniques and use of wearable devices for emotions detection.

Yacine Brek Prieto

Keywords: Emotion detection, machine learning, wearable device, physiological signals, videos

Abstract

This project has covered the detection of emotions by using of wearable devices. Specifically, it seeks to predict the neutral state and the six basic emotions: sadness, joy, surprise, fear, disgust and anger. Machine learning techniques have been used for this purpose. Likewise, to evoke the participants emotions, videos referring each emotion and the neutral state have been used.

A headband and a bracelet have been used for this project. To obtain the data sample, volunteers participation has been needed. Volunteers have worn these devices while they were watching the videos.

Therefore, it can be said that the experiment consists of the following parts:

- Participants selection.
- Data collection.
- Data processing.
- Models elaboration.

The main reason why this work has been carried out is because of my interest in helping medicine by using computer science. Like this, people who suffer from mental disorders, cerebral palsy or autism, among other diseases, can be helped.

Yo, **Yacine Brek Prieto**, alumno de la titulación TITULACIÓN de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77037521X, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Yacine Brek Prieto

Granada a 5 de septiembre de 2021.

D. **Francisco Manuel García Moreno**, Profesor del Departamento de lenguajes y Sistemas Informáticos de la Universidad de Granada.

D.^a **María del Campo Bermúdez Edo**, Profesora del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Sistema de detección inteligente de emociones mediante dispositivos wearables, Aplicación de técnicas de machine learning y utilización de dispositivos wearables para la detección de emociones*, ha sido realizado bajo su supervisión por **Yacine Brek Prieto**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 5 de septiembre de 2021.

Los directores:

Francisco Manuel García Moreno María del Campo Bermúdez Edo

Agradecimientos

Me gustaría agradecer a mis familiares y amigos por el apoyo incondicional durante la realización de este proyecto. En especial, a mis padres, mi abuela, mi hermano y a Paula. Además, quiero agradecer y hacer otra mención especial a mis tutores, Francisco Manuel García Moreno y María del Campo Bermúdez Edo, por ofrecerme esta gran oportunidad y ayudarme siempre que lo he necesitado. También, le doy las gracias a los participantes en el experimento por haber hecho esto posible y haber colaborado.

Índice general

1. Introducción	21
1.1. Descripción del problema	22
1.2. Descripción de los dispositivos utilizados	23
1.3. Motivación	25
1.4. Objetivos	26
1.5. Cronograma	27
1.6. Presupuesto	27
2. Estado del arte	29
3. Fundamentos teóricos	35
3.1. Inteligencia Artificial	35
3.2. Machine learning	36
4. Algoritmos	39
4.1. Support Vector Machine	39
4.1.1. Clasificación lineal de SVM	39
4.1.2. Clasificación no lineal de SVM	42
4.1.3. Multiclasificación	45
4.2. Random Forest	47
4.2.1. Árboles de decisión	47
4.2.2. Random Forest	50
4.3. Regresión logística	51
4.3.1. Multiclasificación	52
5. Experimentos y discusión	53
5.1. Fases del experimento	53
5.1.1. Selección de los vídeos	53
5.1.2. Recolección de los datos y protocolo	54
5.1.3. Tratamiento de los datos	55
5.1.4. Modelos	65
5.1.5. Validación y métricas	66
5.2. Resultados	68
5.3. Lenguaje de programación librerías	72

Índice de figuras

2.1. Proceso de reconocimiento de emociones usando señales biológicas de <i>wearables</i>	31
3.1. Definiciones de inteligencia artificial organizadas en 4 categorías [1].	36
4.1. Conjunto de datos iris con varios clasificadores [2]	40
4.2. Conjunto de datos iris con outliers [2]	41
4.3. [2]	42
4.4. <i>SVM</i> con frontera no lineal [3]	44
4.5. Cambio de dimensión [3]	44
4.6. <i>SVM</i> con truco del Kernel [3]	45
4.7. Conjunto de datos de 3 clases distintas: verde, rojo y azul [4]	46
4.8. Ejemplo de multclasificación con <i>One-to-One</i> [4]	46
4.9. Ejemplo de multclasificación con <i>One-to-Rest</i> [4]	47
4.10. Árbol de decisión de Iris [2]	49
4.11. Límites del árbol de decisión de Iris [2]	49
4.12. Función logística [2]	51
5.1. Histogramas correspondientes a la distribución de las características de los datos referentes a los fragmentos intensos	61
5.2. Histogramas correspondientes a la distribución de las características de los datos referentes a los vídeos completos	62
5.3. Histograma de clases correspondiente a los datos de los fragmentos intensos	63
5.4. Histograma de clases correspondiente a los datos de los vídeos completos	63
5.5. Matriz de correlación correspondiente a los datos de los fragmentos intensos	64
5.6. Matriz de correlación correspondiente a los datos de los vídeos completos	65
5.7. <i>cross validation</i> [5]	67
5.8. Matriz de confusión del modelo que usa <i>Random Forest</i> para los datos de los fragmentos intensos	71

5.9. Matriz de confusión del modelo que usa <i>Random Forest</i> para los datos de los vídeos completos	72
--	----

Índice de cuadros

1.1. Cronograma del proyecto.	27
1.2. Presupuesto del proyecto.	28

Capítulo 1

Introducción

Las emociones juegan un papel fundamental en la vida de los seres humanos y, a pesar de que todos las experimentamos constantemente, no dejan de ser complejas desde el punto de vista científico. Éstas impulsan la mayoría de nuestras decisiones [6], no solo las intuitivas [7], por lo que tienen un papel fundamental en nuestra vida cotidiana. Por ende, la detección de emociones es un tema complejo de tratar pero a su vez necesario, debido a la gran cantidad de aplicaciones que puede tener en el ámbito de la medicina y la psicología, entre otros.

En este trabajo de fin de grado, para la detección de emociones se utilizan señales fisiológicas, ya que se trabaja con dispositivos *wearables*¹ que tienen sensores para captarlas. La repercusión y la mejoría que han adquirido este tipo de dispositivos a lo largo de los últimos años se ha visto reflejada también en el ámbito de la detección de emociones, ya que estos se han empezado a utilizar frecuentemente para este tipo de experimentos, demostrando así que son una buena opción para la obtención de señales fisiológicas. En este trabajo, se han utilizado dos dispositivos que han sido fundamentales para la recolección de señales fisiológicas de cada usuario. Se trata de una pulsera y de una diadema inteligente.

Como era de esperar, el área del *machine learning* también ha influido en este campo debido a la importancia que está adquiriendo para la resolución de problemas multidisciplinares. Concretamente, son puntos clave el resurgimiento de las redes neuronales y los buenos resultados que ofrece el uso de algoritmos clásicos de *machine learning* en ciertos problemas. Por tanto, en este trabajo se han usado diferentes modelos para abordar el problema a tratar y finalmente se ha optado por el que mejores resultados aporta.

¹Los *wearables* son dispositivos electrónicos inteligentes incorporados a la vestimenta o usados corporalmente como implantes o accesorios que pueden actuar como extensión del cuerpo o mente del usuario

Para provocar distintas emociones en los usuarios hay diversas opciones. Las más frecuentes son las siguientes [8]:

- Visualización de vídeos.
- Visualización de imágenes.
- Reproducción de diferentes canciones o géneros musicales.
- Reproducción de sonidos.

La opción elegida para este trabajo ha sido la visualización de vídeos, concretamente escenas de películas, por ser la opción más frecuente en este tipo de experimentos [8]. Además, este método es más eficaz que otros verificados por estudios anteriores [9]. Esta opción permite que el usuario pueda sentir emociones sin experimentar consecuencias físicas, como por ejemplo ocurre con el miedo: no es necesario exponer al sujeto a una situación arriesgada. Además, mediante los vídeos el usuario utiliza varios sentidos, concretamente la vista y el oído. Así se consigue una experiencia más real y, consecuentemente, unas emociones más intensas. Otra principal ventaja que tiene dicha elección es que se trata de un entorno cerrado (también llamados experimentos de laboratorio), que son los más usados en esta corriente de trabajos, a diferencia del entorno abierto (también llamados experimentos de campo, que se realizan en la vida real) [8]. De esta manera es más factible controlar elementos externos que pueden perjudicar el experimento. Además, el uso de vídeos permite que siempre se tenga la misma escena para provocar una emoción a todos los usuarios, siendo así las mismas condiciones experimentales para todos.

1.1. Descripción del problema

El problema planteado en este trabajo consiste en detectar el estado neutro y las seis emociones básicas² mediante el uso de *wearables*, para captar señales fisiológicas mientras se evocan las emociones a través de fragmentos de películas.

En el ámbito del *machine learning*, dicha problemática pertenece al aprendizaje supervisado, y más concretamente a clasificación. A continuación, se explica esto más detalladamente:

²Las seis emociones básicas son tristeza, alegría, miedo, asco, sorpresa e ira

- Se cataloga como aprendizaje supervisado porque se utilizará una muestra etiquetada, donde cada etiqueta puede ser una de las seis emociones básicas o el estado neutro.
- Se asigna a clasificación porque las clases que se quieren predecir son las emociones y el estado neutro, lo cual son valores discretos y no continuos. Particularmente, se trata de multclasificación porque hay más de dos clases a predecir.

Si se consiguiera resolver este problema sería un gran avance para la ciencia, pero en especial para dos campos: interacción máquina humano y medicina [8]. Para el primer campo la detección de emociones podría tener gran utilidad, ya que esto le facilitaría mucho la comprensión a la máquina [10]. Esto sería un gran paso para alcanzar una interacción más real y natural entre ambos. Asimismo, en medicina podría contribuir bastante en el ámbito de la salud mental [11], [9], [12], enfermedades como el autismo [13], [14] o trastornos mentales [15], entre otras [16]. Por lo tanto, se puede apreciar que no son pocas las aplicaciones y utilidades que este campo posee. Asimismo, todas son muy relevantes porque pueden facilitar mucho la vida al humano. Es por esto, por lo que hay tanto interés en resolver este problema.

1.2. Descripción de los dispositivos utilizados

Como se ha mencionado anteriormente, para el desarrollo de este proyecto se han utilizado dos *wearables*.

Uno de los dispositivos utilizados es la pulsera Empatica E4 [17]. Se trata de un *wearable* que ofrece datos fisiológicos en tiempo real, permitiendo a los investigadores poder realizar análisis y visualizaciones con detalle a través de dichos datos. En especial, es uno de los dispositivos más utilizados en trabajos relacionados con el reconocimiento de emociones y el uso de *wearables* [8].

Los datos que se obtienen con la pulsera son:

- Actividad electrodérmica (EDA), también conocida como respuesta galvánica de la piel (GSR).
- Frecuencia cardíaca (HR).
- Pulso de volumen de sangre (BVP).
- Temperatura de la piel.
- Aceleración.

Por tanto, la pulsera recoge el EDA, HR y BVP que según la literatura científica [8] son parámetros que se han utilizado en diversos trabajos de la misma vertiente debido a su importancia.

La pulsera cuenta con los siguientes sensores:

- Sensor PPG: Mide el pulso sanguíneo a partir del cual se puede derivar la variabilidad de la frecuencia cardíaca.
- Sensor EDA (Sensor GSR): Mide los cambios que fluctúan constantemente en ciertas propiedades eléctricas de la piel.
- Termopila infrarroja: Mide la temperatura de la piel.
- Acelerómetro de tres ejes: Capta la actividad basada en el movimiento.

Con este dispositivo se han realizado diversos experimentos de investigación en los que, a partir de las señales fisiológicas recogidas, se han llevado a cabo estudios y análisis de los datos para obtener conclusiones. Uno de los experimentos consistió en evocar los sentimientos de visitantes alemanes y holandeses de un museo sobre la segunda guerra mundial. Para ello se utilizó el *wearable E4 wristband* y se observaron detalles como que en ocasiones los sentimientos de los holandeses y alemanes eran distintos, destacando la culpabilidad en estos últimos. Finalmente, como conclusión se obtuvo que los sentimientos acerca de este hecho histórico siguen estando muy presentes y afectan al público de una manera muy compleja, más allá de simples categorías históricas.

La pulsera permite tomar los datos en el laboratorio y analizarlos en casa. Además, con el modo de grabación se puede grabar hasta 36 horas con una resolución de sincronización de 5 segundos gracias a la memoria interna de la que dispone E4. Esto es ideal para estudios de larga duración. Además, se puede importar los datos vía USB a una plataforma segura en una nube que proporcionan los fabricantes del dispositivo. En la nube se pueden visualizar y administrar los datos, los cuales están protegidos con técnicas criptográficas. También, se puede descargar los datos en formato CSV para facilitar el procesamiento y el análisis en aplicaciones de terceros.

Asimismo, hay un modo de transmisión por bluetooth que permite ver los datos recogidos por los sensores del dispositivo en tiempo real. Una vez que finalice la sesión, los datos se cargan automáticamente en la plataforma de la nube mencionada anteriormente.

El otro dispositivo utilizado es la diadema Muse 2 [18]. Se trata de un *wearable* que normalmente se utiliza para la meditación en personas de todos los niveles. Cuenta con retroalimentación de los datos en tiempo real y mide

señales fisiológicas del cuerpo, cerebro y corazón. Cabe destacar que brinda a los usuarios retroalimentación de los datos respectivos al estado mental en tiempo real.

La diadema proporciona los siguientes datos:

- Retroalimentación de ondas cerebrales (EGG = electroencefalograma) en tiempo real.
- Variabilidad de la frecuencia cardíaca o HRV (PPG + Oximetría de Pulso).
- Acelerómetro.
- Respiración (PPG + Giroscopio).

El electroencefalograma es bastante utilizado para la detección de emociones [8]. Este se obtiene con la diadema gracias a cuatro sensores que posee. Hay dos que están más próximos a las orejas y otros dos al centro de la frente. Cada uno de estos proporciona cinco ondas cerebrales con distinta frecuencia.

1.3. Motivación

En este proyecto se busca aplicar conocimientos de ingeniería informática, con la finalidad de colaborar en el ámbito de la psicología y la medicina, pudiendo así ayudar a pacientes que sufren parálisis cerebral, autismo, trastornos emocionales, entre otras enfermedades. Habría más aplicaciones existentes como puede ser mejorar la interacción entre máquinas y humanos pero mi principal deseo es el mencionado anteriormente, ya que no hay mayor satisfacción que poder ayudar a quien más lo necesita.

Por ende, en este trabajo se pretende desarrollar un buen sistema de detección de emociones que sirva de ayuda a personas con dificultades para expresarlas. Aunque en otros trabajos como [13] [14] ya se ha tratado la detección de emociones desde el contexto terapéutico (donde se pretende ayudar y enseñar a los niños con autismo a comprender mejor sus emociones), según la revisión literaria [8] todavía no encontramos trabajos completos sobre este tema. Es decir, la mayoría de los proyectos realizados hasta el momento sobre esta temática tienen diversas carencias como pueden ser:

- Muestra poco representativa respecto a la población (personas de distinto sexo, de diferentes edades, entre otras cosas) [9].
- Se prueban uno o dos modelos de *machine learning* [19] [20].

- No optimizan lo suficiente los parámetros del modelo de *machine learning* [8].
- No tratan el desbalanceo en caso de que lo haya [8].

Dichas carencias se intentaran paliar en este proyecto para conseguir un trabajo más completo sobre esta temática. Además, hay que tener en cuenta que a pesar de que la detección de emociones es un tema de gran interés y relevancia, todavía no abundan las investigaciones respectivas a este tema.

1.4. Objetivos

El principal objetivo de este trabajo es conseguir, a través de un modelo de *machine learning*, una buena precisión al predecir las seis emociones básicas, además del estado neutral. Es decir, en total habría que predecir siete clases: tristeza, alegría, sorpresa, miedo, asco, ira y el estado neutro.

Para cumplir el objetivo principal, es necesario cumplir los siguientes subobjetivos:

- Selección de los vídeos: es necesario encontrar vídeos que representen correctamente las seis emociones y el estado neutro.
- Recolección del personal y de datos: es importante obtener una muestra representativa de la población.
- Tratamiento de los datos: es imprescindible preparar bien los datos para poder entrenar bien los modelos.
- Realizar predicciones: es fundamental optimizar bien los hiperparámetros y escoger algoritmos de aprendizaje adecuados al problema para obtener un modelo que realice buenas predicciones.

Asimismo, se busca obtener conclusiones que puedan ayudar a futuros trabajos y contribuir al conocimiento de este ámbito. Esto será un factor determinante a la hora de realizar otros experimentos, ya que se puede seguir este enfoque e intentar mejorarlo o cambiarlo completamente.

Por otro lado, hay que ser conscientes de la dificultad que conlleva realizar un experimento de esta temática, ya que las pruebas no se le pueden repetir a un sujeto bajo ningún concepto porque no servirían, los dispositivos deben funcionar correctamente, hay que contar con la voluntad de los sujetos en una situación de pandemia mundial para que participen en el experimento, cada participante debe tener las mismas condiciones, hay que realizar la

muestra de datos manualmente, realizar un preprocesado de datos obtenidos de distintos dispositivos, hacer diversos modelos, entre otras cosas.

1.5. Cronograma

El tiempo empleado en la realización del trabajo es de seis meses. Concretamente, se comenzó en el mes de marzo y se terminó a finales de agosto / principios de septiembre. No se ha seguido una planificación totalmente estricta. De hecho, se ha llevado a cabo un desarrollo ágil, permitiendo así re-planificar según los avances realizados frente a los esperados. A continuación, se muestra en la siguiente tabla el desarrollo del trabajo más detalladamente.

Marzo	Abril
Investigación sobre los dispositivos.	Investigación sobre la literatura científica.
Recolección de información relevante.	Recolección de información relevante.
Mayo	Junio
Investigación sobre la literatura científica.	Definición del protocolo.
Recolección de información relevante.	Recolección de datos.
	Realización de la documentación.
Julio	Agosto
Tratamiento de los datos.	Realización de los modelos.
Realización de la documentación.	Realización de la documentación.
	Revisiones del trabajo.

Cuadro 1.1: Cronograma del proyecto.

1.6. Presupuesto

Para obtener el presupuesto de este proyecto hace falta cubrir los gastos relativos en cuanto a dispositivos y personal. Para calcular el sueldo medio de un ingeniero nos hemos basado en [21]. En la siguiente tabla se puede apreciar el presupuesto total y su respectivos desglose.

Presupuesto Hardware	Diadema Muse 2: 271'98 euros. Pulsera E4: 1690'14 euros. Móvil Xiaomi MI A3: 249 euros. Portatil HP Pavilion 15-CX0005NS: 882'60 euros.
Presupuesto de personal	Salario medio de un ingeniero informático 2164 euros / mes durante seis meses: 12984 euros.
Total	Total del presupuesto: 16077'72 euros.

Cuadro 1.2: Presupuesto del proyecto.

Capítulo 2

Estado del arte

Basándonos en la revisión literaria [8], se puede afirmar que la mayor parte de la investigación acerca del reconocimiento de emociones que hay disponible hasta el momento se ha centrado en las reacciones de los participantes provocadas por estímulos preparados en un entorno controlado.

Según la literatura científica, una emoción se puede definir como un sentimiento breve, intenso y dirigido [22]. Además, las emociones pueden ser identificadas de las siguientes maneras [8]:

- Percepción subjetiva del participante, es decir, autoevaluación del sujeto.
- Reacción del organismo del participante, concretamente mediante señales fisiológicas. [9]
- Señales del comportamiento como la expresión facial, voz, movimientos corporales, etc. [11]
- Evaluación externa realizada por los compañeros del sujeto. Por ejemplo, un adulto reconociendo el estado de un niño. [13]

Como se ha mencionado anteriormente, nuestro interés se centra en el segundo punto, ya que este trabajo se basa en dicho enfoque. Cabe destacar que este punto de vista tiene una clara ventaja sobre los demás, y es que requiere de dispositivos que miden señales fisiológicas de manera objetiva. Un detalle a tener en cuenta es que según la revisión literaria [8] se prima el uso de dispositivos no invasivos¹.

Los estudios sobre el reconocimiento de emociones han cambiado su enfoque a lo largo del tiempo. Concretamente, han pasado de centrarse en

¹Los dispositivos no invasivos son aquellos que no involucran instrumentos que rompen la piel o que penetran físicamente en el cuerpo.

soluciones EEG [23], [24], a análisis facial y del habla [25], [26], y finalmente a señales fisiológicas [22], [27]. Además, un dato relevante es que se ha concluido que los experimentos sobre reconocimiento de emociones son notablemente más complejos cuando se realizan fuera de un entorno cerrado y controlado [22].

Por lo general, en este tipo de investigaciones de reconocimiento sobre emociones podemos distinguir seis etapas [8]:

1. Modelo emocional.
2. Reclutamiento y análisis de los participantes.
3. Preparación y anotación de estímulos.
4. Recopilación de señales con *wearables*, procesamiento y autoevaluación.
5. Razonamiento y validación.
6. Pruebas en la vida real.

La primera etapa es fundamental, ya que la decisión sobre el modelo emocional influye directamente en el resultado. A continuación, está la etapa de reclutamiento y análisis de los participantes donde se estudian qué perfiles son válidos y se reclutan sujetos para el experimento. Hay usuarios que deben ser excluidos, ya que si padecen enfermedades o problemas cardíacos pueden interferir significativamente con las señales fisiológicas. La tercera etapa determina la configuración del experimento, puesto que esto variará en función de diversos factores como puede ser que el experimento sea de campo o de laboratorio, entre otros. También se valora cómo se van a preparar los estímulos y a realizar las respectivas anotaciones. Tras todo esto, está la cuarta etapa donde las principales tareas son: recopilar las señales fisiológicas de los usuarios con los dispositivos, autoevaluar la intensidad de las emociones por cada usuario, preprocesar los datos y etiquetarlos. La penúltima etapa, se corresponde con el entrenamiento y la validación del modelo. Por último, en la etapa seis, se utiliza el modelo en la vida real.

Más concretamente, en la figura 2.1 se puede apreciar el proceso para el reconocimiento de emociones usando señales fisiológicas obtenidas a partir de dispositivos *wearables*.

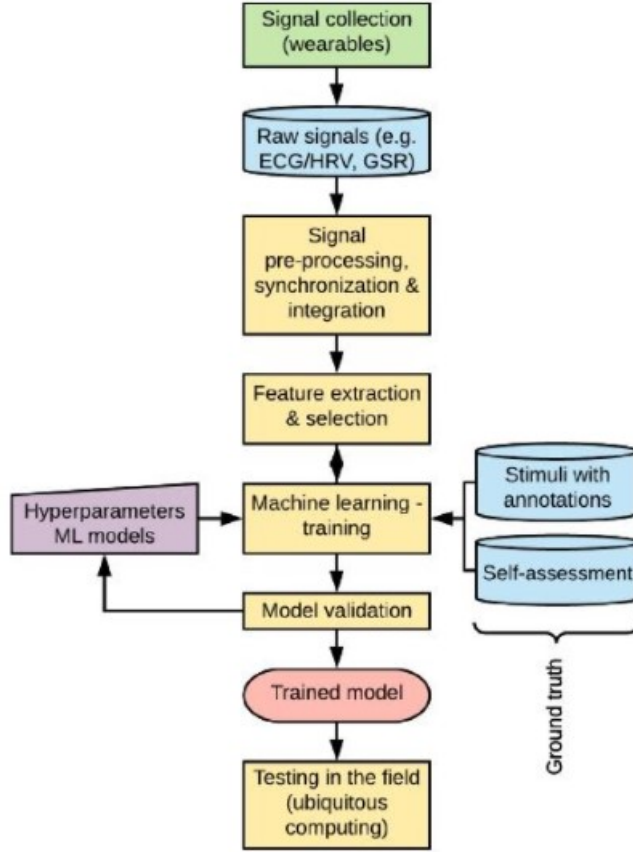


Figura 2.1: Proceso de reconocimiento de emociones usando señales biológicas de *wearables*

[8]

Debido a la complejidad de las emociones, éstas se pueden modelar de diversas maneras conceptualmente. Por lo que no existe un modelo único y común en la comunidad científica [8]. Un enfoque muy conocido es el de las seis emociones básicas² propuesto por Ekman y Friesen [28] utilizados en [29]. Otro punto de vista es el de la rueda de las emociones de Plutchik³ [30] con ocho emociones principales utilizadas como $8 \times 3 = 24$ *emojis* [31] o limitado a solo 6 emociones básicas [32]. Por otro lado, muchos autores han decidido aplicar sus propias categorías de emociones, las cuales suelen ser un

²Las seis emociones básicas son: tristeza, alegría, sorpresa, miedo, asco e ira.

³La rueda de las emociones de Plutchik cuenta con ocho emociones: alegría, confianza, miedo, sorpresa, tristeza, aversión, ira y anticipación. Además, cada emoción cuenta con tres ejes principales: tipología, antagonismo e intensidad. De ahí las $8 \times 3 = 24$ *emojis*

subconjunto o una modificación de las clásicas. Algunos ejemplos de estas categorías son: feliz-triste-ira + estado neutral [33] extendido en [34] con miedo; feliz-triste-ira-dolor [14]; alegría tristeza-ira-placer [15], entre otros.

Los participantes son un elemento imprescindible en este tipo de experimentos. Por tanto, hay que tener en cuenta que varios estudios han revelado que la edad [35], el género [36] y la personalidad [37] influyen en las emociones. Además, en muchos trabajos la salud de los participantes se tiene en cuenta. Concretamente, se consideran el historial médico y de medicamentos [16], [11], [31], [38], [9], [39], [40], información sobre problemas de la vista [38], [41], y si el sujeto está en periodo de embarazo [16]. Algunos trabajos tienen un cuestionario sobre depresión [15], [38], exceso de azúcar [31], o consumo de tabaco, alcohol o café [16], [39], [41], [40]. Otra práctica que se suele pedir a los sujetos de los experimentos es que valoren la intensidad de las emociones experimentadas [11], [9], [39], [42], [43] o que seleccionen sus emociones de una lista cerrada [29], [44], [45]. Las preguntas abiertas también son utilizadas [33]. Incluso se usa una combinación de estos dos modelos [46], [47]. Por otro lado, hay autores que utilizan cuestionarios previamente establecidos o versiones modificadas [16], [29], [14], [41], [40], [12].

Otro elemento fundamental en esta vertiente de experimentos es el procedimiento llevado a cabo para provocar emociones. Lo más utilizado suele ser:

- Vídeos [16], [31], [14], [15], [38], [46], [45], [9], [39], [43].
- Imágenes [32], [48], [49], [41], [50].
- Música y sonidos [33], [15], [49], [42], [19].

Por otro lado, muchos trabajos utilizan experimentos que pueden ser probados en la vida real como jugar a un juego [51], [49], resolver un problema de matemáticas [46], aprender [44], pasear por la ciudad [12] y otros [11], [13], [29], [52].

El reconocimiento de emociones tiene un rango muy variado de aplicaciones, pero según la revisión literaria tiene dos ámbitos de aplicación principales: interacción humano-computadora y medicina [8]. En este primer campo se podría obtener una notable mejoría si las máquinas comprendieran las emociones de los humanos [10]. Sería posible establecer una conexión afectiva entre la máquina y el humano que permitiese que los asistentes virtuales pudieran atender a los humanos de una manera más natural y mejor [49]. También podría mejorar las recomendaciones para los resultados de búsquedas [42], la interfaz de usuario y el contenido [46], [47]. Los robots que comprendiesen las emociones podrían proporcionar una mejor experiencia

al sujeto [32], [53]. Las sugerencias del contenido televisivo podrían ser más adecuadas para el usuario [43]. Además, en el ámbito de los videojuegos las emociones tienen un papel fundamental, ya que contribuirían al proceso del diseño del mundo del juego basado en el estado emocional del jugador [49] y la experiencia del usuario se podría adaptar en función de la retroalimentación emocional haciendo el juego más realista [40]. Por otro lado, está el ámbito de la medicina que puede beneficiarse de la detección de emociones puesto que el sistema fisiológico y la capacidad de sentir están entrelazadas. La detección de emociones puede ayudar a monitorear nuestro bienestar [34], [38]. También se puede controlar el estrés a partir de la detección de emociones, por ejemplo para reducir la probabilidad de enfermedades cardiovasculares [16]. La salud mental también se vería beneficiada gracias a la detección de emociones [11], [9], [12]; la autoconciencia puede mejorar la salud mental [31]. Se ha demostrado que los estados emocionales negativos pueden tener un impacto perjudicial en nuestra salud [39]. Además, la detección de emociones podría ayudar a enseñar a los niños que sufren autismo a comprender mejor sus emociones [13], [14] y también se propone como una herramienta para ayudar a personas con trastornos emocionales [15]. Otras aplicaciones serían: detectar si un conductor tiene sueño [45], monitorear la actitud en un aula [44], ajustar la dificultad del contenido de una clase [51] o crear un canal emocional alternativo a través de la música [33].

A pesar de que no hay una gran cantidad de trabajos dedicados al reconocimiento de emociones (27 trabajos tras filtrar y quitar aquellos que no cumplen los criterios según la revisión literaria) [8], contamos con algunos de cierto interés. En [9] se experimenta con 21 personas de edades comprendidas entre los 29 y 39 años sin antecedentes de enfermedades psiquiátricas o problemas médicos complejos. Dicho experimento utiliza fragmentos de películas para provocar cuatro emociones⁴, lo cual fue más eficiente que otros métodos verificados por estudios previos. Concretamente, se utilizaron 4 tipos de vídeos cuya duración ronda entre 3 y 10 minutos cada uno. Estos vídeos se corresponden a una emoción distinta (ira, miedo, alegría y tristeza) y entre cada uno había un vídeo neutral de 30 segundos cuya función principal consistía en que el usuario no mezclase emociones. Tras la visualización de los vídeos, se pidió a los usuarios que rellenaran una encuesta para indicar la intensidad de cada emoción en una escala del 0 (nada) al 5 (extremo). Además, este estudio utilizaba un dispositivo que proporcionaba el electrocardiograma de cada usuario. En este estudio se concluyó que es viable reconocer emociones en intervalos de tiempo relativamente cortos. En cuanto a los resultados, se obtuvo un 50 % de *accuracy* en todas las emociones excepto en la de miedo que se consiguió un 25 %.

⁴Las cuatro emociones son: miedo, tristeza, ira y alegría

Otro trabajo de interés es [20], ya que utiliza otro estímulo diferente a los vídeos para detectar las emociones, concretamente la música. En este experimento se propone detectar la emoción de un grupo de personas con la ayuda de *wearables*, en particular, se usan pulseras inteligentes. Para dicho trabajo se tomaron señales fisiológicas de 32 participantes que escucharon 40 canciones. El objetivo era conseguir un modelo que detectara las emociones en función de la música que se estuviera reproduciendo en el momento y así saber cual era la canción más idónea para el conjunto de usuarios de la sala.

Por último, es importante mencionar otro tipo de trabajos que se focalizan más en la creación del dispositivo inteligente, usado posteriormente para la detección de emociones. Un claro ejemplo es [19], donde se muestra el diseño de un brazalete que combina biosensores, hardware con bajo coste energético y el codiseño del software. Los autores han creado un prototipo para demostrar que los algoritmos de redes neuronales pueden funcionar en este sistema y además se probó una aplicación de detección de emociones. En este trabajo, hacen una implementación para la detección de emociones con dos clases de salida: alegría y tristeza (seleccionando el mínimo tamaño de la red neuronal para conseguir 100 % de *accuracy*⁵). Se utilizó el *Augsburg Dataset* para la recopilación del conjunto de entrenamiento y prueba. Además, evaluaron diferentes topologías de red usando distintos valores para el número de características de entrada. El mejor *accuracy* se consiguió con 8 características. El entrenamiento y el test fue realizado con la librería *FANN*. Finalmente, se implementó la red en el microcontrolador y se recogieron las medidas.

Cabe destacar que en los trabajos de detección de emociones se han aplicado algoritmos de *machine learning* y *deep learning*⁶⁷ como son las redes neuronales convolucionales [51], [12], *LSTM* (*long short-term memory*) [45], [12], o redes neuronales con *backpropagation* [46]. Pero **sobre todo** se han utilizado algoritmos clásicos, de aprendizaje supervisado, siendo *Support Vector Machine* el más usado [8].

⁵Hay que tener en cuenta que se consigue un 100 % de *accuracy* pero solo con dos clases y además son opuestas.

⁶Los algoritmos de *deep learning* se han utilizado minoritariamente. Por lo que predominan los algoritmos clásicos de *machine learning*

⁷*Deep learning* pertenece al campo del *machine learning* y destaca por estar sobre todo basado en redes neuronales artificiales, las cuales utilizan múltiples capas de procesamiento para extraer características progresivamente.

Capítulo 3

Fundamentos teóricos

3.1. Inteligencia Artificial

La inteligencia artificial es un ámbito que está en pleno auge, por tanto es muy habitual que aparezca frecuentemente este término en los medios de comunicación. Pese a la abundante propaganda que recibimos diariamente sobre dicho campo de estudio, muchas personas no saben qué es realmente la inteligencia artificial. A priori, debemos saber que no es un campo que ha surgido recientemente. De hecho, su origen se remonta a poco después de la Segunda Guerra Mundial y se le asignó su nombre en 1956. Este campo es tan interesante que, junto a la biología molecular, es donde a la mayoría de científicos de otras disciplinas les gustaría trabajar [1]. También hace muy atractivo a este ámbito el hecho de que sea multidisciplinar y tenga numerosas aplicaciones en distintos campos de estudio. Otro motivo por el que desata gran interés es que todavía hay diversos problemas a resolver y mucho potencial para desarrollar.

En la imagen 3.1 se puede apreciar una tabla en la que se muestra los cuatro enfoques, con algunas definiciones, que se han seguido a lo largo de la historia respecto a la inteligencia artificial. Las definiciones que aparecen en la parte de arriba de la tabla hacen referencia al razonamiento y procesos mentales. En cambio, la parte de abajo se corresponde con la conducta. Por otro lado, tenemos la parte izquierda de la tabla que alude a la forma de actuar del ser humano. Mientras que la parte derecha se basa en la racionalidad. Hay que aclarar que cuando hablamos de racionalidad nos referimos a hacer lo correcto en función del conocimiento del que se disponga [1].

A pesar de haber sido los enfoques seguidos a lo largo de la historia para la inteligencia artificial, ha habido controversia entre las perspectivas centradas en la racionalidad y las centradas en los humanos. Esto se debe a que el enfoque humano es fundamentalmente una ciencia empírica que realiza hipó-

tesis y confirmaciones mediante la realización de experimentos. Mientras que el enfoque racional es una mezcla de ingeniería y matemáticas. Por tanto, estos grupos se han ayudado e ignorado el uno al otro a lo largo del tiempo [1].

Sistemas que piensan como humanos	Sistemas que piensan racionalmente
«El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985)	«El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985)
«[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje...» (Bellman, 1978)	«El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)
Sistemas que actúan como humanos	Sistemas que actúan racionalmente
«El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990)	«La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole <i>et al.</i> , 1998)
«El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)	«IA... está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)

Figura 3.1: Definiciones de inteligencia artificial organizadas en 4 categorías [1].

3.2. Machine learning

El *machine learning* o aprendizaje automático es un área que pertenece a la inteligencia artificial y básicamente trata de hacer que las máquinas mejoren en alguna tarea aprendiendo de los datos, en lugar de tener que codificar reglas explícitamente. En otras palabras, se podría decir que el aprendizaje automático es la ciencia (y el arte) de programar computadoras para que puedan aprender de los datos [2].

Una definición general de este ámbito sería la siguiente [2]:
 “*Machine learning* es el campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente”. *Arthur Samuel, 1959*

En cambio, una definición más orientada a la ingeniería sería [2]:
 “Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna tarea T y alguna medida de desempeño P, si su desempeño en T, medido por P, mejora con la experiencia E”. *Tom Mitchell, 1997*

Hay diferentes clasificaciones de esta área dependiendo del criterio que se utilice [2].

Dependiendo de si tienen supervisión humana o no podemos encontrar los siguientes tipos[2]:

- Aprendizaje supervisado: el conjunto de entrenamiento está etiquetado. Además, se trata de clasificación cuando se predicen valores discretos y regresión cuando se predicen valores continuos.
- Aprendizaje no supervisado: el conjunto de entrenamiento no está etiquetado.
- Aprendizaje semisupervisado: el conjunto de entrenamiento está parcialmente etiquetado (generalmente muchos datos sin etiquetar y algunos pocos datos etiquetados).
- Aprendizaje por refuerzo: El sistema de aprendizaje puede observar el entorno, seleccionar y realizar acciones. A cambio, obtiene recompensas o sanciones. Por tanto, debe aprender por sí mismo cuál es la mejor estrategia para obtener la mayor recompensa a lo largo del tiempo.

Dependiendo de si pueden o no aprender gradualmente sobre la marcha:

- Batch learning: equivale a cuando el modelo realiza el aprendizaje de manera no incremental.
- Online learning: corresponde cuando el modelo realiza el aprendizaje de manera incremental.

Dependiendo de que funcionen simplemente comparando nuevos puntos de datos con puntos de datos conocidos o que, en su lugar, detecten patrones en los datos de entrenamiento y creen un modelo predictivo, como lo hacen los científicos:

- Basado en instancia: el sistema aprende los ejemplos de memoria, luego generaliza a nuevos casos usando una medida de similitud.
- Basado en modelo: generaliza a partir de un conjunto de ejemplos, construyendo un modelo de estos y luego usa ese modelo para hacer predicciones.

La primera aplicación de *machine learning* que tuvo bastante popularidad, facilitando la vida a muchas personas, tuvo lugar en la década de 1990:

ésta fue el filtro de spam [2].

Por último, se podría decir que el problema de *machine learning* a resolver en este trabajo pertenece a aprendizaje supervisado, particularmente se trata de un problema de clasificación. Ya que tenemos las etiquetas de las instancias y cada una representa a una clase discreta.

Capítulo 4

Algoritmos

A continuación, se explicarán los algoritmos de aprendizaje utilizados en los distintos modelos del experimento. Gran parte de las explicaciones están basadas en la información contenida en el libro [2]. Es relevante saber que algunos algoritmos valdrán tanto para tareas de clasificación como de regresión, entre otras. Pero al estar enfrentándonos a un problema de multclasificación, siempre se explicará el funcionamiento correspondiente a clasificación y si es necesario se harán aclaraciones para adaptarlo a multclasificación.

4.1. Support Vector Machine

Support Vector Machine (SVM) [2] es un modelo de *machine learning* capaz de tratar problemas de clasificación lineal o no lineal, regresión e incluso detección de outliers. Las SVMs son idóneas para problemas de clasificación que tienen conjuntos de datos complejos pero de tamaño pequeño o mediano. Debido a la naturaleza del problema que se va a abordar, resulta interesante el enfoque para problemas de clasificación, por tanto se va a explicar el modelo referente a dicha perspectiva.

Es importante saber que las SVMs suelen tener una precisión más alta que un perceptrón o una neurona artificial [3]. Esto hace que sea un modelo atractivo para diversas aplicaciones y a veces se combinen con redes convolucionales.

4.1.1. Clasificación lineal de SVM

La explicación del SVM se va a apoyar en la figura 4.1, ya que así se puede hacer de manera más visual e intuitiva. En la imagen de la izquierda perteneciente a la figura 4.1 se observa que las dos clases del conjunto de entrenamiento se pueden separar con las líneas continuas. Esto se debe a que son linealmente separables. Cada línea de la imagen representa un límite de

decisión correspondiente a un modelo. Se puede apreciar que el modelo que representa la línea discontinua es malo, ya que no separa correctamente las clases. En cambio, los modelos correspondientes a las líneas continuas separan correctamente las clases. Estos dos últimos modelos cuentan con una peculiaridad y es que sus límites de decisión se encuentran tan cerca de las instancias que probablemente estos modelos no clasifiquen tan bien nuevas instancias. En la imagen de la derecha de la figura 4.1 hay una línea continua que representa el límite de decisión de un clasificador SVM. Se puede apreciar que dicha línea separa bien ambas clases y se mantiene lo más alejada posible de las instancias más cercanas (las rodeadas), por lo que se resuelve el problema mencionado anteriormente sobre clasificar mal nuevas instancias. Por tanto, se puede pensar que el clasificador SVM se ajusta buscando la mayor distancia entre las clases. A esto se le conoce como clasificador de margen máximo y a las dos instancias rodeadas de la figura 4.1 se les llama vectores de soporte: son las instancias más cercanas entre una clase y otra.

Cabe destacar que, aunque se añadan más instancias de entrenamiento, si éstas son más lejanas a la línea que separa las clases que los vectores de soporte, no afectarán al límite de decisión. Esto se debe a que dicho límite está exclusivamente determinado por los vectores de soporte.

Es importante resaltar tres conceptos:

- Los vectores de soporte: Son los puntos más cercanos entre una clase y otra.
- El margen: Es la distancia entre el límite de decisión y los vectores de soporte.
- Clasificador de margen máximo: Es el límite de decisión que consigue el mayor margen posible.

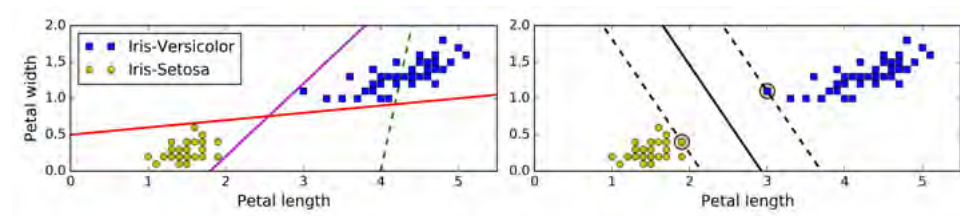


Figura 4.1: Conjunto de datos iris con varios clasificadores [2]

En definitiva, el SVM busca obtener el clasificador de margen máximo. A pesar de que hay muchas maneras de implementarlo, lo que básicamente hace es detectar primero los puntos más cercanos entre una clase y otra,

después encuentra la línea recta que los conecta y por último se traza una frontera perpendicular que divide a esta línea recta en dos.

Margen de clasificación suave

Anteriormente se ha descrito SVM para el caso ideal, es decir, las condiciones son idóneas porque hay dos clases que se pueden separar perfectamente con una línea recta y, consecuentemente, con el clasificador de margen máximo. Pero, desafortunadamente, no siempre vamos a tener dichas condiciones. El margen que se ha utilizado hasta ahora no permite instancias dentro de este y exige que cada una esté colocada en el lado correcto. Esto se conoce como clasificación de margen duro y tiene dos principales inconvenientes:

- Solo funciona si los datos son linealmente separables.
- Es muy sensible a outliers (valores atípicos).

A continuación, se van a ejemplificar estos dos principales inconvenientes mediante imágenes para facilitar su comprensión.

La distribución de los datos en la imagen de la izquierda pertinente a la figura 4.2 es igual a la de la figura 4.1 pero con la incorporación de un outlier. Se puede observar que debido a la posición del outlier no es posible encontrar un margen duro. En cambio, en la imagen de la derecha de esta misma figura el outlier tiene otra posición que nos permite establecer un margen duro. Si comparamos el límite de decisión obtenido en las imágenes de la derecha de la figura 4.1 y 4.2 vemos que son muy distintos. Probablemente, el correspondiente a la figura 4.1 generalice mejor que el de la figura 4.2.

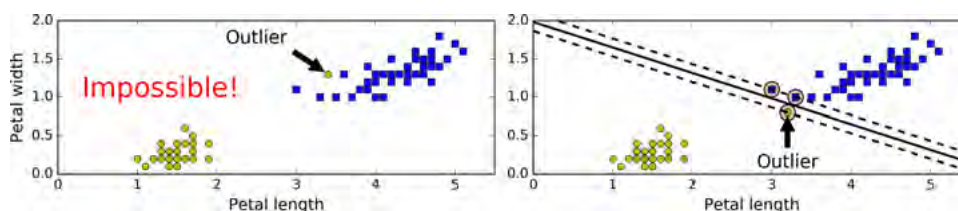


Figura 4.2: Conjunto de datos iris con outliers [2]

Como se puede intuir, para solucionar ambos problemas se podría optar por el uso de un margen menos restrictivo, es decir más flexible. Entonces, habría que buscar un equilibrio entre tener el mayor margen posible y limitar las violaciones de este. A esto se le llama clasificación de margen suave.

Para buscar el equilibrio entre tener el mayor margen posible y limitar las violaciones existe un parámetro C . Concretamente se trata de un hiperparámetro, por lo que debe configurarse antes del entrenamiento y permanece

constante durante este. Hay que tener en cuenta que un C menor genera márgenes más amplios y conforme C aumente se generan márgenes más pequeños. Por tanto, el valor de C se escoge de manera empírica buscando el menor error posible. Cabe destacar que si el modelo SVM tiene *overfitting*¹, se puede intentar regularizar reduciendo C .

En la figura 4.3 tenemos dos imágenes correspondientes una a un valor alto de C y otra a un valor bajo. En la imagen de la izquierda, cuyo valor del parámetro es alto, observamos que tiene un margen más pequeño que la imagen de la derecha que cuenta con un valor de C más bajo. Por tanto, la imagen de la izquierda tiene menos violaciones en el margen que el caso de la derecha. Mientras que el modelo de la derecha probablemente generalice mejor. Incluso en este conjunto de entrenamiento se puede observar que comete menos fallos el modelo de la derecha, aunque tenga más violaciones en el margen, ya que hay más instancias en el lado del límite de decisión correspondiente que en el modelo de la derecha.

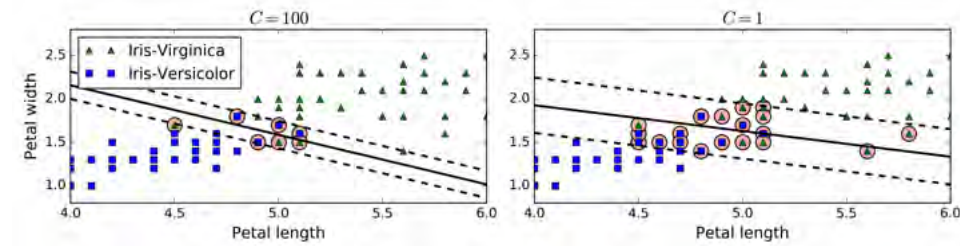


Figura 4.3: [2]

4.1.2. Clasificación no lineal de SVM

Hasta ahora se ha tratado el modelo de *Support Vector Machine* para el caso de clasificación lineal. Sin embargo, no siempre va a ser este el escenario presente en nuestros problemas, ya que puede ser necesario algo más que una línea recta para clasificar las instancias. Por tanto, en este apartado se va a abordar la aplicación de este modelo para problemas no lineales.

En la figura 4.4 se puede apreciar que las instancias no pueden ser clasificadas mediante una frontera lineal, por tanto se usa una frontera no lineal. Entonces, la pregunta sería: ¿Cómo se hace esto si con SVM hasta ahora solo se ha visto el uso de fronteras lineales?. La respuesta es sencilla: agregando más dimensiones a cada instancia, hasta tener una disposición que permita

¹Cuando hay *overfitting* el modelo funciona bien con el conjunto de entrenamiento, pero no generaliza bien. Por tanto, con nuevos datos no funcionará tan bien. El *overfitting* ocurre cuando el modelo es demasiado complejo en relación a cantidad o ruido en el conjunto de entrenamiento.

aplicar *Support Vector Machine* como se estaba usando anteriormente, es decir hasta que los datos sean linealmente separables. Véase como en la figura 4.5 se ha aplicado dicha transformación del espacio.

Ahora la dificultad recae en cómo añadir dimensiones a los datos sin tener un elevado coste computacional. Puesto que añadir características polinomiales es una buena opción para aumentar la dimensionalidad pero un grado de polinomio bajo no permite manejar conjuntos de datos muy complejos y con un grado de polinomio alto se obtiene un modelo demasiado lento debido a la complejidad que conlleva. Por tanto, se utiliza una técnica llamada el truco del Kernel: permite obtener el mismo resultado que si se agregasen características polinomiales, incluso un alto grado, sin tener que añadirlo. Esto hace que sea una técnica de gran utilidad debido a las prestaciones que aporta de manera muy eficiente.

En la figura 4.6 se clarifica cual sería el proceso a seguir por el truco del kernel. Básicamente, lo que se hace es mapear el conjunto de datos no separable a un espacio de mayores dimensiones con una función no lineal. Una vez que se obtiene el conjunto de datos linealmente separable, se aplica el modelo de SVM como se hacía anteriormente para obtener el límite de decisión óptimo. Por último, se hace la transformada inversa para aplicar la clasificación. Realmente, en la práctica el truco del Kernel no implementa todos estos pasos, ya que el mapeo y el cálculo del hiperplano se calculan de una manera simplificada. Para ello, se usa álgebra lineal con la finalidad de hacer más eficiente y rápido el proceso.

Cabe destacar que las transformaciones más usadas en el truco del Kernel se suelen apoyar en el uso de funciones lineales y de funciones gaussianas.

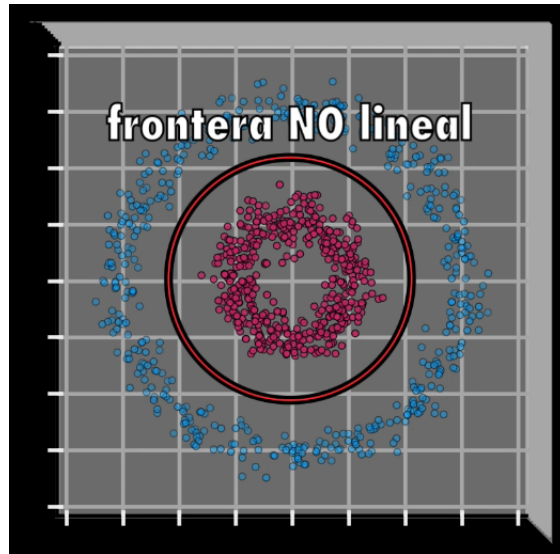


Figura 4.4: *SVM* con frontera no lineal [3]

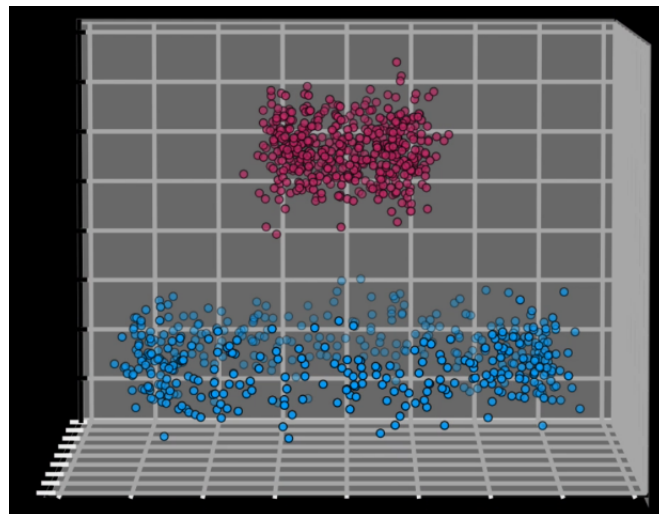


Figura 4.5: Cambio de dimensión [3]

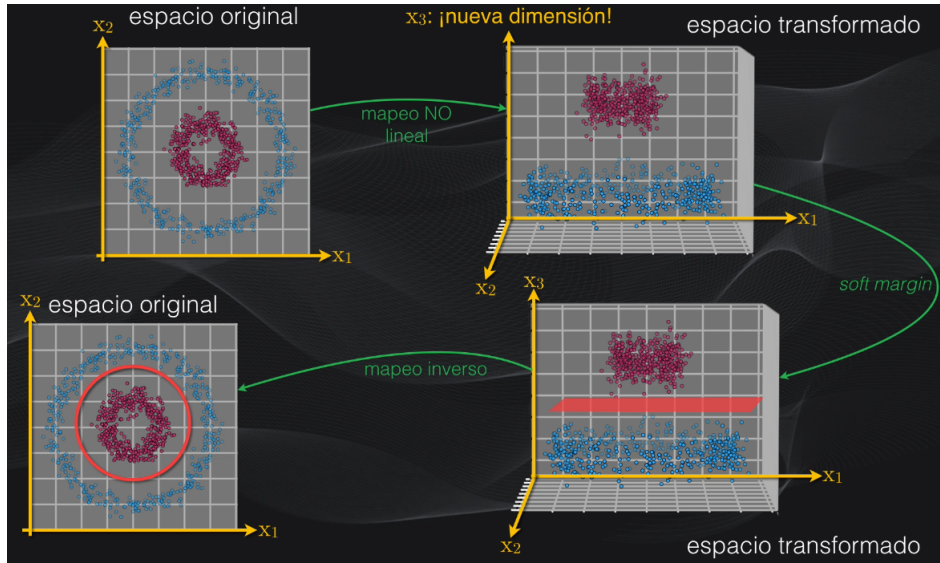


Figura 4.6: *SVM* con truco del Kernel [3]

4.1.3. Multclasificación

En su forma más simple *SVM* no soporta la multclasificación de forma nativa. Es decir, para clasificación trata solo el caso binario. Por tanto, para resolver multclasificación, se utiliza el mismo principio tras descomponer el problema en varios problemas de clasificación binaria [4].

La idea consiste en mapear los datos a un espacio de altas dimensiones para hacer que las dos clases sean linealmente separables. Esto se llama aproximación *One-to-One* y por tanto, se descompone el problema de multclasificación en varios problemas de clasificación binaria. Entonces, se usa un clasificador binario por cada par de clases.

Otra aproximación para la resolución de este problema es *One-to-Rest*. En esta caso, la descomposición se reduce a un clasificador binario por cada clase.

Un único *SVM* puede resolver un problema de clasificación binaria y diferenciar entre dos clases. Así, de acuerdo a las dos aproximaciones, para clasificar puntos de un conjunto de datos de m clases:

- En la aproximación *One-to-Rest*, el clasificador puede usar m *SVMs*. Cada uno predice la pertenencia de los datos a cada clase.
- En la aproximación *One-to-One*, el clasificador puede usar $\frac{m(m-1)}{2}$ *SVMs*.

Tomemos como ejemplo un problema de clasificación con 3 clases: verde, rojo y azul. Ilustrado en la figura 4.7.

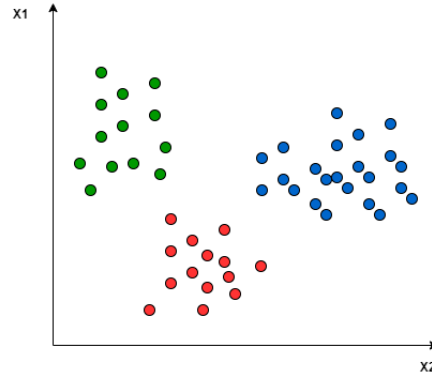


Figura 4.7: Conjunto de datos de 3 clases distintas: verde, rojo y azul [4]

Si se aplican las dos aproximaciones explicadas anteriormente a este conjunto de datos, ocurre lo siguiente:

En la aproximación *One-to-One*, se necesita un hiperplano para separar entre cada par de clases, descuidando los puntos de la tercera clase. Esto significa que la separación solo tiene en cuenta los puntos de las dos clases de la separación actual. Por ejemplo, la línea roja y azul intenta maximizar la separación sólo entre los puntos rojos y azules, obviando los puntos verdes. Este ejemplo se ilustra en la figura 4.8.

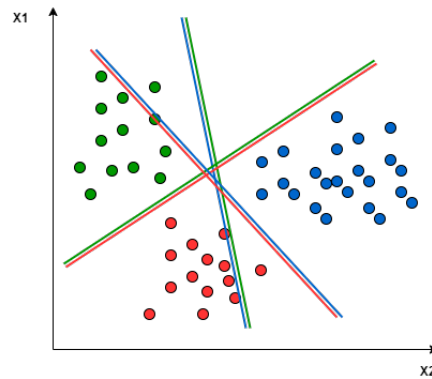


Figura 4.8: Ejemplo de multclasificación con *One-to-One* [4]

Por otro lado, En la aproximación *One-to-Rest*, se requiere de un hiperplano para separar entre una clase y las demás. Esto significa que la separación tiene en cuenta todos los puntos, dividiéndolos en dos grupos: uno para los puntos de una clase y otro grupo para el resto. Por ejemplo,

la línea verde trata de maximizar la separación entre los puntos verdes y el resto. En la figura 4.9 se puede ver este ejemplo.

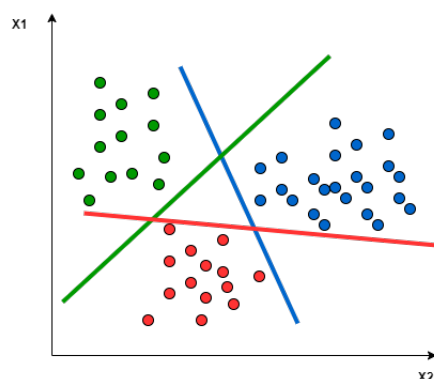


Figura 4.9: Ejemplo de multclasificación con *One-to-Rest* [4]

4.2. Random Forest

4.2.1. Árboles de decisión

Los árboles de decisión son los componentes fundamentales de *Random Forest*, por tanto el primer paso será explicarlos para facilitar la comprensión del algoritmo.

Al igual que *Support Vector Machine*, los árboles de decisión son algoritmos versátiles de *machine learning* que pueden realizar tanto tareas de clasificación como de regresión. Se tratan de algoritmos muy potentes que son capaces de ajustar conjuntos de datos complejos [2].

En la figura 4.10 se puede apreciar un árbol de decisión. Un ejemplo de cómo se realizaría una predicción sería el siguiente. Se tiene una flor que se quiere clasificar, por tanto se empieza en el nodo raíz (profundidad 0): y se hace la pregunta de si la longitud del pétalo es menor o igual a 2'45 cm. Si la respuesta es sí, entonces se llega a un nodo hoja que es el de color naranja y la flor se clasificaría como setosa. En caso contrario, se iría al nodo de abajo a la derecha y en función de si la anchura del pétalo de la flor es menor o igual a 1'75 cm o no lo es, se clasificará como versicolor o virginica. El atributo *samples* corresponde al número de instancias de entrenamiento que se ha aplicado, por ejemplo, hay 100 instancias de entrenamiento que tienen una longitud de pétalo mayor que 2'45 cm (profundidad 1, derecha), de las cuales 54 tienen un ancho de pétalo inferior o igual a 1'75 cm (profundidad 2, izquierda). El atributo de *value* de un nodo indica a cuántas instancias de entrenamiento de cada clase se aplica éste: por ejemplo, el nodo morado se

aplica a 0 setosa, 1 versicolor y 45 virginica. Por último, el atributo gini de un nodo mide su impureza: un nodo es “puro” ($\text{gini} = 0$) si todas las instancias de entrenamiento que aplica pertenecen a la misma clase. Por ejemplo, el nodo naranja es puro porque solo se aplica a instancias de entrenamiento de setosa, por tanto su puntuación de gini es 0. La puntuación de gini de un nodo i se calcula así: $G_i = 1 - \sum_{k=1}^n p_{i,k}^2$ donde $p_{i,k}$ es la proporción de instancias de clase k entre las instancias de entrenamiento en el i -ésimo nodo. Por ejemplo, el nodo izquierdo, de profundidad 2, tiene la siguiente puntuación de gini: $1 - (0/54)^2 - (49/54)^2 - (5/54)^2 \approx 0.168$.

En la figura 4.11 se pueden apreciar los límites de decisión que establece el árbol. Hay que tener en cuenta que la profundidad máxima se ha establecido en 2, por lo que el árbol de decisión se detiene ahí. Sin embargo, si se hubiera establecido la profundidad máxima en 3, los dos nodos de profundidad 2 agregarían, cada uno, otro límite de decisión (representado por las líneas de puntos).

Como se puede ver, los árboles de decisión son bastante intuitivos y sus decisiones son fáciles de interpretar. Incluso las reglas de clasificación que proporcionan son simples y se podrían aplicar manualmente si fuese necesario. Estos modelos a menudo se denominan modelos de caja blanca.

Los árboles de decisión también pueden estimar la probabilidad de que una instancia pertenezca a una clase k . Para esto, primero se recorre el árbol para encontrar el nodo hoja correspondiente a esta instancia y luego devuelve la proporción de instancias de entrenamiento de la clase k en este nodo. Un posible ejemplo sería el siguiente: si se ha encontrado una flor cuyos pétalos miden 7 cm de largo y 1.4 cm de ancho, le correspondería el nodo izquierdo de profundidad 2. Entonces el árbol de decisión generaría las siguientes probabilidades: 0% para setosa (0/54), 90.7% para versicolor (49/54) y 9.3% para virginica (5/54). Evidentemente, si se pide que se prediga la clase, debería indicar que pertenece a la clase versicolor, ya que tiene la probabilidad más alta.

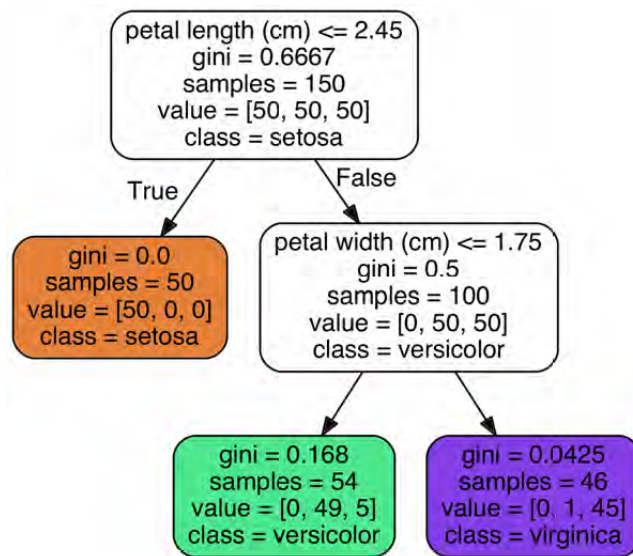


Figura 4.10: Árbol de decisión de Iris [2]

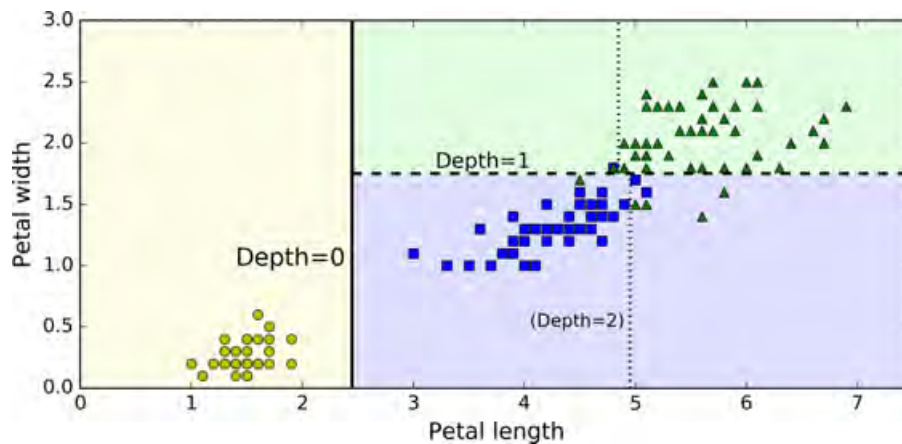


Figura 4.11: Límites del árbol de decisión de Iris [2]

Algoritmo *CART*

El *CART* (*Classification And Regression Tree*) es un algoritmo muy popular utilizado para entrenar árboles de decisión. De hecho es el algoritmo que utiliza *Scikit-Learn* [2].

Una de las ventajas que tiene este algoritmo es que está basado en una idea simple. Lo primero que hace es dividir el conjunto de entrenamiento en dos subconjuntos, usando para esto una sola característica k y un umbral t_k (por ejemplo, longitud del pétalo ≤ 2.45 cm). ¿En qué se basa para elegir k y

t_k ? Para escogerlo busca la pareja de k y t_k que produzca los subconjuntos más puros (ponderados por su tamaño). Esta es la función de costo que el algoritmo intenta minimizar:

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

Donde $\begin{cases} G_{left/right} \text{ mide la impureza del subconjunto izquierdo/derecho} \\ m_{left/right} \text{ es el número de instancias en el subconjunto izquierdo/derecho.} \end{cases}$

Una vez que se ha dividido con éxito el conjunto de entrenamiento en dos, se divide los subconjuntos usando la misma lógica, luego los subsubconjuntos y así sucesivamente, de forma recursiva. La recursividad termina cuando se alcanza la profundidad máxima (definida por el hiperparámetro *max_depth*), o si no puede encontrar una división que reduzca la impureza. Hay otros hiperparámetros que controlan la condición de detención aunque no son tan usuales como por ejemplo el mínimo número de divisiones.

Por desgracia, encontrar el árbol óptimo tiene una complejidad demasiado elevada. Por lo tanto, hay que conformarse con encontrar una solución razonablemente buena.

4.2.2. Random Forest

Imaginemos que se hace una pregunta compleja a miles de personas seleccionadas al azar y luego se agregan las respuestas. En muchos casos, la respuesta obtenida será mejor que la respuesta de un experto. Esto se podría extrapolar al *machine learning*: si se recogen las predicciones de un grupo de predictores (como clasificadores o regresores), con frecuencia se obtendrán mejores predicciones que con el mejor predictor individual. Un grupo de predictores se llama *ensemble*, por tanto, esta técnica se conoce como *Ensemble Learning* y un algoritmo de *Ensemble Learning* se le llama *Ensemble Method* [2].

Random Forest es un ensemble method para resolver problemas de clasificación, regresión y otras tareas que operan mediante la construcción de una multitud de árboles de decisión en el momento del entrenamiento. Básicamente, *Random Forest* consiste en entrenar un conjunto de árboles de decisión, cada uno en un subconjunto aleatorio distinto dentro del conjunto de entrenamiento. Para obtener el resultado final, basta con obtener las predicciones de todos los árboles individuales y entonces predecir la clase que obtiene la mayor cantidad de votos. A pesar de la simplicidad de este

algoritmo, es relevante señalar que se trata de uno de los que más potencial tienen en *machine learning*.

4.3. Regresión logística

Regresión logística [2] es un algoritmo de aprendizaje que se utiliza tanto para problemas de regresión como de clasificación. Este algoritmo en clasificación trata de estimar la probabilidad de que una instancia pertenezca a una clase en particular. Por ejemplo, podría calcular la probabilidad de que un correo pertenezca a spam. En el caso de que la probabilidad estimada esté por encima del 50 %, el modelo predice que la instancia pertenece a dicha clase (llamada clase positiva y etiquetada como “1”) o, si estuviera por debajo, predice que no (entonces, corresponde a la clase negativa y etiquetado como “0”). En este caso tendríamos un clasificador binario.

Para realizar las estimaciones, este algoritmo, al igual que un modelo de regresión lineal, calcula una suma ponderada de las características de entrada, más un término de sesgo. A diferencia de la regresión lineal, que genera el resultado directamente, la regresión logística genera la logística de este resultado. La probabilidad estimada del modelo de regresión logística (forma vectorizada) se define como:

$$\hat{p} = h_{\theta} = \sigma(\theta^T \cdot x)$$

La logística, señalada como $\sigma()$, es una función sigmoide (es decir, en forma de S) que genera un número entre 0 y 1. Se define como se muestra en la figura 4.12 y en la siguiente ecuación:

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

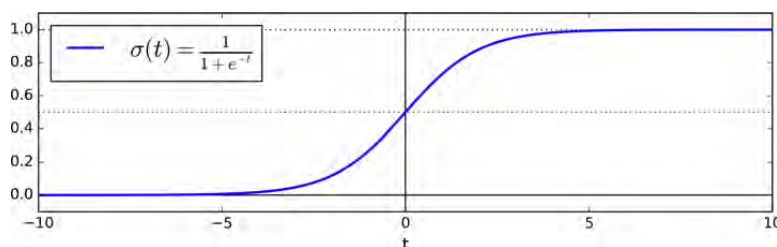


Figura 4.12: Función logística [2]

Una vez que el modelo de regresión logística ha estimado la probabilidad $\hat{p} = h_{\theta}$ de que una instancia x pertenezca a la clase positiva, puede hacer su

predicción \hat{y} fácilmente:

$$\hat{y} = \begin{cases} 0 & \text{Si } \hat{p} < 0.5 \\ 1 & \text{Si } \hat{p} \geq 0.5 \end{cases}$$

En la figura 4.12 se puede apreciar que $\sigma(t) < 0.5$ cuando $t < 0$, y $\sigma(t) \geq 0.5$ cuando $t \geq 0$, por lo que un modelo de regresión logística predice 1 si $\theta^T \cdot x$ es positivo y 0 si es negativo.

4.3.1. Multclasificación

El modelo de regresión logística se puede generalizar para admitir varias clases directamente, sin tener que entrenar y combinar múltiples clasificadores binarios. Esto se llama Regresión Softmax o Regresión logística multinomial. La idea es bastante sencilla: cuando se le da una instancia x , el modelo de regresión Softmax primero calcula una puntuación $s_k(x)$ para cada clase k , luego estima la probabilidad de cada clase aplicando la función softmax (también llamada exponencial normalizada) a la puntuaciones. La ecuación para calcular $s_k(x)$ debería parecer familiar, ya que es como la ecuación para la predicción de regresión lineal.

$$s_k(x) = \theta_k^T \cdot x$$

Cuando se haya calculado la puntuación de cada clase para la instancia x , se puede estimar la probabilidad \hat{p}_k de que la instancia pertenezca a la clase k ejecutando las puntuaciones a través de la función softmax: calcula la exponencial de cada puntuación, luego los normaliza (dividiendo por la suma de todos los exponenciales).

$$\hat{p}_k = \sigma(s(x))_k = \frac{\exp(s_k(x))}{\sum_{j=1}^k \exp(s_j(x))}$$

- k es el número de clases.
- $s(x)$ es un vector que contiene las puntuaciones de cada clase para la instancia x .
- $\sigma(s(x))_k$ es la probabilidad estimada de que la instancia x pertenezca a la clase k dados los puntajes de cada clase para esa instancia.

Capítulo 5

Experimentos y discusión

5.1. Fases del experimento

5.1.1. Selección de los vídeos

El primer paso de este experimento fue buscar un vídeo por cada emoción básica de las seis que hay según Ekman y Friesen [28]. Además, también era necesario uno correspondiente a neutralidad. Para encontrarlos se buscó en el artículo [54], donde viene cada vídeo catalogado con su correspondiente emoción. Asimismo, se especificaba a que fragmento de la película correspondía el vídeo, por lo que se buscó cada película y se recortó la parte correspondiente. Para escoger los vídeos de las distintas emociones se consultó con personas que no iban a participar en el experimento, para seleccionar aquellos vídeos que más se adecuaban a cada emoción (teniendo en cuenta que tuviera la etiqueta correspondiente). Esta fue la elección final:

- Tristeza: fragmento de Cry Freedom.
- Neutro: fragmento de Blue.
- Alegría: fragmento de Alive.
- Sorpresa: fragmento de Capricorn One.
- Miedo: fragmento de Scream.
- Asco: fragmento de Pink Flamingos.
- Ira: fragmento de American History X.

Tras seleccionar los fragmentos de cada película, lo siguiente que se hizo fue juntarlos en este orden: neutral, tristeza, neutral, alegría, neutral, sorpresa, neutral, miedo, neutral, asco, neutral, ira. La razón por la que se comienza con un vídeo neutral es para que el sujeto se tranquilice y empiece la prueba

en un estado objetivo. También se utilizan vídeos neutrales tras cada emoción para que estas no se mezclen y el participante no tenga interferencias. La duración total del conjunto de vídeos es de 21 minutos y 11 segundos.

5.1.2. Recolección de los datos y protocolo

Para la recolección de datos se ha citado a un total de 26 personas (15 hombres y 11 mujeres de diversas edades) por separado, siendo el número de participantes restringido debido a la situación que padecemos por la pandemia. De estas 26 muestras se han eliminado 3 debido a problemas que han surgido durante la recolección de datos, por tanto el número final de sujetos es 23. Aunque puede parecer una muestra pequeña, no es así, ya que por cada participante se ha obtenido una gran cantidad de muestras respectivas a cada emoción. Esto último se explicará con más detenimiento en el pre-procesado de los datos.

Cada participante ha sido examinado en la misma sala intentando que tuvieran las mismas condiciones dentro de lo posible. En la habitación se ha procurado que siempre hubiera la misma iluminación, por lo que se evitaba la luz exterior y la prueba se realizaba con la luz apagada. Además, para mantener la misma temperatura la puerta estaba cerrada y se activaba el aire. Nadie podía interrumpir la prueba, ni acceder a la habitación, bajo ningún concepto y en caso de que esto ocurriera se anulaba sin volverse a repetir.

Lo primero que se le hacía a cada sujeto al entrar en la habitación era indicarle el asiento y pedirle que se tranquilizara mientras le colocaba los *wearables*. Primero le ponía la pulsera y le pedía que dejase la mano apoyada en el brazo del asiento para que este dispositivo no provocase problemas durante la prueba. A continuación, le colocaba la diadema y comprobaba que la conexión fuese correcta. Una vez que los dispositivos estaban bien ajustados, le pedía al sujeto que no se moviera durante la prueba para no alterar el funcionamiento de los *wearables* y activaba la grabación con cada uno, entonces comprobaba su correcto funcionamiento durante unos minutos. Tras esto iniciaba el vídeo y me colocaba a su lado observando el ordenador para hacer las respectivas marcas con el móvil en el archivo csv de la diadema. Hacía 3 marcas en cada vídeo respectivo a las emociones básicas y en el segundo vídeo neutral. La primera marca se correspondía al inicio del vídeo, la segunda al comienzo del momento intenso de la emoción del vídeo respectivo y la tercera era referente al final del vídeo que coincidía casualmente con el final del momento intenso. Por tanto, con la tercera marca conocía ambos finales, ya que realmente era el mismo. Cabe destacar que en el vídeo neutral no hay momento intenso, por lo que la marca 2 se hacía al inicio del vídeo, es decir, justo después de la marca 1, y la marca 3 se hacía al final, evidentemente. El uso de estas marcas se explicará detenidamente en el siguiente subapartado.

Una vez que finalizaba el vídeo, detenía la grabación de los dispositivos y le hacía unas preguntas al usuario para rellenar un cuestionario basado en el del trabajo [9]. Dicho cuestionario consistía en una encuesta para indicar la intensidad de cada emoción en una escala del 0 (mínimo) al 5 (máximo), de tal manera que así podría saber si los vídeos estaban causando las emociones deseadas.

5.1.3. Tratamiento de los datos

Unión de los datos de la diadema y pulsera

Una vez realizadas las pruebas y obtenidos los datos, el siguiente paso es tratarlos para que se pueda trabajar correctamente con ellos. Lo primero que se debe hacer es leer los datos de los csv proporcionados por los dispositivos. Es importante tener en cuenta que ambos aportan estos documentos en un formato distinto, por lo que hay que ajustarlos a un formato común. Concretamente, se adaptarán los datos procedentes de la pulsera al formato seguido por el archivo que proporciona la diadema. Esto se hace así porque la diadema por cada sujeto aporta un solo archivo csv cuyo contenido está organizado de manera más intuitiva y manejable. Esto se debe a que cada fila tiene la fecha y hora de cada muestreo y aparecen los valores de las distintas características en su correspondiente columna. Mientras que la pulsera cuenta con un archivo diferente por cada señal fisiológica y además, la primera fila hace referencia a la fecha y hora inicial a la que se comienza a grabar y la segunda a la frecuencia de muestreo en hercios (Hz). Las demás filas corresponden a los datos de cada muestreo, los cuales no van acompañados de la hora a la que se captaron.

Para realizar la lectura de datos se comienza por los archivos referentes a la pulsera. De esos se seleccionan solo aquellos que aportan información útil al problema, particularmente son:

- TEMP: Datos del sensor de temperatura corporal expresada en grados en la escala Celsius ($^{\circ}\text{C}$).¹
- EDA: Datos del sensor de actividad electrodérmica expresados como microsiemens (μS).²
- BVP: Datos del fotopleletismógrafo.³
- HR: Datos correspondientes al promedio de la frecuencia cardíaca.

¹La escala Celsius es una escala de temperatura que asigna el valor cero (0°C) al agua en proceso de fusión, y el valor cien (100°C) al agua en proceso de ebullición.

²Se denomina siemens (S) a la unidad derivada del sistema internacional para la medida de la conductancia eléctrica (G). Su inversa es la resistencia eléctrica (R).

³El fotopleletismógrafo permite medir la presión sanguínea de una persona de forma no invasiva.

Es de suma importancia saber que estos cuatro parámetros son útiles y se suelen usar con frecuencia en este tipo de experimentos según advierte la revisión literaria [8].

Por otro lado se ignoran:

- ACC: Datos del sensor acelerómetro de 3 ejes.
- IBI: Datos del tiempo entre los latidos del corazón de los individuos extraídos de la señal de BVP.
- Tags: Marcas realizadas durante la grabación.

El ACC no se tiene en cuenta porque los participantes durante la prueba deben estar quietos para no perjudicar el funcionamiento de los dispositivos por lo que no tiene ningún sentido tener en cuenta la aceleración, ya que va a tener un valor de cero. Las etiquetas (Tags) tampoco aportan información útil, ya que no tienen que ver con las señales fisiológicas y para delimitar cuando empieza o acaba un vídeo uso las de la diadema. El IBI podría parecer una buena opción pero este parámetro no sigue una frecuencia de muestreo, por lo que se hace complejo integrarlo con los demás datos de acuerdo al formato que se sigue. Además, se obtiene del BVP y este ya es considerado, por tanto, podría ser información redundante y hay que tener en cuenta que es fundamental reducir, en medida de lo posible, la complejidad computacional con la información que no sea estrictamente necesaria.

Tras la selección de los archivos, correspondientes a los parámetros relevantes de la pulsera, se pasa la fecha y hora inicial de formato *timestamp* a *date time* y se calcula para cada muestra. Esto se hace pasando la frecuencia de hercios a segundos y sumándola a la hora inicial tantas veces como instancias haya en el archivo. Una vez hecho esto, se almacena cada archivo en un *dataframe* distinto donde cada fila consta de la fecha y hora de la muestra, y la medida correspondiente al archivo. En este momento los datos de la pulsera estarían en el mismo formato que los de la diadema.

A continuación, se eliminan aquellos parámetros que no son relevantes en la diadema para el estudio de las emociones: la batería, el acelerómetro, la conexión, el giroscopio ⁴, entre otros. Solamente nos quedamos con los datos respectivos a las ondas cerebrales en sus distintas frecuencias (Delta_TP9, Delta_AF7, Delta_AF8, Delta_TP10, Theta_TP9, Theta_AF7, Theta_AF8, Theta_TP10, Alpha_TP9, Alpha_AF7, Alpha_AF8, Alpha_TP10, Beta_TP9, Beta_AF7, Beta_AF8, Beta_TP10, Gamma_TP9, Gamma_AF7,

⁴El giroscopio mide la cantidad de rotación desde la última posición en la que se encontraba la diadema. No es relevante para el problema, ya que el sujeto no debe moverse durante la prueba.

Gamma_AF8, Gamma_TP10), los datos en bruto de la electroencefalografía (RAW_TP9, RAW_AF7, RAW_AF8, RAW_TP10) de los cuatro sensores que tiene la diadema y las marcas (elements) servirán para saber los límites de los vídeos. Las ondas cerebrales e información relacionada con estas, se suele usar en este tipo de experimentos según afirma la revisión literaria [8].

Entonces, se procede a unir los datos de la diadema y de la pulsera cronológicamente. Si una hora coincide en ambos dispositivos, evidentemente aparece una sola vez con los datos de ambos. En caso de que no coincida para algún dispositivo, dicho parámetro se queda vacío en la correspondiente columna. Con la finalidad de alinear los diferentes datos recopilados, provenientes de los diferentes sensores que trabajan a distinta frecuencia, es necesario realizar un alineamiento o *resampling*⁵ de los datos. Esto se llevará a cabo más adelante.

Etiquetado de las instancias

Una vez que se tienen los datos de ambos dispositivos juntos, lo siguiente es proceder a etiquetar la muestra. Por tanto, se separan los datos en distintos *dataframe* de tal manera que haya dos por cada emoción. Para esta tarea las marcas realizadas con la diadema durante el experimento son fundamentales. Esto se explicó concretamente en el apartado anterior. Lo que se hace es crear un *dataframe* por cada pareja de marcas 1 y 3, esto corresponde a los datos de los vídeos completos de cada emoción. Lo mismo se hace con las marcas 2 y 3 que hacen referencia a los datos de los momentos intensos de los vídeos. Tras esto habrá siete parejas de marcas 1 y 3, y otras siete de marcas 2 y 3. Son siete porque hay una por cada emoción básica y además el estado neutro. El motivo de que se tenga en cuenta los datos referentes a los momentos intensos y a los vídeos completos es porque se trabajará con ambos a la hora de realizar modelos, pero esto se explicará más adelante detalladamente.

Cuando se tienen los siete *dataframe* correspondientes a los datos de los vídeos completos, se aplica *resampling* y se almacenan estos *dataframes* en orden en un vector de tal manera que se sabe que el primero corresponde a tristeza, el segundo a neutro, el tercero a alegría, el cuarto a sorpresa, el quinto a miedo, el sexto a asco y el último a ira. Entonces, se pasan los datos a una matriz y se etiqueta cada fila del *dataframe*. Gracias a que estos están en orden se sabe la etiqueta que corresponde en cada momento. Todo este proceso es análogo en otra matriz para los datos de los momentos intensos de los vídeos. Finalmente, se obtienen dos matrices, una correspondiente a los vídeos completos con 49191 instancias y otro a los momentos intensos

⁵Se alinean los datos básicamente para que estén en la misma frecuencia. Después se interpolará los valores faltantes (en caso de que los haya tras el alineamiento).

con 15640 instancias, donde cada fila se corresponde a una instancia y cada columna a una característica. La última columna se corresponde a la etiqueta de dicha instancia.

Preprocesado de datos

El preprocesado que se aplica a ambos conjuntos de datos, tanto a los correspondientes a los vídeos completos como al de los fragmentos intensos, es muy similar pero con algunos matices distintos. Este apartado es muy relevante puesto que el preprocesado se encarga de mantener la calidad de los datos que posteriormente serán usados para los modelos.

Ambos conjuntos de los datos se han separado en un conjunto de training y otro de test. El primero se usa para entrenar el modelo y el segundo para testear este. La tasa de error en casos diferentes a aquellos con los que se ha entrenado se llama error de generalización (error fuera de la muestra) y este valor nos indica cómo de bien actuará el modelo en instancias nunca vistas anteriormente. El conjunto de test constituye el 20 % del conjunto de datos y el de entrenamiento se corresponde al 80 %. A partir de ahora, en este apartado cuando se haga referencia al conjunto de los vídeos completos y fragmentos intensos, nos estaremos refiriendo a su respectivo conjunto de entrenamiento que es a lo que se aplica el preprocesado. Excepto para la estandarización que se aplica al conjunto entero de datos, es decir tanto al de entrenamiento como al de test (también conocido como conjunto de prueba).

Para estos dos conjuntos de datos se aplica el tratamiento de *outliers*⁶, estandarización y tratamiento de desbalanceo de datos.

El tratamiento de *outliers* es una práctica común cuyo objetivo es tratar aquellos valores atípicos que pueden perjudicar el funcionamiento del modelo. Podemos observar en la figura 5.1, como en la quinta fila (contando de arriba hacia abajo) y primeras cuatro columnas (de derecha a izquierda) se aprecia que la escala en el eje x llega hasta 750000, esto indica que por ahí rondan algunos valores. Por lo tanto, la distribución de estas características demostrarían la presencia de valores atípicos en los datos relativos a los fragmentos intensos. En el caso de los vídeos completos podemos ver la figura 5.2, en la quinta fila, primera y tercera columna ocurre lo mismo que en el caso anterior y por tanto estas características delatan la existencia de *outliers*. En esa misma fila pero en la columna segunda y cuarta se podría apreciar también algo pero no tan exagerado como en los casos anteriores. Para tratar los outliers en ambos caso, se ha optado por eliminar alrededor del 4 %⁷ de los datos con más valores atípicos.

⁶Los *outliers* son valores atípicos de la muestra de datos.

⁷Se elimina un 4 % de los datos para tratar los *outliers* porque estos suelen rondar el

Una de las transformaciones más importantes a aplicar a los datos es el escalado de características, ya que salvo algunas excepciones los modelos de *machine learning* no funcionan bien cuando las entradas numéricas tienen escalas muy diferentes [2]. Hay dos maneras de mantener los atributos en la misma escala: estandarizando ⁸ o normalizando ⁹. En las figuras 5.2 y 5.1 se puede apreciar que la mayoría de distribuciones de los datos tienen una forma Gaussiana. Es este el motivo por el que se estandariza en vez de normalizar.

En las figuras 5.4 y 5.3 se puede apreciar que hay desbalanceo ¹⁰ en las distintas clases. Para tratar el desbalanceo se indica en los parámetros de cada modelo y además se utiliza la métrica *balanced accuracy* ¹¹.

En particular, el desbalanceo para *Random Forest* se trata de la siguiente manera. Si el parámetro *class_weight* toma valor “balanced”, utiliza los valores de “y” (etiquetas) para ajustar automáticamente los pesos de forma inversamente proporcional a las frecuencias en los datos de entrada como:

$$\frac{\text{num_muestras}}{\text{num_clases} \cdot \text{num_elementos_clase}}$$

.

En *SVM* y Regresión Logística el parámetro *class_weight* funciona de la misma forma.

Por otro lado, el *balanced accuracy* se basa en dos métricas más comunes: sensibilidad (también conocida como tasa de verdaderos positivos o *recall*) y especificidad (también conocida como tasa de verdaderos negativos, o 1 - tasa de falsos positivos). La sensibilidad responde a cuántos de los casos positivos se han detectado de todos los positivos que hay (en porcentaje). La especificidad responde a cuantos de los casos negativos se han detectado de todos los negativos que hay (en porcentaje). El *balanced accuracy* es una

5 % de los datos. Es decir, un 10 o 20 % no se consideraría *outliers* porque son muchos datos. Por tanto, se ha ido ajustando el parámetro de la función hasta aproximarse al 5 % (sin pasarme al 10 % o más).

⁸Se resta la media a los datos y se divide entre la varianza de tal forma que la distribución tenga varianza unitaria.

⁹Los valores se desplazan y reescalan para que acaben estando en un rango entre 0 y 1.

¹⁰Cuando se dice que hay desbalanceo quiere decir que hay una diferencia sustancial en el número de datos de una clase respecto a las demás.

¹¹*Balanced accuracy* tiene en cuenta el desbalanceo de las clases a la hora de calcular los resultados.

media aritmética simple de ambas medidas:

$$\frac{sensibilidad + especificidad}{2}$$

. Se ha explicado para el caso binario, aunque se puede extender sencillamente a multclasificación con una estrategia *One-Versus-Rest*.

Por último en las figuras 5.6 y 5.5 se pueden ver las matrices de correlación ¹² ¹³ de los dos conjuntos de datos. En general la correlación entre características no es muy alta, por lo que no se considera necesario eliminar ninguna característica de las presentes.

¹²Si la correlación es muy alta quiere decir las características son proporcionales.

¹³La matriz de correlación muestra todas las características pero no las nombra todas, es decir, nombra una sí y otra no. La matriz muestra las características en el siguiente orden: Delta_TP9, Delta_AF7, Delta_AF8, Delta_TP10, Theta_TP9, Theta_AF7, Theta_AF8, Theta_TP10, Alpha_TP9, Alpha_AF7, Alpha_AF8, Alpha_TP10, Beta_TP9, Beta_AF7, Beta_AF8, Beta_TP10, Gamma_TP9, Gamma_AF7, Gamma_AF8, Gamma_TP10, RAW_TP9, RAW_AF7, RAW_AF8, RAW_TP10, BVP, HR, TEMP, EDA

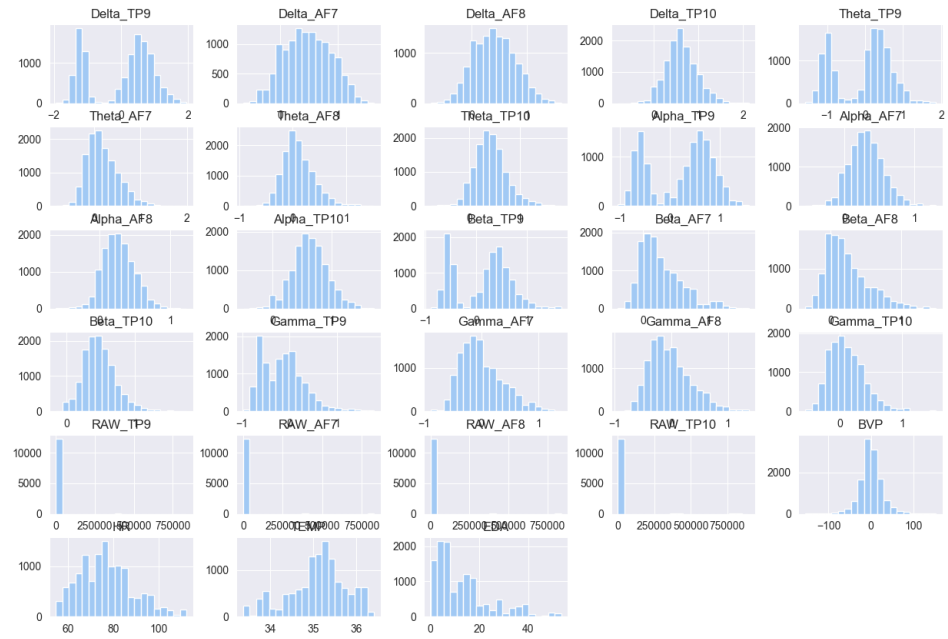


Figura 5.1: Histogramas correspondientes a la distribución de las características de los datos referentes a los fragmentos intensos

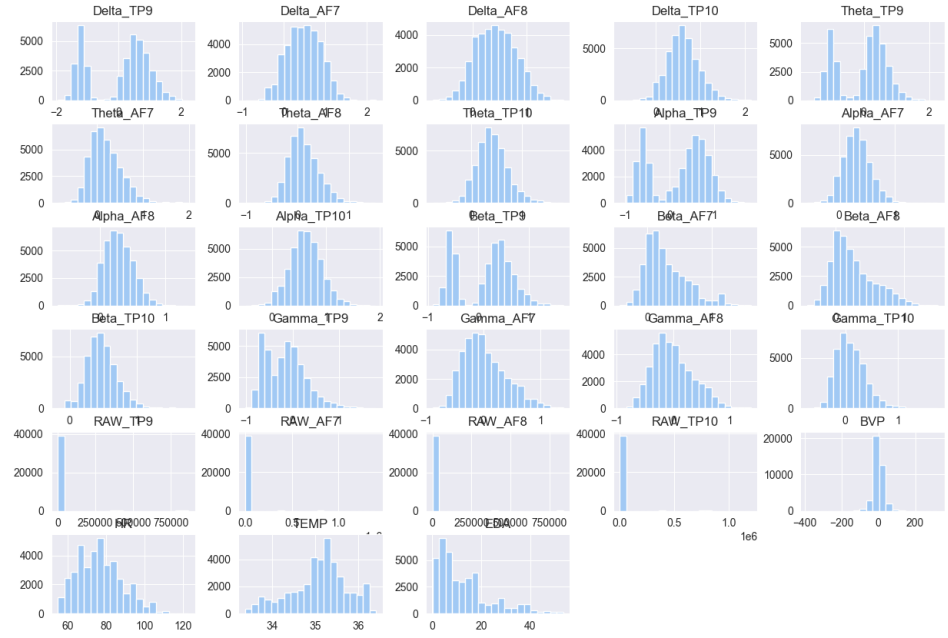


Figura 5.2: Histogramas correspondientes a la distribución de las características de los datos referentes a los vídeos completos

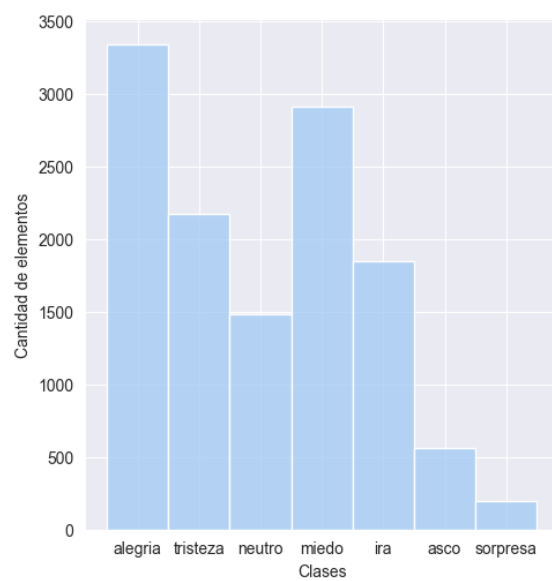


Figura 5.3: Histograma de clases correspondiente a los datos de los fragmentos intensos

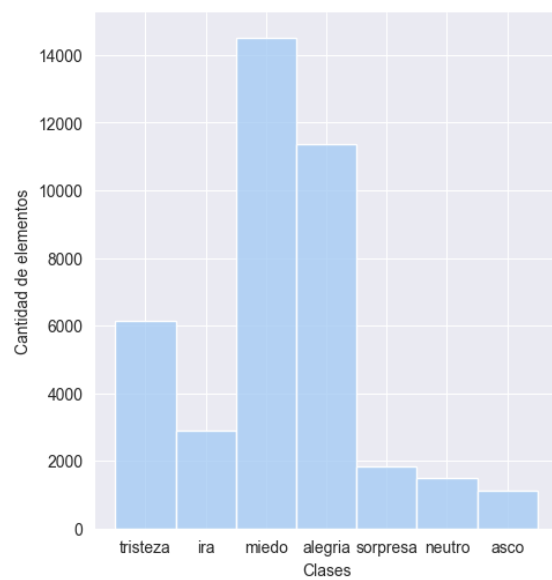


Figura 5.4: Histograma de clases correspondiente a los datos de los vídeos completos

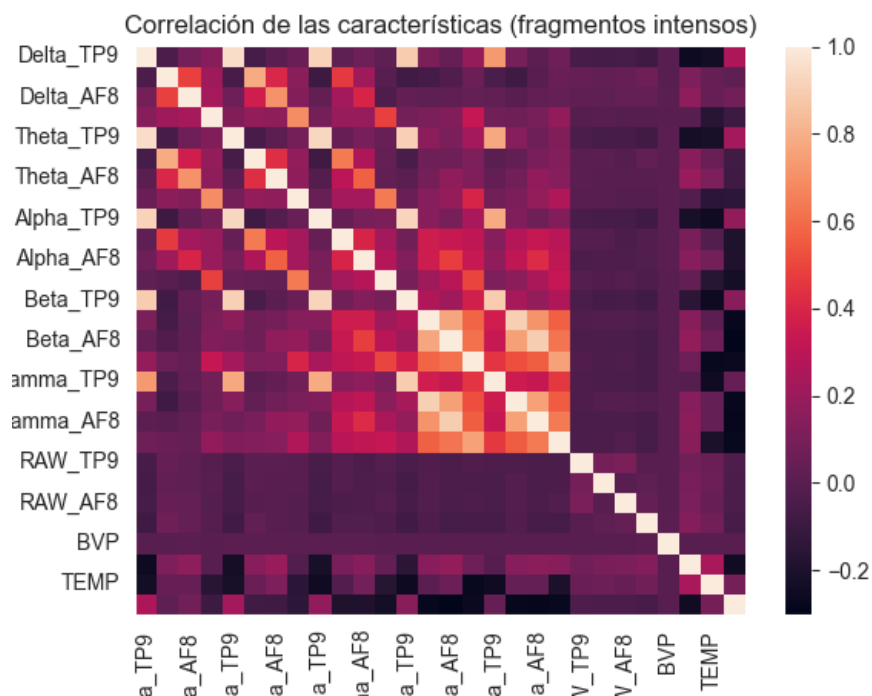


Figura 5.5: Matriz de correlación correspondiente a los datos de los fragmentos intensos

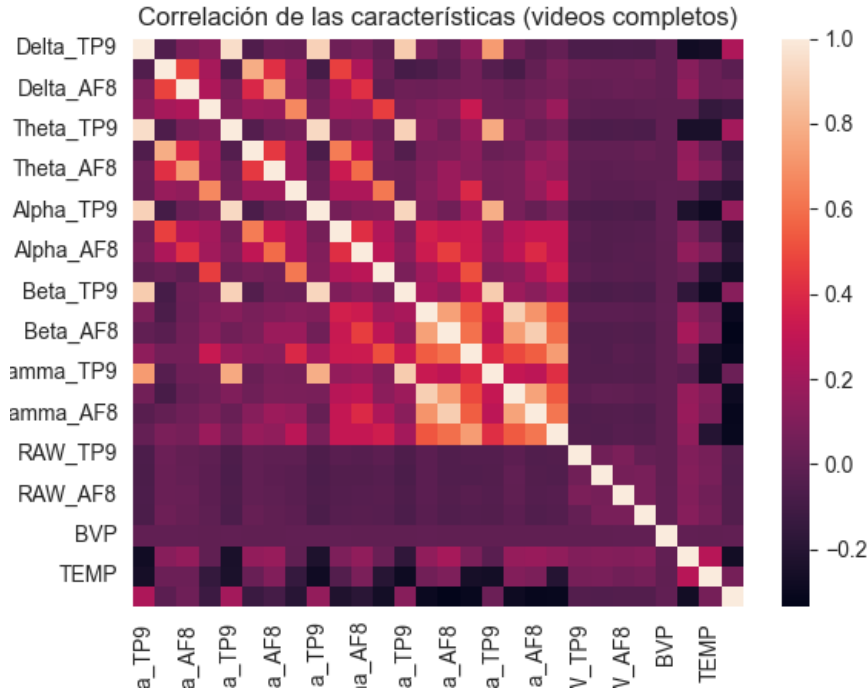


Figura 5.6: Matriz de correlación correspondiente a los datos de los vídeos completos

5.1.4. Modelos

En este trabajo se prueban varios algoritmos de aprendizaje: regresión logística, *Support Vector Machine* y *Random Forest*. Estos se aplican tanto al conjunto de datos correspondiente a los momentos intensos como al conjunto referente a los vídeos completos. Además, se prueban varios parámetros hasta encontrar el óptimo.

Se utilizan algoritmos clásicos porque son los que más se usan en este tipo de problemas según la revisión literaria, entre ellos se destaca *Support Vector Machine* [8]. *Random forest* y SVM son opciones muy interesantes porque están capacitados para resolver problemas complejos. Las redes neuronales también tienen esta capacidad pero un gran inconveniente que tienen es que a veces actúan como “caja negra” haciendo difícil interpretar los resultados. Por otro lado, está regresión logística que se suele usar como modelo base.

Los parámetros utilizados para los datos de los fragmentos intensos:

- Regresión logística. C : 0.001, 0.01, 1, 10, 100, 1000, 10000, 100000.

- *Support Vector Machine*. C : 2^{-2} , 2^{-1} , 2^0 , 2^1 , 2^2 , 2^3 , 2^4 , 2^5 . Γ : 2^{-5} , 2^{-4} , 2^{-3} , 2^{-2} , 2^{-1} , 2^0 , 2^1 .
- *Random Forest*. max_depth : 5, 10, 20, 30, 40, 50, 100, 200. $n_estimators$: 10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000.

Para los datos de los vídeos completos:

- Regresión logística. C : 0.001, 0.01, 1, 10, 100, 1000, 10000.
- *Support Vector Machine*. C : 2^{-2} , 2^{-1} , 2^0 , 2^1 , 2^2 , 2^3 , 2^4 , 2^5 . Γ : 2^{-5} , 2^{-4} , 2^{-3} , 2^{-2} , 2^{-1} , 2^0 , 2^1 .
- *Random Forest*. max_depth : 10, 20, 50, 100, 200. $n_estimators$: 50, 100, 200, 300, 400, 500, 600.

Las pruebas con los distintos parámetros se han realizado con *grid search* que es una función que se encarga de probar todas las combinaciones posibles de los parámetros indicados. Además, a veces se han ido añadiendo valores si el mejor parámetro era justamente el inicial o el final, ya que eso podía ser una posible señal de que otro parámetro que no estuviera en el rango especificado constituyera el óptimo. Por ejemplo, si el último valor para C en regresión logística es 100000 y más adelante no hay ninguno, sería necesario probar el valor 1000000, ya que pudiera ser que todavía no se hubiera alcanzado el óptimo.

-

5.1.5. Validación y métricas

Si se entrena varias veces el conjunto de entrenamiento buscando mejoras y luego se comprueba con test si las ha habido, y tras esto se vuelve a repetir dicho proceso hasta decidir parar, habría un problema. Este sería que se ha medido el error de generalización ¹⁴ varias veces en el conjunto de test, por tanto, se ha adaptado el modelo y los hiperparámetros para producir el mejor funcionamiento para el conjunto de test. Es decir, el modelo probablemente no va a mejorar tan bien con nuevos datos cómo ha mejorado con el conjunto de test. Una solución para esto, es tener un segundo conjunto llamado **conjunto de validación**. De esta forma, se entrenan varios modelos con diversos hiperparámetros usando el conjunto de entrenamiento, entonces se selecciona el modelo y los hiperparámetros que mejor actúan en el conjunto de validación. Una vez se esté satisfecho del todo con el modelo entonces se

¹⁴El error de generalización hace referencia al error fuera de la muestra, es decir al error obtenido por el modelo con nuevos datos

ejecuta una sola vez con el conjunto de test para obtener una estimación del error de generalización.

Una opción mejor que la validación es usar *cross validation*, puesto que si se usa varios modelos no se tiene que “malgastar” tantos datos, si no que con la rotación que se produce no habría problema. Además, se evita un posible “sesgo” de que por casualidad la combinación del conjunto de entrenamiento y validación funcione mejor, ya que al probar con varias combinaciones y luego hacer la media se obtiene un resultado final más fiable. En la figura 5.7 se puede apreciar una imagen explicativa.

Para los modelos de este trabajo se ha usado *5 fold cross validation*. En cuanto a la métrica se utiliza *balanced accuracy*. Además, se utiliza matriz de confusión¹⁵ para poder ver cuándo una clase es confundida con otra, esto permite trabajar de forma separada con distintos tipos de error.

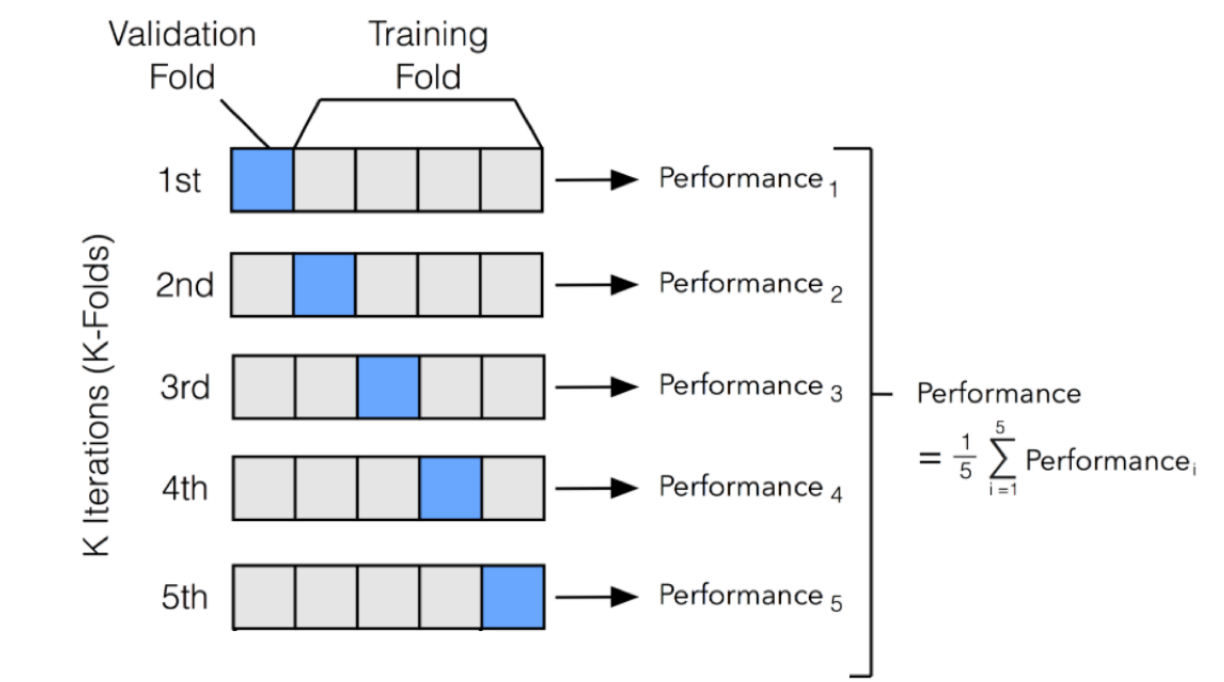


Figura 5.7: *cross validation* [5]

¹⁵Cada columna de la matriz de confusión representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los principales beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

5.2. Resultados

Los parámetros que aportan mejores resultados para los datos de los fragmentos intensos son:

- Regresión logística. C : 10.
- *Support Vector Machine*. C : $2^4 = 16$ y Gamma : $2^{-4} = 0.0625$.
- *Random Forest*. max_depth : 20. $n_estimators$: 900.

Concretamente, los resultados para los mejores parámetros para los datos de los fragmentos intensos son:

- Regresión logística. *Balanced accuracy*: 0.323.
- *Support Vector Machine*. *Balanced accuracy*: 0.885.
- *Random Forest*. *Balanced accuracy*: 0.9126.

Los parámetros que aportan mejores resultados para los datos de los vídeos completos son:

- Regresión logística. C : 1.
- *Support Vector Machine*. C : $2^4 = 16$ y Gamma : $2^{-3} = 0.125$.
- *Random Forest*. max_depth : 20. $n_estimators$: 500.

Concretamente, los resultados para los mejores parámetros para los datos de los vídeos completos son:

- Regresión logística. *Balanced accuracy*: 0.321.
- *Support Vector Machine*. *Balanced accuracy*: 0.833.
- *Random Forest*. *Balanced accuracy*: 0.943.

Estos resultados se han obtenido con *cross validation* y además de para validar, se usan para comparar entre los modelos cual es el mejor. Una vez se sabe el mejor modelo para cada conjunto de datos se calcula el error de generalización para hacernos una idea de como funcionarán con datos que no han sido entrenados (para eso se ha reservado el conjunto de test). Este paso se hará al final del apartado.

La Regresión Logística se ha utilizado porque suele usarse en los problemas como un primer modelo a superar posteriormente. Además, era buena idea tener un algoritmo que funcione bien con atributos que sean más o menos lineales y clases que sean linealmente separables porque para el caso contrario ya están los otros dos algoritmos. También, otra ventaja de este es que el resultado del modelo se puede interpretar como una probabilidad. Esto puede llegar a ser muy interesante, ya que se puede utilizar, por ejemplo, para hacer un ranking y no solamente una clasificación.

Por otro lado, está *Support Vector Machine* que destaca por ser idóneo para problemas de clasificación que tienen conjuntos de datos complejos pero de tamaño pequeño o mediano. En este caso se cuenta con un tamaño de datos mediano, ya que un conjunto de datos ronda las 49000 instancias y el otro las 15000. Asimismo, aparenta ser un conjunto de datos complejo, ya que cuenta con 28 características y por tanto con una dimensionalidad considerable. Esto último beneficia al algoritmo porque trabaja bien en espacios que poseen un número notable de dimensiones. Otro fuerte de *SVM* es que trabaja bien con problemas cuyas clases no son linealmente separables, por lo tanto en este caso cubre ese “hueco” que deja Regresión Logística.

Random Forest es la última alternativa y es que al ser un *Ensemble Method* usa varios predictores, lo cual hace que sea un algoritmo de los que más potencial puede llegar a alcanzar en *machine learning*. Una de las ventajas que tiene es que no espera que las relaciones o dependencias sean lineales, o que los atributos interactúen de forma lineal. Al igual que *SVM*, suelen trabajar bien en problemas complejos y en espacios de alta dimensionalidad.

Tras estas explicaciones se puede concluir algunas cosas en los resultados. En ambos conjuntos de datos los modelos que mejor han funcionado han sido los que utilizan *Random Forest* como algoritmo de aprendizaje. *Support Vector Machine* también ha tenido un buen desempeño. Esto puede ser debido a que ambos algoritmos de aprendizaje suelen funcionar bien en problemas complejos y con una dimensionalidad considerable. Mientras que con Regresión Logística no suele ocurrir lo mismo, ya que funciona mejor en problemas de menor complejidad.

Otro detalle importante, es que posiblemente los datos no sean linealmente separables. Solo hay que ver la diferencia tan desmesurada que hay entre los resultados que aporta el modelo que usa Regresión Logística y los otros dos modelos. El hecho de que los datos no sean linealmente separables hace, con una gran probabilidad, que regresión logística no vaya a obtener resultados decentes. En cambio, los otros dos modelos se adaptan mucho mejor a estos casos, ya que por ejemplo *SVM* cuenta con el truco del kernel para adaptarse a problemas no lineales.

En la figura 5.8 se puede observar la matriz de confusión del modelo que usa *Random Forest* para los datos correspondientes a los fragmentos intensos. Si nos fijamos en la diagonal nos percatamos de que en general hay una buena clasificación para la mayoría de las clases. Para interpretar esta matriz hay que tener en cuenta el histograma que contiene el número de elementos por clase 5.3, ya que la de asco a simple vista podría parecer que se clasifica mal y no es así. Si nos fijamos, tiene un 0.3 de fallos frente a un 4 de aciertos, lo cual está muy bien, por tanto el problema es que tiene poca cantidad de elementos respecto a las demás clases como se aprecia en el histograma 5.3. En cambio, tanto la clase de alegría como la de miedo no cometen ningún fallo pero la de alegría aparece con un color más amarillo debido a que tiene mayor cantidad de instancias. Llama la atención en la matriz la clase sorpresa, debido a que tiene un 0.7 de aciertos frente a 0.6 de clasificaciones erróneas. Sobre todo se etiquetan muestras correspondientes a sorpresa como si fuesen de alegría, concretamente un 0.5, lo cual es una cantidad bastante alta. En cambio, un 0.1 son etiquetadas como si fueran muestras de miedo. En general, teniendo en cuenta el desbalanceo¹⁶ presente que se puede apreciar en el histograma 5.8 se puede afirmar que excepto para la clase de sorpresa (también hay que tener en cuenta que es la que menos instancias tiene) la clasificación es bastante acertada en las demás clases. Por otra parte, en la figura 5.9 se halla la matriz de confusión del modelo que usa *Random Forest* para los datos referentes a los vídeos completos. Teniendo en cuenta el desbalanceo en este conjunto de datos 5.4, se puede afirmar que la clasificación de instancias es bastante buena para cada clase, ya que las que tienen un color más oscuro en la matriz de confusión 5.9 no es porque cometan una cantidad de errores significativa (de hecho ninguna clase comete una cantidad de errores “notable”), sino que poseen una considerablemente menor cantidad de elementos.

Después de analizar la matrices de confusión de los dos mejores modelos, observamos que ambas son bastante buenas, aunque parece ser que la mejor opción es el modelo que ofrece *Random Forest* con vídeos completos debido a que tiene un mejor *balanced accuracy*. Por tanto, finalmente este es el modelo seleccionado y se procede a calcular el error de generalización para hacernos una idea de como funcionaría con datos que no han sido entrenados (para eso se ha reservado el conjunto de test). El *balanced accuracy* obtenido con el conjunto de test es 0.9417. Como se puede concluir tras ver este último resultado, en el modelo final no hay *overfitting*, ya que el *balanced*

¹⁶Hay que tener cuidado porque el desbalanceo se trata en cada modelo (para ambos conjuntos de datos) y además, mediante el uso de la métrica *balanced accuracy*. Pero la matriz de confusión no tiene en cuenta ni la métrica, ni lo que aplica cada modelo. Por tanto, sería erróneo pensar que el desbalanceo no se ha tratado y que ha afectado a los modelos y a la predicción.

accuracy obtenido en la etapa de validación (con los datos de entrenamiento y realizando *cross validation*) es 0.943 frente a 0.9417 de balanced accuracy con el conjunto de test (error de generalización). Es decir, apenas hay diferencia. Para que hubiera *overfitting* tendría que haber un resultado mucho mejor en validación que el que proporcionase el modelo con el conjunto de test.

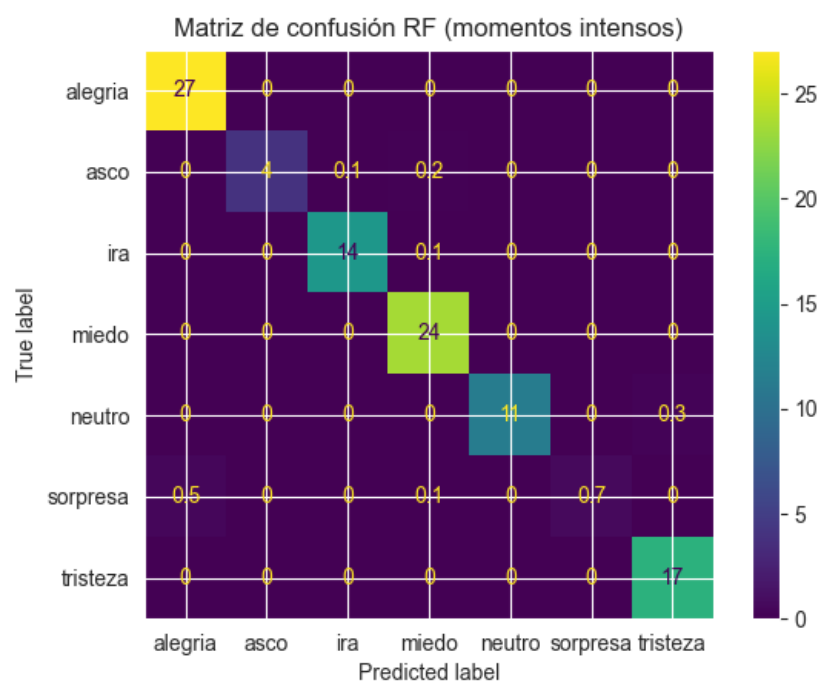


Figura 5.8: Matriz de confusión del modelo que usa *Random Forest* para los datos de los fragmentos intensos

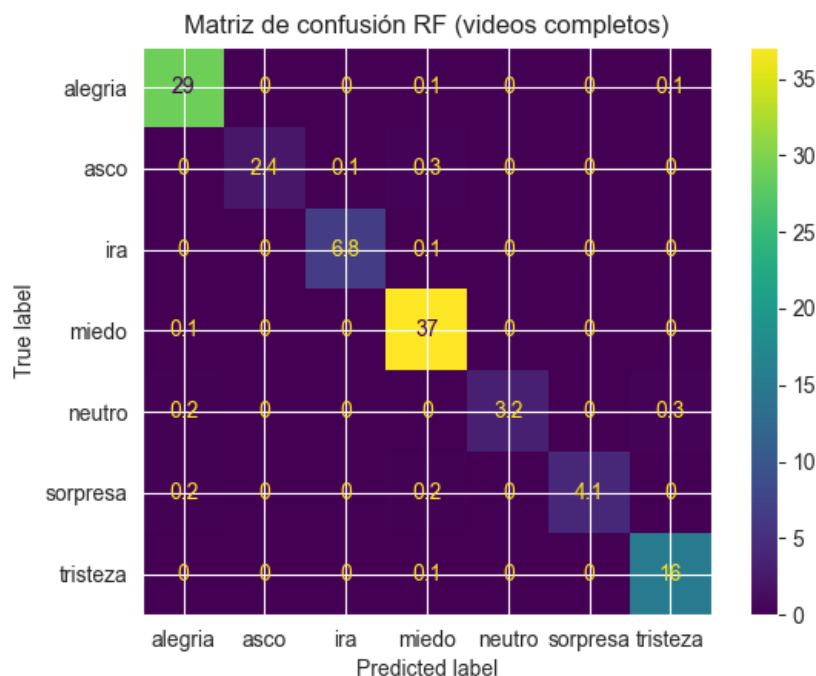


Figura 5.9: Matriz de confusión del modelo que usa *Random Forest* para los datos de los vídeos completos

5.3. Lenguaje de programación librerías

Para cerrar este capítulo se va a comentar algunas consideraciones sobre el lenguaje de programación y las librerías.

El lenguaje de programación utilizado para este proyecto es python. En cuanto, a las librerías se han usado las siguientes:

- pandas: se ha utilizado principalmente para trabajar con los *dataframe*.
- numpy: se ha usado para poder utilizar diferentes estructura de datos.
- seaborn: se ha utilizado para la representación de histogramas y matrices de correlación.
- matplotlib.pyplot: su principal función ha sido generar gráficos.
- sklearn: ha sido fundamental para la creación de los modelos de *machine learning* y el preprocesado de datos.

Capítulo 6

Conclusión

La detección de emociones es un problema complejo que requiere de una gran cantidad de estudio e investigación, por lo que todavía queda mucho trabajo por delante. Se podría decir que apenas está en sus inicios y es un tema con bastante potencial a desarrollar. Aun así, en este proyecto se ha demostrado que es posible clasificar las seis emociones básicas y el estado neutral con unos resultados bastante exitosos (94 % de *balanced accuracy*). Probablemente, parte de este triunfo se deba a que se ha intentado suplir las carencias generales (justo se comentaban al inicio de la memoria) de los proyectos de esta misma temática. La rigurosidad con la que se han llevado a cabo las pruebas para mantener las mismas condiciones experimentales (dentro de lo que se podía controlar) ha sido un factor fundamental. Puesto que los datos son la base del modelo y si estos tuvieran demasiadas interferencias o los sujetos hubieran sido expuestos a condiciones muy dispares, no se podría garantizar una buena calidad en la recolección de datos. Para futuros trabajos relacionados, queda margen de mejora por explorar, por ejemplo, sería idóneo contar con un laboratorio profesional y quizás valorar la inclusión de algún otro dispositivo más. Con esta última consideración se podría descubrir alguna otra señal fisiológica o parámetro que pudiese ser determinante.

En este proyecto se obtuvieron dos conjuntos de datos: el respectivo a los vídeos completos y el correspondiente a los fragmentos intensos de cada vídeo. Esto se hizo con la idea de poder desarrollar modelos con ambos datos, tal y como se ha hecho, e intentar obtener mayores garantías de éxito. Lo llamativo es que antes de realizar el experimento completo parecería intuitivo pensar que el modelo con los fragmentos intensos va a aportar mejores resultados debido a que las clases van a ser más “puras”. Pero se sabe que finalmente el modelo que mejores resultados aporta es el que utiliza el conjunto de datos referente a los vídeos completos y el algoritmo de aprendizaje de *Random Forest*. Durante la encuesta con los participantes, la gran

mayoría coincidió en que todos los vídeos completos provocaban la emoción correspondiente durante toda o casi toda su duración (excepto el de tristeza que al principio según los sujetos no transmitía la emoción indicada), lo cual quiere decir que realmente sirve el vídeo entero para obtener muestras representativas de su etiqueta correspondiente (es decir, con los datos de los vídeos completos tenemos muchas más instancias “útiles” que con el otro, aproximadamente 49000 frente a 15000). Asimismo, hay que tener en cuenta que las emociones son graduales, es decir, se pueden experimentar con más o menos intensidad y en este caso el conjunto de vídeos completos podía tener un rango más amplio de intensidades. Estas dos ideas pueden ser las consecuencias de que el mejor modelo use el conjunto de datos referente a los vídeos completos. Evidentemente, se sabe que cada conjunto de datos ha recibido por separado su tratamiento y que a ambos se le ha aplicado los mismos algoritmos de aprendizaje, buscando sus respectivos hiperparámetros óptimos.

Algo importante a comentar para este tipo de problemas es la cantidad de datos. En estos experimentos se puede conseguir una gran cuantía de dos maneras: utilizando muchas personas y una menor duración de la prueba o bien utilizando menos personas pero mayor duración de la prueba. En este proyecto, concretamente, se ha tenido que optar por la segunda elección debido a la situación pandémica que sufrimos actualmente. Aun así, es importante resaltar que una muestra como la de este trabajo, que consta de 26 personas y un conjunto de datos de 49000 instancias y otro de 15000, es aparentemente más que suficiente.

Cabe destacar que los vídeos son representativos porque en la encuesta, tras realizar la media de cada emoción, se ha obtenido que ninguno da un resultado por debajo de 3, es decir las puntuaciones rondan entre 3 y 5 (en una escala de 0 a 5).

Un proyecto muy ambicioso como futuro trabajo, sería intentar alcanzar una buena predicción con más emociones aparte de las básicas y del estado neutral. Para esto, probablemente se requiera un laboratorio profesional y más dispositivos. Es ésta una de las razones por las que la detección de emociones es un campo complejo, pues se puede llegar a un número de emociones demasiado grande y esto sería muy difícil de estudiar, ya que se trataría de intentar explicar o entender el funcionamiento del cerebro en el ámbito emocional. Asimismo, para un trabajo futuro sería interesante probar los modelos en entorno reales y poder probar el enfoque de usar menos instancias por persona y más sujetos (si la situación pandémica lo permitiese). También, en un futuro trabajo sería interesante probar *feature selection* y no solo eliminar características poco útiles manualmente como se ha hecho en este proyecto.

Bibliografía

- [1] S.J. Russell and Norvig. *Inteligencia artificial: un enfoque moderno*. 2004.
- [2] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligentsystems*. O'Reilly Media, 2017.
- [3] Miguel Sotaquirá. Las máquinas de soporte vectorial: una explicación completa. <https://www.codificandobits.com/blog/maquinas-de-soporte-vectorial/>, 2021.
- [4] Baeldung. Multiclass classification using support vector machines. <https://www.baeldung.com/cs/svm-multiclass-classification>, 2021.
- [5] chirag676. Importance of cross validation: Are evaluation metrics enough? <https://www.analyticsvidhya.com/blog/2021/05/importance-of-cross-validation-are-evaluation-metrics-enough/>, 2021.
- [6] Jennifer S. Lerner, Ye Li, Piercarlo Valdesolo, and Karim S. Kassam. Emotion and decision making. *Annual Review of Psychology*, (1):799–823, 2015.
- [7] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.
- [8] Stanisław Saganowski, Anna Dutkowiak, Adam Dziadek, Maciej Dzieżyc, Joanna Komoszyńska, Weronika Michalska, Adam Polak, Michał Ujma, and Przemysław Kazienko. Emotion recognition using wearables: A systematic literature review. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–6, 2020.
- [9] Han Wen Guo, Yu Shun Huang, Jen Chien Chien, and Jiann Shing Shieh. Short-term analysis of heart rate variability for emotion recognition via a wearable ecg device. In *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pages 262–265, 2015.

- [10] Feri Setiawan, Sunder Ali Khowaja, Aria Ghora Prabono, Bernardo Nugroho Yahya, and Seok-Lyong Lee. A framework for real time emotion recognition based on human ans using pervasive device. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, pages 805–806, 2018.
- [11] Maulik Kamdar and Michelle Wu. Prism: A data-driven platform for monitoring mental health. 2016.
- [12] Eiman Kanjo, Eman M.G. Younis, and Chee Siang Ang. Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection. *Information Fusion*, pages 46–56, 2019.
- [13] Huanghao Feng, Hosein M. Golshan, and Mohammad H. Mahoor. A wavelet-based approach to emotion classification using eda signals. *Expert Systems with Applications*, pages 77–86, 2018.
- [14] David Pollreisz and Nima TaheriNejad. A simple algorithm for emotion recognition, using physiological signals of a smart watch. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2353–2356, 2017.
- [15] Cheng He, Yun-jin Yao, and Xue-song Ye. An emotion recognition system based on physiological signals obtained by wearable sensors. In *Wearable Sensors and Robots*, pages 15–25, 2017.
- [16] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, page 400–408, 2018.
- [17] E4 wristband. <https://www.empatica.com/en-eu/research/e4/>.
- [18] Muse. <https://choosemuse.com/es/muse-2/>.
- [19] Michele Magno, Michael Pritz, Philipp Mayer, and Luca Benini. Deepemote: Towards multi-layer neural networks in a low power wearable multi-sensors bracelet. In *2017 7th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, pages 32–37, 2017.
- [20] J.A. Rincon, Angelo Costa, Paulo Novais, Vicente Julián, and Carlos Carrascosa. Using non-invasive wearables for detecting emotions with intelligent agents. 2016.
- [21] Indeed. ¿cuánto se gana en españa de ingeniero/a informático/a? <https://es.indeed.com/career/ingeniero-inform%C3%A1tico/salaries>, 2021.

- [22] Philip Schmidt, Attila Reiss, Robert Dürichen, and Kristof Van Laerhoven. Wearable-based affect recognition—a review. *Sensors*, 2019.
- [23] Morteza Zangeneh Soroush, Keivan Maghooli, Kamal Setarehdan, and Ali Motie Nasrabadi. A review on eeg signals based emotion recognition. *International Clinical Neuroscience Journal*, pages 118–129, 2017.
- [24] Tao Xu, Yun Zhou, Zi Wang, and Yixin Peng. Learning emotions eeg-based recognition and brain activity: A survey study on bci for intelligent tutoring system. *Procedia Computer Science*, pages 376–382, 2018.
- [25] Catherine Marechal, Dariusz Mikołajewski, Krzysztof Tyburek, Piotr Prokopowicz, Lamine Bougueroua, Corinne Ancourt, and Katarzyna Węgrzyn-Wolska. *Survey on AI-Based Multimodal Methods for Emotion Detection*, pages 307–324. 2019.
- [26] Maria Egger, Matthias Ley, and Sten Hanke. Emotion recognition from physiological signal analysis: A review. *Electronic Notes in Theoretical Computer Science*, pages 35–55, 2019.
- [27] Lin Shu, Jinyan Xie, Mingyue Yang, Ziyi Li, Zhenqi Li, Dan Liao, Xiangmin Xu, and Xinyi Yang. A review of emotion recognition using physiological signals. *Sensors*, 2018.
- [28] P. Ekman and W. Friesen. Facial action coding system: a technique for the measurement of facial movement. 1978.
- [29] Philip Schmidt, Attila Reiss, Robert Dürichen, and Kristof Van Laerhoven. Labelling affective states in the wild": Practical guidelines and lessons learned. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, page 654–659, 2018.
- [30] Robert Plutchik. *Emotions and life: Perspectives from psychology, biology, and evolution*. 2003.
- [31] Amani Albraikan, Basim Hafidh, and Abdulmotaleb El Saddik. iaware: A real-time emotional biofeedback system based on physiological signals. *IEEE Access*, pages 78780–78789, 2018.
- [32] Kanlaya Rattanyu, Michiko Ohkura, and Makoto Mizukawa. Emotion monitoring from physiological signals for service robots in the living space. pages 580–583, 2010.
- [33] Xi Lu, Xiaohang Liu, and Erik Stolterman Bergqvist. "it sounds like she is sad": Introducing a biosensing prototype that transforms emotions into real-time music and facilitates social interaction. page 1–6, 2019.

- [34] Long Hu, Jun Yang, Min Chen, Yongfeng Qian, and Joel J.P.C. Rodrigues. Scai-svsc: Smart clothing for effective interaction with a sustainable vital sign collection. *Future Generation Computer Systems*, pages 329–338, 2018.
- [35] Luz Fernández-Aguilar, Jorge Ricarte, Laura Ros, and Jose M. Latorre. Emotional differences in young and older adults: Films as mood induction procedure. *Frontiers in Psychology*, page 1110, 2018.
- [36] Yaling Deng, Lei Chang, Meng Yang, Meng Huo, and R. Zhou. Gender differences in emotional response: Inconsistency between experience and expressivity. *PLoS ONE*, 2016.
- [37] Angelo Costa, Jaime A. Rincon, Carlos Carrascosa, Vicente Julian, and Paulo Novais. Emotions detection on an ambient intelligent system using wearable devices. *Future Generation Computer Systems*, pages 479–489, 2019.
- [38] L. Fernández-Aguilar, A. Martínez-Rodrigo, J. Moncho-Bogani, A. Fernández-Caballero, and J. M. Latorre. Emotion detection in aging adults through continuous monitoring of electro-dermal activity and heart-rate variability. In *IWINAC*, 2019.
- [39] Bobo Zhao, Zhu Wang, Zhiwen Yu, and Bin Guo. Emotionsense: Emotion recognition based on wearable wristband. In *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 346–355, 2018.
- [40] Tianyuan Xu, Ruixiang Yin, Lin Shu, and Xiangmin Xu. Emotion recognition using frontal eeg in vr affective scenes. In *2019 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, pages 1–4, 2019.
- [41] Martin Ragot, Nicolas Martin, Sonia Em, Nico Pallamin, and Jean-Marc Diverrez. Emotion Recognition Using Physiological Signals: Laboratory vs. Wearable Sensors. In *Applied Human Factors and Ergonomics*, pages 813 – 22, 2017.
- [42] Rishabh Gupta, Mojtaba Khomami Abadi, Jesús Alejandro Cárdenes Cabré, Fabio Morreale, Tiago H. Falk, and Nicu Sebe. A quality adaptive multimodal affect recognition system for user-centric multimedia indexing. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, page 317–320, 2016.

- [43] Amir Jalilifard, Alfredo Guilherme da Silva, and Md. Kafiul Islam. Brain-tv connection: Toward establishing emotional connection with smart tvs. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 726–729, 2017.
- [44] Minh Dao, Duc Tien Dang Nguyen, Asem Kasem, and Hung Tran. Healthyclassroom - a proof-of-concept study for discovering students' daily moods and classroom emotions to enhance a learning-teaching process using heterogeneous sensors. pages 685–691, 01 2018.
- [45] Bahareh Nakisa, Mohammad Naim Rastgoo, Andry Rakotonirainy, Frederic Maire, and Vinod Chandran. Long short term memory hyperparameter optimization for a neural network based emotion recognition framework. *IEEE Access*, pages 49325–49338, 2018.
- [46] C. Lisetti and Fatma Nasoz. Categorizing autonomic nervous system (ans) emotional signals using bio-sensors for hri within the maui paradigm. *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 277–284, 2006.
- [47] Christine Lætitia Lisetti and Fatma Nasoz. Using noninvasive wearable computers to recognize human emotions from physiological signals. *EURASIP journal on advances in signal processing*, pages 1–16, 2004.
- [48] Kanlaya Rattanyu and Makoto Mizukawa. Emotion recognition using biological signal in intelligent space. In *Human-Computer Interaction. Towards Mobile and Intelligent Interaction Environments*, pages 586–592, 2011.
- [49] Grzegorz J. Nalepa, Krzysztof Kutt, Barbara Giżycka, Paweł Jemioło, and Szymon Bobek. Analysis and use of the emotional context with wearable devices for games and intelligent assistants. *Sensors*, 2019.
- [50] Yu Nie, Yang Wu, ZhongYao Yang, Guangzhi Sun, Yongjian Yang, and Xuanyi Hong. Emotional evaluation based on svm. In *Proceedings of the 2017 2nd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2017)*, pages 641–644, 2017/03.
- [51] Marco Maier, Daniel Elsner, Chadly Marouane, Meike Zehnle, and Christoph Fuchs. Deepflow: Detecting optimal user experience from physiological data using deep neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1415–1421, 2019.
- [52] Marina V. Sokolova, Antonio Fernández-Caballero, María T. López, Arturo Martínez-Rodrigo, Roberto Zangróniz, and José Manuel Pastor.

- A distributed architecture for multimodal emotion identification. In *Trends in Practical Applications of Agents, Multi-Agent Systems and Sustainability*, pages 125–132, 2015.
- [53] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele Univ.*, 2004.
- [54] T. Gilman, Razan Shaheen, Karin Nylocks, Danielle Halachoff, Jessica Chapman, Jessica Flynn, Lindsey Matt, and Karin Coifman. A film set for the elicitation of emotion in research: A comprehensive catalog derived from four decades of investigation. *Behavior Research Methods*, 2017.