industriales
etsii

Escuela Técnica Superior de Ingeniería Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Detección de emociones en personas mayores con posible deterioro cognitivo usando un sensor RGB-D y nuevas técnicas de Visión Artificial.

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Autor: **María Dolores Belchí Martínez**
Director: Nieves Pavón Pulido
Codirector: María Trinidad Herrero Ezquerro

Cartagena, 12 de diciembre de 2023.

Universidad Politécnica de Cartagena

Índice.

Índice de Figuras.

1. Introducción y objetivos.

Las emociones son reacciones psicofisiológicas que presentan los humanos ante distintos estímulos del entorno y, además, forman parte de la comunicación no verbal de las personas. Estas no solo se exteriorizan a través de las expresiones faciales, sino que también interviene la entonación de la voz, los gestos, la elección de las palabras, etc.

Es por ello que, lograr de forma precisa el reconocimiento de emociones mediante algoritmos es una tarea complicada, ya que, al emplear imágenes de personas, se reduce considerablemente la información proveniente de los gestos de la mano, el tono de la voz y, en general, el contexto que rodea la comunicación de una persona.

En este trabajo se crea un conjunto de datos a partir de imágenes a las que se le aplica la solución de MediaPipe para obtener los landmarks o puntos de referencia facial.
Seguidamente, se lleva a cabo, mediante la aplicación Classification Learner de Matlab, la extracción de las características más relevantes del rostro y el posterior entrenamiento de los distintos modelos de clasificación disponibles, para el reconocimiento de emociones.

El objetivo es analizar cómo afecta el uso de todas las características o la elección de las más relevantes, en el rendimiento del modelo de clasificación y determinar qué tipo de conjunto de datos es el que ofrece mejores resultados en esta tarea.

Para llevar a cabo este objetivo, se deben alcanzar los siguientes subobjetivos:

1) Creación de un "dataset" o conjunto de datos, con un número significativo de imágenes de personas distintas en cuanto a sexo biológico, edad o tipo de expresión facial, entre otros.
2) Proceso de ordenación de datos del "dataset" y clasificación manual previa según ciertas características cualitativas, por ejemplo, sexo o rango de edad.
3) Aplicación de técnicas de extracción de características.
4) Aplicación de modelos de clasificación.
5) Comparación de modelos de clasificación mediante la aplicación de métricas adecuadas.

1.1. Estructura de la memoria.

Esta memoria está compuesta por las siguientes secciones:

1. **Introducción y objetivos**. En el capítulo actual se exponen los objetivos del proyecto y una breve descripción de la memoria.
2. **Antecedentes**. Se presenta el inicio de la Visión Artificial y sus usos en distintas áreas, junto con el ejemplo del robot asistencial Ryan.
3. **Estado del arte**. Se describe el procedimiento general que se lleva a cabo para el análisis de imágenes en Visión Artificial y el reconocimiento de la región de interés (ROI). Además de hace una revisión de los datasets más utilizados en tareas de reconocimiento de expresiones faciales y el uso de los landmarks de MediaPipe en estudios recientes.
4. **Inteligencia Artificial**. En este apartado se describen los conceptos básicos de la Inteligencia Artificial, Machine Learning y Deep Learning, así como los distintos tipos de aprendizaje y el funcionamiento de las redes neuronales.
5. **Creación del Dataset**. Durante este capítulo de explica el procedimiento llevado a cabo para la creación del dataset empleado en este trabajo.

6. **Herramientas para la clasificación de características y entrenamiento de modelos**. Se exponen las herramientas que emplearemos durante la extracción de características y el entrenamiento de modelos de clasificación.
7. **Descripción del proceso de selección de características y entrenamiento de los modelos de clasificación**. Se detalla el procedimiento seguido para la obtención de las características más importantes y la selección de los modelos de clasificación.
8. **Resultados**. En esta sección se presentan los resultados obtenidos en las distintas pruebas realizadas y la comparación de las características relevantes según su clasificación por sexo biológico y edad.
9. **Conclusiones y trabajo futuro**. Se exponen las conclusiones obtenidas en función de los resultados y el trabajo futuro, junto con las posibles mejoras.

2. Antecedentes.

La visión artificial o visión por computador es la disciplina que tiene como objetivo el procesamiento y análisis de imágenes que capturan el mundo real para que un ordenador sea capaz de comprender y tratar dichas fotografías como lo hacemos los humanos.

Este concepto nació en los años 60 y ha estado ligado a la evolución de las cámaras de fotografía y su posterior implicación en el ámbito científico.
Es en esta década, cuando se dedujo que el procesamiento digital de imágenes empezaba con formas sencillas y básicas, como lo bordes. Evolucionando, desde conseguir escanear fotografías hasta transformar las imágenes de su forma bidimensional a tridimensional.

Seguidamente, se obtuvo un gran progreso en los años 80, gracias al desarrollo de la ingeniería informática y la producción de microprocesadores más avanzados. Naciendo de esta forma, el estudio de la aplicación de la visión artificial y su implicación con la inteligencia artificial (Porta, 2020).

Tras estos avances, los campos de estudio de la visión artificial se han ampliado significativamente, por lo que podemos emplearla en numerosas aplicaciones y en diferentes contextos: industria, medicina, seguridad y robótica, entre otras.
Algunos ejemplos de aplicaciones son: diagnóstico de enfermedades, control de calidad y detección de defecto, desarrollo de vehículos autónomos, etc.

2.1. Ryan. Robot Asistencial Social.

El avance en la capacidad de procesamiento de datos de la informática ha incentivado a los investigadores a desarrollar aplicaciones muy útiles para las personas.

En particular, se ha abierto un amplio abanico de posibilidades en el ámbito sanitario, donde la implementación de la visión artificial ofrece gran ayuda en el análisis de imágenes médicas, como resonancias magnéticas, tomografía computarizada, endoscopias o rayos X, suponiendo un gran apoyo para los profesionales en labores como el diseño y seguimiento de tratamientos o detección de enfermedades.

Estas imágenes son complicadas de analizar por lo que solo pueden ser interpretadas por profesionales capacitados para ello, no obstante, podemos encontrar distintos factores que afecten al rendimiento de los médicos, como el agotamiento o la propia complejidad de la imagen, que disminuya la precisión del diagnóstico. Por ello, los sistemas de visión artificial ayudan a detectar patrones que son difíciles de hallar para los profesionales, consiguiendo así, diagnósticos más precisos.
Cabe destacar que estas herramientas son empleadas para asistir al personal médico y no para su reemplazo (Grisales, 2022).

Por otra parte, dentro del ámbito de la medicina encontramos el área de salud mental, que es, actualmente, uno de los principales focos de preocupación para la sociedad. Y es que, tras el uso excesivo de dispositivos móviles y redes sociales, en consonancia con la tendencia de aislamiento social presente hoy en día, se han disparado los casos de depresión y otras enfermedades mentales.

Otro factor muy importante es la reciente pandemia de COVID-19, que ha supuesto un gran impacto en la sociedad, aumentando considerablemente los diagnósticos de estas enfermedades.

En este contexto, es importante resaltar que el problema del aislamiento social no se limita únicamente a la generación más joven, sino que afecta de manera significativa a las personas mayores y, además, esta situación se ha visto agravada durante la pandemia de COVID-19, puesto que las restricciones de movilidad y medidas de distanciamiento se sumaron al ya existente panorama de aislamiento que sufren nuestros mayores debido a factores como la jubilación, soledad no deseada o dependencia.

Esta situación puede conllevar serios problemas de salud física, mental y emocional. Y es, precisamente, en este último punto donde los recientes avances de la visión e inteligencia artificial pueden sernos de gran utilidad para desarrollar aplicaciones que sirvan de ayuda a las personas mayores (Bequir, 2022).

Por ello, vamos a destacar el trabajo de *DreamFace Technology*, con sede en Denver (Colorado) que han creado un robot asistencial, llamado Ryan, equipado con inteligencia artificial para proporcionar apoyo y compañía a personas mayores que sufren depresión y/o demencia.

Ryan se caracteriza por ser un robot de asistencia social, cuyo objetivo es el de proporcionar compañía a las personas. Este tipo de robots forman parte de una rama de la robótica denominada Robótica de Asistencia Social (SAR), que se encarga de desarrollar sistemas dotados de inteligencia artificial, los cuales están capacitados para comunicarse con los usuarios de forma coherente. Además, estos robots están compuestos por sensores que les permite percibir y procesar el entorno con el fin de comportarse según sea necesario en cada instante. (Pareto Boada, 2022)

Centrándonos en el robot creado por DreamFace Technology, Ryan tiene un aspecto humanoide, capaz de mostrar un rostro expresivo en 3D gracias a la técnica de proyección trasera o retroproyección, mostrando así las expresiones faciales animadas. Además, tiene situada en la cabeza una cámara RGB con la que recopila imágenes para su algoritmo de reconocimiento de rostros. Este robot también dispone de un cuello con dos grados de libertad para poder seguir al usuario y mantener así, el contacto visual.
Por otra parte, en el torso se encuentra una pantalla de 10 pulgadas, interactiva para los usuarios, con la que puede reproducir contenido multimedia. En esta misma zona también se sitúa una cámara RGB-D Kinect, junto con otros elementos como altavoces, micrófono y, por supuesto, el resto de los componentes de E/S, cálculo y la alimentación del sistema (ver Figura 1).

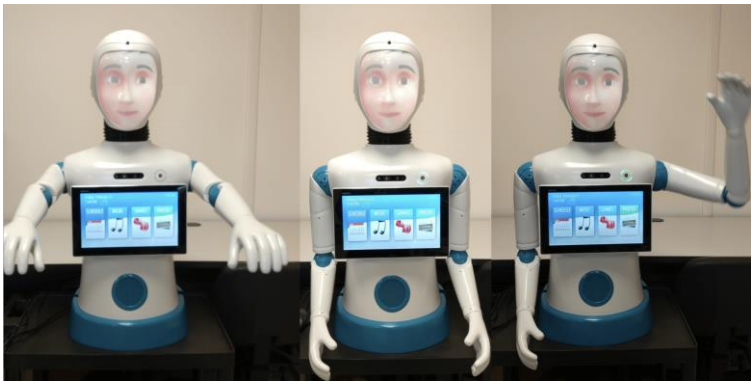

Figura 1. Ryan. Robot asistencial de DreamFace Technology.

En cuanto a su funcionamiento, es capaz de reconocer las expresiones faciales para su posterior procesamiento, consiguiendo así, simular y reaccionar a las emociones de los pacientes. Para ello, consta de un algoritmo de reconocimiento emocional multimodal, es decir, emplea el Análisis de Sentimiento (SA) y el Reconocimiento de Expresiones Faciales (FER), mejorando de esta manera, la precisión para interpretar las emociones de los usuarios.
Para el reconocimiento facial de expresiones, emplea el algoritmo de reconocimiento facial de Viola-Jones, que toma como entrada una imagen recortada del rostro de la persona que, junto con una red neuronal profunda, da como resultado un número que representa las tres clasificaciones de las emociones según su grado, es decir, neutro (0), positivo (+1) o negativo (-1) (Abdollahi, Mahoor, Zandie, Siewierski, & & Qualls, 2023).
Y en cuanto al análisis de emociones, utiliza la herramienta de procesamiento le lenguaje natural CoreNLP, desarrollada en Standford, que dispone de un etiquetador de sentimientos. Esta utilidad se basa en el algoritmo RNTN (Red de Tensor Neural Recursivo) que toma como entrada una oración y analiza el sentimiento de ésta a partir de la combinación de las palabras (López Barbosa, 2015).

Las interacciones Humano – Robot se llevan a cabo mediante conversaciones con los usuarios, por lo que realizaron una recopilación de distintos temas de diálogos. El robot es capaz de empatizar con los usuarios, durante las conversaciones, creando así un ambiente más agradable y consiguiendo una mejor aceptación por parte de las personas.
Por ejemplo, si al hablar de un tema en particular se detecta una expresión positiva en el usuario, Ryan ofrecerá una respuesta acorde a ese sentimiento. Esto es posible gracias a la implementación del reconocimiento emocional multimodal que se lleva a cabo en el transcurso de la charla.

En el estudio *Artificial Emotional Intelligence in Socially Assistive Robots for Older Adults* (2022) (Abdollahi, Mahoor, Zandie, Siewierski, & & Qualls, 2023), se puso a prueba la eficacia de este robot a la hora de empatizar con las personas mayores de una residencia. Para ellos se crearon 4 grupos de participantes que interactuarían con dos variantes de Ryan: el empático y el no empático.
Cuando la versión empática conversaba con los usuarios, estaba activado el reconocimiento de las expresiones faciales y el análisis de emociones, lo que le permitía a Ryan reflejar las emociones de los participantes durante el diálogo. Por el contrario, el Ryan no empático no disponía de la función de reconocimiento emocional multimodal, por lo que no contaba con expresión facial ni detección de sentimientos al hablar.
Para el análisis de este estudio se tuvo en cuenta el cómputo de palabras por parte de los participantes en respuesta al robot y el número de expresiones faciales recogidas durante la conversación.

Como resultado, se obtuvo que, durante la interacción Humano – Robot, con la versión de Ryan empático, hubo un mayor recuento de palabras, lo que sugiere una mayor implicación e interés por parte de los usuarios en dicha interacción, de igual forma, la recopilación de expresiones positivas fue mayor. Por lo tanto, el empleo de la visión e inteligencia artificial en este tipo de aplicaciones puede ayudar a las personas mayores a mejorar su bienestar e intentar disminuir el sentimiento de soledad y aislamiento.

Por otra parte, el robot presenta ciertas limitaciones que debemos considerar, como el procesamiento de otras entradas de datos, es decir, se podrían incluir el análisis de la postura

del usuario, el movimiento de manos, los sonidos e incluso el movimiento de ojos. Esto proporcionaría más información para lograr que este sistema fuese más preciso en la tarea de reconocimiento. Adicionalmente, sería necesario extender el contenido de los temas a tratar en los diálogos y, además, corregir las interrupciones de Ryan durante las conversaciones debido al ritmo lento y las largas pausas de los participantes al hablar (Abdollahi, Mahoor, Zandie, Siewierski, & & Qualls, 2023).

3. Estado del arte.

En este apartado se va a realizar una revisión de la literatura con el objetivo de comprender los avances en el ámbito del reconocimiento facial, así como las técnicas y metodologías más recientes.

3.1. Proceso general de cómputo.

Generalmente, el procedimiento que se lleva a cabo durante una tarea de visión por computador es el siguiente (ver Figura 2):

1. **Adquisición de imágenes**: Se toma como entrada imágenes o videos captados por una cámara digital o sensor.
2. **Preparación de imágenes**: durante este paso se lleva a cabo el "preprocesamiento" ya que se actúa sobre la imagen para mejorar su calidad, su condición lumínica, reducción de ruido y, en general, eliminar aquella información que no es relevante. Además, también se puede transformar la imagen a escala de grises para disminuir la exigencia de cómputo informático.
3. **Extracción de características**. Etapa donde se selecciona las características, como formas o patrones, más significativas de la imagen con el fin de discretizarlas.
4. **Clasificación**. Se lleva a cabo el análisis de las características para ofrecer una clasificación de estas según su grado de similitud.
5. **Resultados y clasificación**. En este paso se valora la precisión del modelo empleado para determinar si su uso es adecuado para la tarea a realizar.

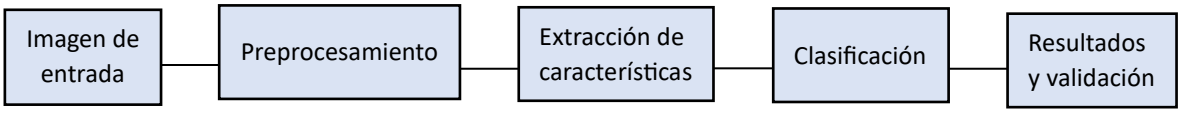

Figura 2. Diagrama de flujo de cómputo general

Esto pasos pueden variar según el tipo de modelo que utilicemos y los requerimientos necesarios (Zago Canal, y otros, 2021).

3.2. Reconocimiento Facial.

Para el reconocimiento de emociones en personas, la fase de detección o reconocimiento facial es crucial para el correcto funcionamiento de un modelo de detección. Es por ello que se debe escoger lo que se denomina "Región de Interés" (ROI), que engloba el rostro de la persona en la imagen de entrada (Zago Canal, y otros, 2021).

Esta tarea se encuentra dentro de la etapa de preprocesamiento ya que la imagen inicial contine información poco relevante como el fondo de la imagen, la pose de la persona o la iluminación. Esta información se elimina mediante técnicas como el recorte de la ROI o la conversión de la imagen a escala de grises, consiguiendo así, un menor coste computacional en el reconocimiento facial.

Algunos métodos que se emplean para el reconocimiento de la región de interés son:

El Algoritmo 'Viola-Jones', desarrollado por Paul Viola y Michael J. Jones en 2003, que se basa en los clasificadores denominados *Haar-like features*. Estos clasificadores son atributos básicos que se buscan en las imágenes con el objetivo de hallar intensidades luminosas dispares entre regiones adyacentes de la imagen, de manera que se lleva a cabo la suma y resta de pixeles blancos y grises que contienen estas áreas.
Una vez completado este proceso, se aplica un clasificador basado en AdaBoost, en el que los *Haar-like features* se disponen en cascada para seleccionar aquellos que son relevantes y descartar lo que no lo son (Parra Barrero, 2015).

El uso de Redes Neuronales Convolucionales (CNN), como la arquitectura MTCNN, Red Convolucional Multitarea en Cascada, ofrecen un alto rendimiento frente a otro tipo de algoritmos. Estas redes toman las imágenes de entrada, procesándolas como matrices de pixeles, a las cuales se le aplicarán una serie de funciones y filtros con el objetivo de clasificar los objetos de la imagen y asignarles una probabilidad en 0 y 1. De esta forma, se entrenan las redes neuronales para ser capaces de detectar los objetos en imágenes que no ha "visto" anteriormente.

3.3. Datasets.

Un conjunto de datos, o dataset, es una colección estructurada de información, que puede ser de distintos tipos de datos, como imágenes, videos, números o texto. Se emplean en investigación, para estudios estadísticos, análisis de patrones y entrenamiento de modelos de aprendizaje automático, entre otros ejemplos (Rodríguez, 2023).

En este caso nos centraremos en los conjuntos de imágenes (ver Figura 3), ya que desempeñan un papel fundamental en el entrenamiento de algoritmos de reconocimiento facial. En estos Datasets, todas las imágenes están etiquetadas por categorías, por ejemplo, edad, género o tipo de expresiones faciales para representar emociones.

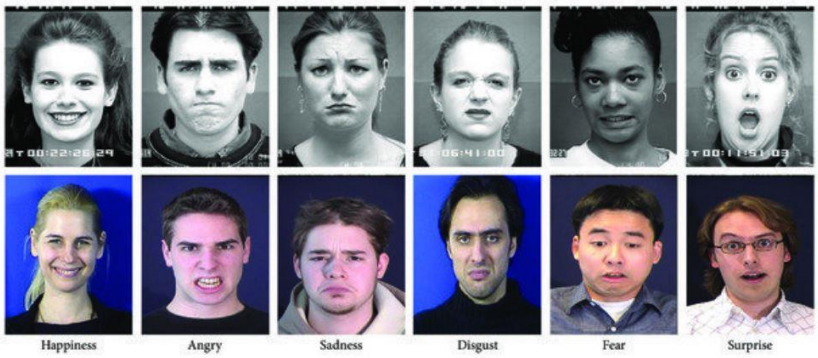

Figura 3. Imágenes de muestra del conjunto de datos CK+ y del conjunto de datos MMI.

Ejemplos de dataset empleados en el reconocimiento facial son:

- Cohn-Kanade (CK), publicado en el año 2000 y creado por un equipo de investigación de la Universidad de Pittsburgh con el objetivo de impulsar el estudio de reconocimiento de expresiones faciales. (Zago Canal, y otros, 2021)

Esta primera versión está formada por 486 imágenes en las que los participantes muestran distintas expresiones solicitadas por los investigadores, de esta forma, las expresiones faciales no surgen de forma espontánea en los participantes, por lo que puede presentar variaciones entre las expresiones capturadas y la reales que se manifiestan de manera natural.
El tamaño de estas muestras es de 640x480 pixeles y se representan en escala de grises. Además, las imágenes se tomaron en un entorno donde la iluminación estaba controlada, así como, el fondo y la disposición de la cámara frente al participante.

Posteriormente, en 2010, se publicó una nueva versión de este dataset denominado 'CK+', donde se incluyeron 107 muestras de expresiones faciales, con la diferencia de que, en esta segunda versión, los participantes no posaron para mostrar una emoción, sino que se obtuvo la expresión de forma natural.
Estas imágenes se categorizaron en 7 emociones distintas: felicidad, ira, tristeza, miedo, sorpresa, disgusto y desprecio.

- FER2013. Este dataset fue creado por Pierre-Luc Carrier y Aaron Courville, en 2013, y está compuesto por 35.887 imágenes, etiquetadas en 7 emociones, al igual que en el dataset CK+ (felicidad, ira, tristeza, miedo, sorpresa, disgusto y desprecio). Este conjunto de imágenes se creó con el objetivo de entrenar y validar algoritmos de clasificación, por lo que el tamaño de las imágenes es de 48x48 pixeles, en escala de grises, lo que reduce el coste computacional de esta tarea.

- MMI Facial Expression Database, se creó en el año 2002 para suplir la falta de bases de datos con muestras de comportamiento y afecto necesarios para el progreso de la investigación en el campo del 'Análisis Automático del Comportamiento Humano' y la creación y validación de modelos de reconocimiento de expresiones faciales.
  Este dataset está formado por más de 2900 videos e imágenes, con una resolución 720x576, de 75 participantes de diferentes etnias, que representan 6 emociones básicas. En este caso, las muestras fueron tomadas mientras los participantes visualizaban una serie de videos diseñados para generar reacciones naturales. (Michel F. Valstar, s.f.)

- JAFFE, (Japanese Female Facial Expression) fue publicado en 1998 por un equipo de investigadores del departamento de Psicología de la universidad Kyushu. Se tomaron imágenes de 10 mujeres japonesas que expresan: felicidad, disgusto, ira, sorpresa, tristeza, miedo y neutralidad, formando así, un dataset de 213 imágenes de 256x256 píxeles. (Michael J, Kamachi, & Gyoba, s.f.)

Unos de los aspectos más importantes para conseguir resultados precisos en el reconocimiento facial es la obtención de un conjunto de datos que contenga un gran número muestras, de forma que sea lo más representativo y diverso posible con todas las categorías que queremos detectar.

Por otra parte, los datos de los conjuntos se dividen, generalmente, en dos secciones, la primera formará parte de 'Entrenamiento' y la segunda para 'Validación' o 'Test'. Esta división se lleva a cabo para lograr que los algoritmos de clasificación aprendan los patrones que se repiten en las imágenes, según en la categoría que se encuentre cada una.
Para esta tarea se emplea el conjunto de Entrenamiento, que contendrá aproximadamente el 80% de los datos y, una vez entrenado el algoritmo, se validará la precisión de los resultados con

el conjunto Test, sobre el 20% restante de los datos, de manera que se emplearán las imágenes que el algoritmo no ha "visto" anteriormente.

Además, también se debe tener en cuenta el proceso de obtención de las imágenes, que implica detalles como el tipo de cámara empleada o las condiciones del entorno. Aquí entran en juego variable relevantes como la condición lumínica o el fondo donde se toma la muestra.

Tras los análisis llevados a cabo en el artículo: *A survey on facial emotion recognition techniques: A state-of-the-art literature review*, se evidencia el peso que tiene la diversidad en los conjuntos de datos, puesto que se tomó distintos Datasets para comparar los resultados que se obtiene al ser empleados para el reconocimiento facial mediante dos técnicas distintas: los métodos clásicos de visión artificial y redes neuronales.

Los Datasets más extendidos, como el JAFFE, son los que presentan más limitaciones en cuanto a resolución, bajo número de muestras y poca variabilidad en cuanto a las nacionalidades de los participantes, esto conlleva la aparición de sesgo en los datos del conjunto y una baja tasa se acierto, ya que solo darían buenos resultados en casos muy concretos con elevada uniformidad en las muestras.

Podemos describir el sesgo como un error sistémico en el cual los datos empleados para entrenar o validar un modelo conducen a resultados que difieren de su valor esperado, por tanto, se obtendrían resultados poco precisos y con bajo rendimiento.

Por otra parte, también se encuentra una diferencia en los conjuntos de datos, como el CK+, que contienen expresiones forzadas o posadas respecto a los que tienen expresiones naturales, como el MMI Facial Expression Database, creando una mayor dificultad en reconocer las expresiones de forma precisa, cuando estas no son obtenidas de forma genuina. Resultando más complicado interpretar el verdadero significado de la expresión del participante. (Zago Canal, y otros, 2021)

No obstante, sería conveniente añadir al dataset imágenes de expresiones de los dos tipos, natural y posada, con el fin de mejorar la precisión global de los algoritmos de clasificación.

3.4. Comparación de técnicas de visión artificial clásicas y métodos basados en redes neuronales.

El enfoque clásico de la visión por computador hace referencia a todos aquellos métodos tradicionales de procesamiento y análisis de imágenes que han sido fundamentales en el ámbito de la visión por computador, ya que muchas de estas técnicas han conseguido grandes resultados por lo que son ampliamente utilizados actualmente.

Estas técnicas basadas en modelos matemáticos se caracterizan por ser métodos de procesamiento digital de imágenes, reconocimiento de patrones, aplicación de filtros o recortes para destacar únicamente la región de interés en imagen; transformaciones morfológicas; técnicas de umbralización y detección de bordes, entre otros. (Domínguez, 2022)

Si nos centramos en el campo de reconocimiento facial, podemos describir el proceso general de cómputo como el mencionado en la sección 3.1, explicado anteriormente, donde después de tomar la imagen de entrada y aplicarle las técnicas de preprocesamiento, se lleva a cabo a extracción de características y su clasificación, es decir, la detección facial en la imagen, llevado

a cabo por un detector capaz de interpretar los vectores de características extraídas y la posterior clasificación en correspondencia al número de clases de salida.

Entre los algoritmos de clasificación más reconocidos encontramos el Support Vector Machine (SVM) o Máquina de Soporte Vectorial, el algoritmo de Naïve Bayes o KNN (K-nearest neighbors).

Por otra parte, tenemos los enfoques basados en Redes Neuronales, lo cuales han ido creciendo y desarrollándose a lo largo del tiempo gracias a la investigación y evolución computacional. Estas redes se inspiran en el funcionamiento del cerebro humano, concretamente en las neuronas biológicas, con la finalidad de aprender a reconocer patrones de los datos de entrada y lograr optimizar su precisión, adaptándose a través de la experiencia. (Quintal, 2023)

En el estudio *A survey on facial emotion recognition techniques* (Zago Canal, y otros, 2021), se detalla la comparación entre estas dos técnicas de trabajo, obteniendo las siguientes conclusiones: los conjuntos de datos o datasets empleados para el entrenamiento de cada método es una variable que repercute en la precisión del resultado, puesto que, aquellos datasets que sean más limitados en cuanto a la variabilidad de los datos, serán más precisos pero solo serán útiles en ese ámbito concreto y estará más restringido. Por tanto, el rendimiento del método disminuirá si se aplica en entornos no tan acotados.
Adicionalmente, esta variación de rendimiento también se observa según el método empleado, puesto que, los algoritmos de clasificación clásicos son adecuados para trabajar con información específica y, a su vez, pueden reducir su rendimiento cuando se trata de información o datos más generales.
Por el contrario, las técnicas que emplean redes neuronales ofrecen resultados más precisos cuando se maneja información de carácter general, es decir, con más diversidad, gracias a su capacidad de abstracción (Zago Canal, y otros, 2021).

Por lo tanto, en el área de reconocimiento de emociones, las redes neuronales presentan la capacidad de generalizar para detectar y clasificar las emociones con conjuntos más extensos, lo que posibilita su aplicación en entornos no tan restrictivos en comparación con los algoritmos clásicos.

3.5. Reconocimiento de emociones con landmarks de MediaPipe.

En el ámbito del reconocimiento de emociones, el uso de MediaPipe como herramienta abre otros campos de estudio. Lo que permite este marco de trabajo es la obtención de puntos de referencia, denominados 'Landmarks', del rostro mediante el uso de una malla sobre este.

Estos landmarks ofrecen datos específicos de la disposición del rostro, por lo que podemos emplearlos para analizar las distintas expresiones y obtener una visión más detallada sobre cómo se manifiestan las emociones en el rostro.

Un ejemplo del uso de MediaPipe en esta área es el reciente estudio "*Emotion recognition at a distance: The robustness of machine learning based on hand-crafted facial features vs deep learning models*" (Bisogni, Cimmino, Marsico, Hao, & Narducci, 2023), donde se realiza la comparación de las técnicas Machine Learning y Deep Learning, en la clasificación de siete emociones distintas.
En él se tomaron dos modelos, uno de Deep Learning para extracción de caracteristicas relevantes y su posterior clasificación, basados en técnicas repartidas en el tiempo junto con una

configuración LSTM. El segundo método emplea MediaPipe para la extracción de las caracteristicas más importantes de las imágenes y un Support Machine Vector para la posterior clasificación de las emociones.

Tras el experimento, se obtuvo que el modelo que empleaba los landmarks de MediaPipe conseguir una mejor detección de las expresiones faciales de las imágenes, cabe destacar que las muestras empleadas pertenecen al Dataset Cohn-Kanade (CK+), a las que se le aplicaron un preprocesamiento de identificación de la región de interés (ROI) y una disminución del tamaño de la imagen a 58x58.

La primera técnica, basada en DL, obtiene un peor rendimiento con el uso de imágenes tan pequeñas, mientras que el segundo método no se ve afectado por la condición de las muestras. Siendo MediaPipe capaz de calcular correctamente todos los landmarks de los rostros. Esto implica que se debe tener en cuenta el tipo de conjunto de dato que se tiene para seleccionar un modelo de clasificación u otro.

Otro ejemplo del uso de MediaPipe es en "*Masked Face Emotion Recognition Based on Facial Landmarks and Deep Learning Approaches for Visually Impaired People*" (Mukhiddinov, Djuraev, Akhmedov, Mukhamadiyev, & Cho, 2023) en el que se llevó a cabo el entrenamiento de un modelo de aprendizaje profundo para el reconocimiento de emociones en personas con el rostro cubierto por una máscara o mascarilla, empleando la solución de MediaPipe para la obtención de las características más relevantes, que en este caso serían los ojos y cejas.

4. Inteligencia Artificial.

La inteligencia artificial es aquella condición que disponen algunos sistemas de imitar el razonamiento humano. Esto es posible gracias al desarrollo de algoritmos que, a partir de una entrada de datos, llevan a cabo el proceso de clasificación y categorización de información para ofrecer resultados, en base a patrones específicos que han aprendido. De forma que aplican estas pautas definidas para ofrecer soluciones a problemas concretos.
Esta habilidad para adquirir conocimiento y amoldarse en función de la información proporcionada es esencial en el progreso de la inteligencia artificial, permitiendo de esta forma, que los sistemas aumenten su autonomía y precisión en sus respuestas y acciones.

4.1. Machine Learning.

El aprendizaje automático, o en inglés Machine Learnig, es la rama de la inteligencia artificial que dota a los sistemas la habilidad de aprender por ellos mismos. Toman como entrada información clasificada por conjuntos de características para comprender las diferencias y similitudes de estas, de este modo, el sistema consigue actuar en consecuencia a lo aprendido cuando obtiene nueva información.
Esta herramienta requiere etapas de entrenamiento de datos clasificados y etiquetados manualmente por un experto y la validación humana para corregir errores y ofrecer soluciones o predicciones más rigurosas.

En la programación o programación manual, el programa es desarrollado por el experto, llevando a cabo una serie de tareas como el diseño de la lógica de dicho programa y la definición de las reglas a seguir, de forma que ante los datos que el programa toma como entrada, obtendremos una salida deseada acorde a ello.
Por tanto, la diferencia presente entre la programación manual y el Machine Learning es que, en el primer caso, es el experto el que debe definir la lógica del programa, mientras que, en el segundo, ante las entradas de las que partimos y las salidas marcadas como objetivo, es el algoritmo el que se encarga de desarrollar la lógica y las reglas a seguir para cumplir con dicha tarea. (Gonzalez L. , 2020)

Un ejemplo básico de aprendizaje automático y como un sistema es capaz de aprender los patrones de ciertos datos, es el filtro de Spam de correo electrónico, que emplea estos tipos de algoritmos para lograr identificar que correos se consideran “basura” y cuales no, de forma que los primeros se separan del resto. De esta forma, una vez aprendidas las pautas, estos pueden tomar decisiones. (Rouhiainen, 2018)

Tipos de Aprendizaje Automático.

En el campo del aprendizaje automático, encontramos diferentes formas en las que los sistemas adquieren y emplean la información, por ello, se presentan 3 tipos principales de aprendizaje automático.

- Aprendizaje Supervisado: Esta categoría necesita la intervención humana para clasificar y categorizar los datos que servirán de entrenamiento para el algoritmo, de este modo, se indica al sistema cual debería ser la salida o respuesta ante determinada información

(ver Figura 4). El algoritmo se encarga de hacer predicciones sobre los conjuntos de datos a la vez que recibe retroalimentación de un usuario, en caso de que la respuesta final no sea la adecuada. Es durante este proceso, que el algoritmo aprende los patrones necesarios (Rouhiainen, 2018).

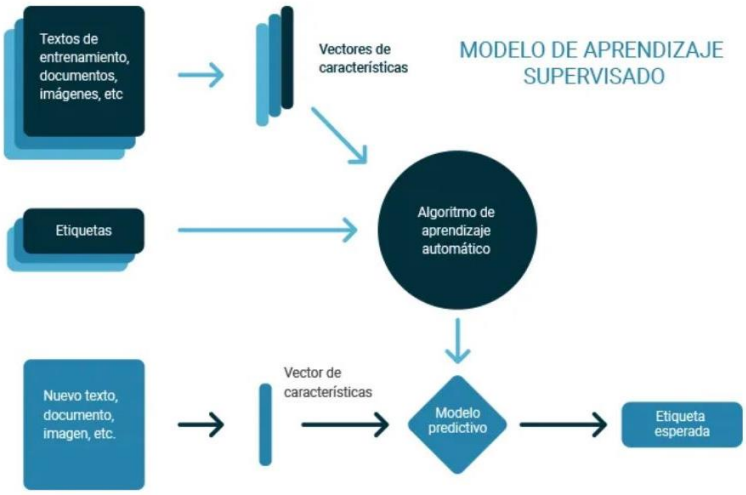

Figura 4. Diagrama de flujo de aprendizaje supervisado

- Aprendizaje No Supervisado: Este tipo de aprendizaje se distingue del supervisado en que no requiere de la clasificación y categorización previa de los datos por parte de un usuario, por lo que este no es un requisito de entrenamiento del modelo (ver Figura 5). Por lo tanto, es el propio algoritmo el que se encarga de clasificar la información, esto lo lleva a cabo mediante la división de los datos en grupos que comparten características similares (Gonzalez J. L., 2020).

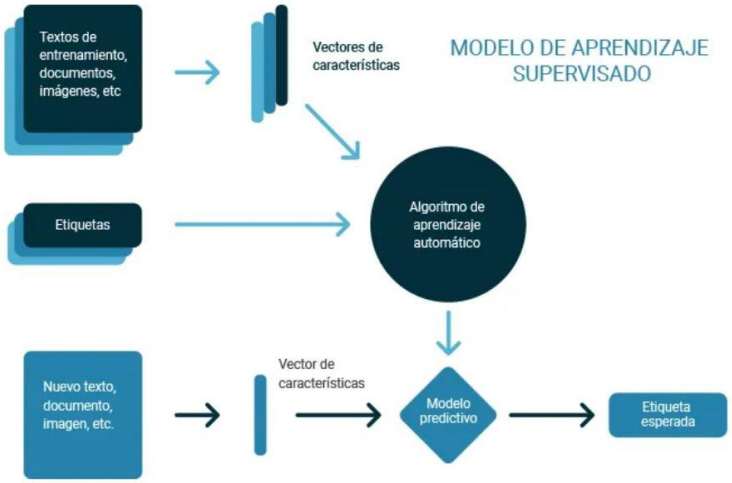

Figura 5. Diagrama de flujo de aprendizaje no supervisado

- Aprendizaje Por Refuerzo: En el aprendizaje por refuerzo, tampoco se requiere de datos clasificados para el entrenamiento, sino que se aprende mediante la experiencia, es decir, ante unas condiciones dadas, el algoritmo se encarga de llevar a cabo un proceso repetitivo con el fin de obtener "recompensas".
  La meta es que el modelo aprenda a tomar decisiones que aumente sus recompensas a lo largo del tiempo, adaptando su comportamiento dependiendo del tipo de retroalimentación que reciba (ver Figura 6).
  Este modelo es el que más se asemeja a la psicología conductista de las personas, puesto que reproduce el sistema acción-recompensa (Gonzalez J. L., 2020).

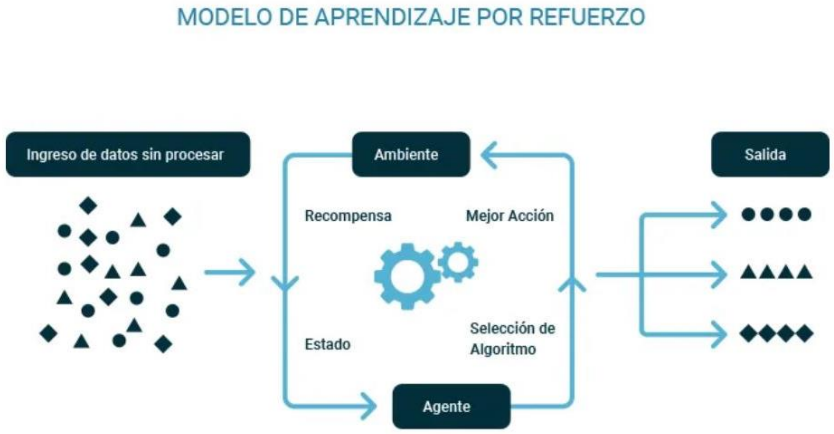

Figura 6. Diagrama de flujo de aprendizaje por refuerzo

4.2. Deep Learning.

El aprendizaje profundo, o Deep Learning, es un subcampo dentro del Machine Learning que emplea redes neuronales con el objetivo de afrontar desafíos que serían muy difíciles de resolver con los métodos tradicionales debido a la complejidad de los datos de entrada (ver Figura 7).
El término "profundo" hace referencia a la estructura interna de las redes neuronales, que están compuestas por numerosas capas, esto lo explicaremos más adelante (Ríos, 2022).

En el aprendizaje profundo se consigue procesar la información de forma análoga al funcionamiento del cerebro humano de forma que, si la red neuronal dispone de los datos suficientes, será capaz de aprender por sí misma.
Esta es la diferencia entre Machine Learning y Deep Learning, mientras que en el primer caso es necesario que haya una intervención humana para clasificar y "enseñar" las características de un elemento para que se lleve a cabo el aprendizaje por parte del sistema y sea capaz de reconocer esos patrones, en el Deep Learning, el sistema adquiere el conocimiento a través de los datos sin procesar y, mientras más datos le sean suministrados, mayor será la capacidad predictiva del sistema. (EKCIT, 2021)
Es por ello que, para el buen funcionamiento de los sistemas de Deep Learning, se precisa de una mayor potencia computacional, además de un conjunto de datos de entrenamiento completo.

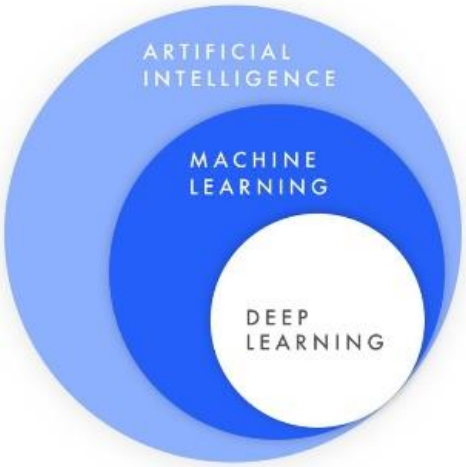

Figura 7. Relación IA, Machine Learning, Deep Learning

Por otra parte, los resultados obtenidos son más precisos, por lo que pueden empleados en aplicaciones como en los servicios financieros para el desarrollo de medidas de prevención, es decir, se emplean técnicas que se basan en la identificación de patrones en las transacciones de los clientes de forma que se pueda detectar y reducir el número de operaciones fraudulentas.

Otra implementación en desarrollo es la conducción autónoma de los vehículos. Con el uso de cámaras y sensores que capten la información del entorno, el vehículo con sistema de Deep Learning, interpretaría los datos con el fin de tomar decisiones ante cualquier situación.
Actualmente, empresas como Tesla, Waymo, Mobileye y Nvidia investigan en cómo mejorar la conducción autónoma de los vehículos invirtiendo recursos en este propósito (Rodriguez, 2021).

4.3. Redes Neuronales.

Como hemos comentado anteriormente, en el aprendizaje de sistemas o creación de modelos, se emplean redes neuronales artificiales (RNA), estas redes pretenden simular el funcionamiento de las neuronas del cerebro humano con el fin de lograr que un sistema tenga la capacidad de aprender de la misma forma que lo hacemos las personas.
Las RNA son un conjunto de unidades de procesamiento que ofrecen un resultado mediante la transformación de los datos de entrada.

Para explicar la base de las Redes Neuronales, tomaremos el Perceptrón como ejemplo.
Se denomina Perceptrón (ver Figura 8), a la unidad básica de red neuronal desarrollado por Frank Rosenbelt en 1957. Este primer modelo estaba formado por una única capa de neuronas y principalmente se ocupaba de la clasificación de patrones a partir de la información de entrada, los datos se introducían en la red en forma de vectores bipolares o escalas binarias, es decir, ‘0’ y ‘1’ (Sarmiento-Ramos, 2020).

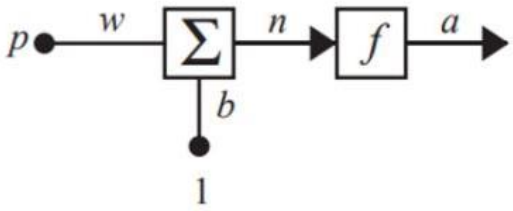

Figura 8. Estructura del Perceptrón de una entrada.

La función que define a la neurona es:

$$a = f(wp + b)$$

Donde *p* se corresponde con la entrada; *w* es el peso que se le asigna a dicha entrada y representa y un término independiente, *b*, que se denomina bias.
Seguidamente, se realiza una suma ponderada entre estos valores y, en función del resultado, el peso *w* se ajustará para conseguir minimizar el error a la salida.

No obstante, el inconveniente que presenta el perceptrón es que solo es aplicable para problemas de clasificación linealmente separables, por lo que, para ser aplicado en todo tipo de problemas, se introdujo las redes neuronales multicapa (Sarmiento-Ramos, 2020).

4.3.1. Elementos fundamentales de una Red Neuronal.

Vamos a describir los elementos básicos que componen una red neuronal (ver Figura 9), para su correcto funcionamiento:

- Función de entrada. Este es el primer paso en el proceso transformación de información en la neurona. La red se alimenta de los datos de entrada y se le asigna un valor que denominamos 'pesos'.
  Los pesos hacen referencia a la ponderación que se le asigna a la entrada de cada neurona y afectarán a la salida de la red, puesto que cuanto mayor sea el valor, más relevancia tomará dicha entrada, y de igual forma ocurrirá en caso contrario.

  Por tanto, la función de entrada se establece como la suma ponderada de la entrada y su peso asignado (Lozano, 2020).

- Función de Activación. Esta función se ocupa de establecer el estado de activación de una neurona según la suma ponderada de sus entradas, dependiendo de si este valor se encuentra por encima o por debajo de un umbral (Matich, 2001),
  Existen numerosas funciones de activación como: la función escalón, Sigmoidal, Rectificadora (ReLu), SoftMax, tangente hiperbólica, etc.

- Función de Pérdidas. Nos proporciona la capacidad de medir el error que se ha producido en el proceso. Será, durante la etapa de entrenamiento, donde el error irá disminuyendo para obtener un modelo más preciso (Marrero, 2022).

- Optimizador. Es el método encargado de generar pesos, cada vez mejores, con cada iteración, disminuyendo, en consecuencia, el error cometido por la red. Ejemplos de optimizadores son: Stochastic Gradient Descent (SGD), Adagrad o RMSprop, entre otros (Martínez, 2020).

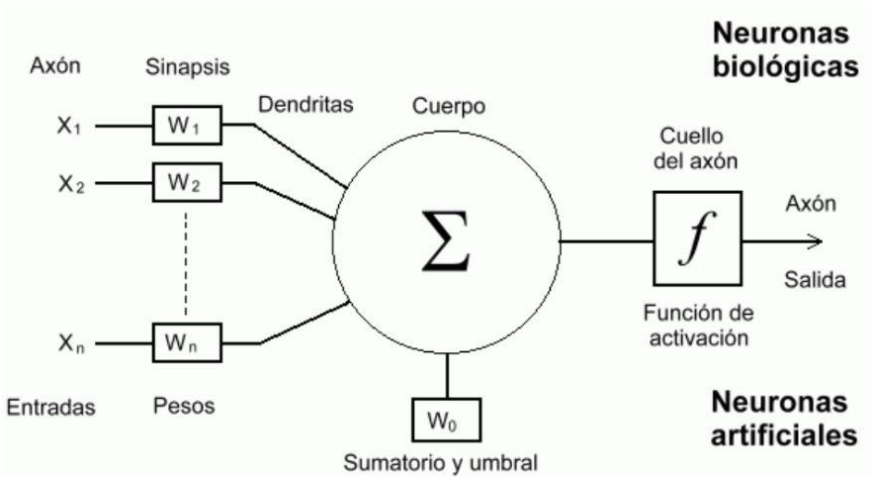

Figura 9. Comparación Neurona Biológica y Artificial.

4.3.2. Arquitectura de las Redes Neuronales.

La arquitectura de las redes neuronales hace referencia a la disposición de las neuronas en capas según el algoritmo que se emplee para su entrenamiento, lo que determina el comportamiento de la red (ver Figura 10).
Encontramos 3 tipos de capas:

- Capa de Entrada. Capa compuesta por neuronas cuya función es recoger la información de entrada a la red y, además, hay tantas neuronas como elementos a clasificar.
  Por ejemplo, en el caso de que los datos de entrada sean imágenes, se deben normalizar los pixeles en valores numéricos y, en la capa de entrada, encontraremos tantas neuronas como valores haya (Matich, 2001).

- Capa Oculta. Estas capas se sitúan entre las de entrada y salida y es aquí donde se efectúan las operaciones matemáticas con el objetivo de detectar y aprender los patrones de los datos. Las neuronas de estas capas pueden estar conectadas entre sí de distintas formas, por lo que esto, junto con el número de neuronas, definirá la topología de la red.
  Por otra parte, las redes neuronales se considerarán más profundas cuanto mayor sea el número de capas ocultas (Franco, 2023).

- Capa de Salida. Es la última capa de la red y ofrece como resultados las predicciones del modelo. Al igual que en la capa de entrada, en la de salida también encontramos que cada neurona representa el valor final que se le asigna a cada elemento que se quería clasificar (Lozano, 2020).

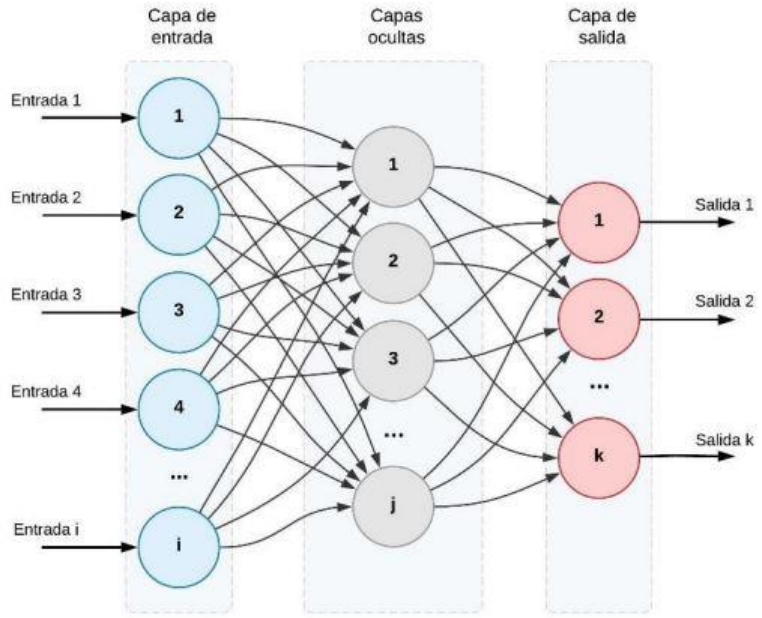

Figura 10. Arquitectura de Red Neuronal.

5. Creación del Dataset.

A continuación, se describirán las tecnologías y aplicaciones empleadas para la creación del dataset de imágenes.

5.1. Python.

"Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica" (Python, s.f.). Este lenguaje es ampliamente utilizado tanto en computación científica como en desarrollo web y se caracteriza por ser de código abierto y de uso gratuito. Compatible, además, con los sistemas operativos Windows, Linux y MacOS.

Hoy en día, Python es uno de los lenguajes de programación más utilizados entre los desarrolladores debido a la gran comunidad que lleva a sus espaldas. Además, un gran número de librerías están a disposición de los usuarios, como NumPy, Pandas, Matplotlib, SciPy o TensorFlow, entre una gran variedad, destinadas a la creación de todo tipo de aplicaciones y, en concreto, para su uso en el ámbito del Machine Learning y ciencia de datos.

En este trabajo, se ha hecho uso de este lenguaje mediante Visual Studio Code, un editor de código fuente creado por Microsoft que permite aplicar distintos lenguajes de programación como Python, C++, Java, etc.

5.2. MediaPipe.

MediaPipe se define como un marco de trabajo empelado para construir aplicaciones de Machine Learning, como detección y reconocimiento de rostros, seguimiento de manos, detección de gestos y de objetos. Este framework está desarrollado por Google y se encuentra disponible en distintas plataformas como Android, iOS, o Web, lo que lo convierte en una herramienta muy accesible.

Como herramienta en este trabajo se ha utilizado la Malla Facial o Face Mesh (ver Figura 11). Esta es una solución que, con una imagen de entrada, traza una malla facial sobre el rostro de la persona de la imagen, estimando 478 puntos, denominados 'Landmarks', sobre el rostro.
Para llevar a cabo esta tarea, emplea el aprendizaje automático para detectar la geometría de la cara en 3D por medio de una imagen, video o retransmisión en tiempo real.

Para la detección facial emplea dos modelos basados en redes neuronales convolucionales (CNN): uno para la detección (Face Detection Model) y cálculo de la posición de la cara y otro modelo de punto de referencia facial (Face Landmark Model), que genera las posiciones de los puntos 3D de la cara e identifica los contornos de los ojos, cejas labios y el del rostro al completo, además de la probabilidad de inclinación que tenga la cara en ese momento.
Conjuntamente, también incluye otro modelo de Malla de Atención (Attention Mesh Model) que proporciona puntos de referencia de mayor precisión en las regiones de la cara más significativas, como son los labios, los ojos y los iris (FaceMesh, s.f.).

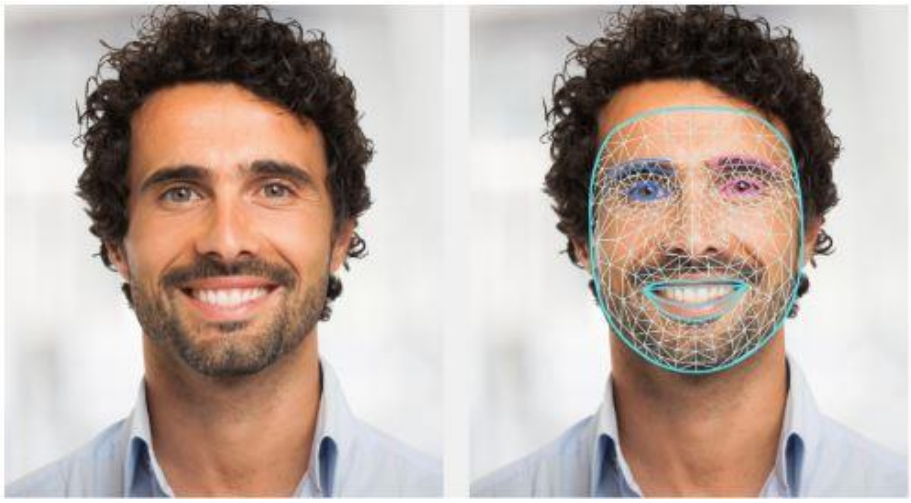

Figura 11. Face Mesh de MediaPipe

Para la detección de expresiones faciales, se ha empleado el Modelo Blendshape V2, basado en red neuronal convolucional, que toma como entrada 146 landmarks, generadas por el modelo Face Mesh, y devuelve 52 valores de Blendshape. Estos son coeficientes, comprendidos en el rango [0,1], que representan diferentes expresiones faciales.

Estos valores nos servirán posteriormente para asociarlos con las distintas expresiones faciales y clasificar así las emociones de los usuarios.

Los 52 coeficientes se muestran en la Figura 12.

1 - browDownLeft
2 - browDownRight
3 - browInnerUp
4 - browOuterUpLeft
5 - browOuterUpRight
6 - cheekPuff
(blendshape predicted by
the FaceMesh model)
7 - cheekSquintLeft
8 - cheekSquintRight
9 - eyeBlinkLeft
10 - eyeBlinkRight
11 - eyeLookDownLeft
12 - eyeLookDownRight
13 - eyeLookInLeft
14 - eyeLookInRight
15 - eyeLookOutLeft
16 - eyeLookOutRight
17 - eyeLookUpLeft
18 - eyeLookUpRight
19 - eyeSquintLeft
20 - eyeSquintRight
21 - eyeWideLeft
22 - eyeWideRight
23 - jawForward
24 - jawLeft
25 - jawOpen
26 - jawRight
27 - mouthClose
28 - mouthDimpleLeft
29 - mouthDimpleRight
30 - mouthFrownLeft
31 - mouthFrownRight
32 - mouthFunnel
33 - mouthLeft
34 - mouthLowerDownLeft
35 - mouthLowerDownRight
36 - mouthPressLeft
37 - mouthPressRight
38 - mouthPucker
39 - mouthRight
40 - mouthRollLower
41 - mouthRollUpper
42 - mouthShrugLower
43 - mouthShrugUpper
44 - mouthSmileLeft
45 - mouthSmileRight
46 - mouthStretchLeft
47 - mouthStretchRight
48 - mouthUpperUpLeft
49 - mouthUpperUpRight
50 - noseSneerLeft
51 - noseSneerRight
52 - tongueOut (blendshape
predicted by the FaceMesh
model)

Figura 12. 52 valores Blendshape.

5.3. Recopilación de imágenes.

El dataset realizado para este trabajo está formado por imágenes de personas de todas las edades expresando distintas emociones.
Para ello, se han creado 5 carpetas que representan cada emoción básica: Neutralidad, Enfado, Felicidad, Tristeza y una última carpeta, Distraído, que almacena las imágenes en las que las personas no están prestando atención a la cámara.
El propósito de esta última carpeta es el de poder detectar la atención que muestra una persona, por ejemplo, ante la realización de una actividad. Su uso está pensado, concretamente, para las personas mayores en el ámbito de las residencias de ancianos, con el objetivo de valorar si una persona está atendiendo a las actividades que realiza o si, por el contrario, se muestra distraída, lo que podría darnos señales sobre el estado de ánimo de los usuarios.

Para la composición del dataset, se han recopilado imágenes de 2 conjuntos de datos distintos. El primero es el CK+, comentado anteriormente, donde las imágenes tienen un tamaño de 640x480 y la mayoría se muestran en escala de grises, solo un mínimo de muestras se encuentra a color (Roy, 2021).

De este conjunto se ha tomado muestras para las emociones Neutralidad, Enfado, Felicidad y Tristeza (ver Figura 13).

Figura 13. Imágenes de muestra de CK+ Dataset.

El segundo conjunto es el FEI Dataset, este es una recopilación de imágenes de rostros creado en el Laboratorio de Inteligencia Artificial en São Bernardo do Campo, São Paulo, Brasil, entre los años 2005 y 2006. En la creación de este conjunto participaron 200 personas, 100 hombre y 100 mujeres de entre 19 y 40 años, de los que se tomaron 14 imágenes de cada uno, sumando un total de 2800 muestras de tamaño 640x480 píxeles. Todas ellas se presentan a color y sobre un fondo claro homogéneo, donde se aprecia a los participantes con una posición frontal erguida con rotación de perfil de hasta 180 grados (Thomaz, s.f.).

A partir de este conjunto de datos, se ha obtenido muestras para las emociones Felicidad y Neutralidad de las imágenes frontales y para la categoría Distraído, se han tomado las fotografías donde el participante no se encuentra mirando directamente a la cámara y además se sitúa girado ante la misma (ver Figura 14).

Figura 14. Imágenes de muestra de FEI Dataset.

Entre estos dos datasets se ha recopilado en total 1002 imágenes, quedando para cada categoría:

- 1. Neutralidad: 263
- 2. Enfado: 58
- 3. Felicidad: 193
- 4. Tristeza: 40
- 5. Distraído: 448

Una vez recopiladas las imágenes, se obtendrán los landmarks de los rostros de las muestras a través del modelo MediaPipe Blendshape V2, mencionado anteriormente.
Este proceso se ha llevado a cabo a través de la aplicación Visual Studio Code, creando un código en Python para ello, importando la solución de MediaPipe a nuestro entorno siguiendo el siguiente procedimiento:

Creamos un archivo Python e instalamos el paquete PyPi para poder utilizar MediaPipe Face Landmarker, mediante el comando:

```
python -m pip install mediapipe
```

Seguidamente importamos las librerías necesarias:

```
import mediapipe as mp
from mediapipe.tasks import python
from mediapipe.tasks.python import vision
import cv2

from mediapipe import solutions
from mediapipe.framework.formats import landmark_pb2
import numpy as np
import matplotlib.pyplot as plt
```

Agregamos las funciones necesarias para visualizar los landmarks sobre el rostro tras la detección

```
BaseOptions = mp.tasks.BaseOptions
FaceLandmarker = mp.tasks.vision.FaceLandmarker
FaceLandmarkerOptions = mp.tasks.vision.FaceLandmarkerOptions
VisionRunningMode = mp.tasks.vision.RunningMode

def draw_landmarks_on_image(rgb_image, detection_result):
  face_landmarks_list = detection_result.face_landmarks
  annotated_image = np.copy(rgb_image)
```

```
 # Loop through the detected faces to visualize.
 for idx in range(len(face_landmarks_list)):
  face_landmarks = face_landmarks_list[idx]

  # Draw the face landmarks.
  face_landmarks_proto = landmark_pb2.NormalizedLandmarkList()
  face_landmarks_proto.landmark.extend([
   landmark_pb2.NormalizedLandmark(x=landmark.x, y=landmark.y, z=landmark.z) for
landmark in face_landmarks
  ])

  solutions.drawing_utils.draw_landmarks(
    image=annotated_image,
    landmark_list=face_landmarks_proto,
    connections=mp.solutions.face_mesh.FACEMESH_TESSELATION,
    landmark_drawing_spec=None,
    connection_drawing_spec=mp.solutions.drawing_styles
    .get_default_face_mesh_tesselation_style())
  solutions.drawing_utils.draw_landmarks(
    image=annotated_image,
    landmark_list=face_landmarks_proto,
    connections=mp.solutions.face_mesh.FACEMESH_CONTOURS,
    landmark_drawing_spec=None,
    connection_drawing_spec=mp.solutions.drawing_styles
    .get_default_face_mesh_contours_style())
  solutions.drawing_utils.draw_landmarks(
    image=annotated_image,
    landmark_list=face_landmarks_proto,
    connections=mp.solutions.face_mesh.FACEMESH_IRISES,
     landmark_drawing_spec=None,
     connection_drawing_spec=mp.solutions.drawing_styles
     .get_default_face_mesh_iris_connections_style())

 return annotated_image

def plot_face_blendshapes_bar_graph(face_blendshapes):
  # Extract the face blendshapes category names and scores.
  face_blendshapes_names = [face_blendshapes_category.category_name for
face_blendshapes_category in face_blendshapes]
  face_blendshapes_scores = [face_blendshapes_category.score for
face_blendshapes_category in face_blendshapes]
  # The blendshapes are ordered in decreasing score value.
  face_blendshapes_ranks = range(len(face_blendshapes_names))

  fig, ax = plt.subplots(figsize=(12, 12))
  bar = ax.barh(face_blendshapes_ranks, face_blendshapes_scores, label=[str(x) for x in
face_blendshapes_ranks])
  ax.set_yticks(face_blendshapes_ranks, face_blendshapes_names)
  ax.invert_yaxis()
```

```
  # Label each bar with values
  for score, patch in zip(face_blendshapes_scores, bar.patches):
    plt.text(patch.get_x() + patch.get_width(), patch.get_y(), f"{score:.4f}", va="top")
  ax.set_xlabel('Score')
  ax.set_title("Face Blendshapes")
  plt.tight_layout()
  plt.show()
```

Esta solución requiere del modelo entrenado que descargamos con:

```
model_path = '/absolute/path/to/face_landmarker.task'
```

A continuación, creamos un objeto para especificar la ruta del modelo y configuramos la función para que trabaje sobre una imagen, indicándolo en *running_mode=VisionRunningMode.IMAGE*, ya que existe la posibilidad de emplear esta solución con video y mediante transmisión en vivo. Queda de la siguiente forma:

```
base_options = python.BaseOptions(model_asset_path='face_landmarker.task')

options = FaceLandmarkerOptions(base_options=base_options,
    running_mode=VisionRunningMode.IMAGE,
    output_face_blendshapes=True,
    output_facial_transformation_matrixes=True,
    num_faces=1)

detector = vision.FaceLandmarker.create_from_options(options)
```

Ahora es el turno de preparar la entrada de imágenes en formato NumPy:

```
numpy_image = cv2.imread("Imagen_1.jpeg")
image = mp.Image(image_format=mp.ImageFormat.SRGB, data=numpy_image)
```

Ya podemos hacer uso del modelo para que detecte los landmarks y muestre la imagen final:

```
# Detectar los puntos de referencia del rostro en la imagen de entrada.
detection_result = detector.detect(image)

# Procesar y visualizar el resultado de la detección
annotated_image = draw_landmarks_on_image(image.numpy_view(), detection_result)
cv2.imshow("Imagen final", annotated_image)
cv2.waitKey(0)
```

El resultado que se obtiene es el que se muestra en la Figura 15 y la Figura 16.

El siguiente paso es obtener los valores numéricos de los landmarks, para ello utilizaremos la siguiente función que almacenará estos valores en un archivo del formato CSV. Este contiene datos tabulares separados por comas. Cada línea representará los coeficientes y características de una persona.

Figura 15. Imagen Original.

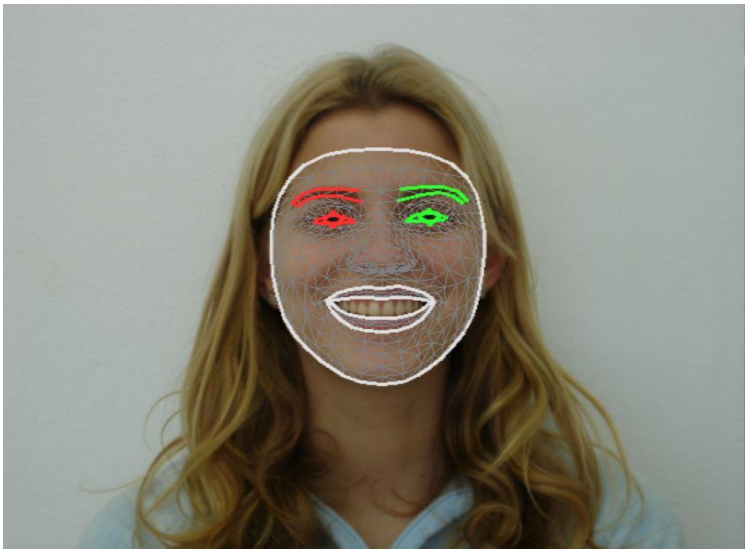

Figura 16. Imagen con Face Mesh.

```
#Libreria necesaria para trabajar con archivos CSV
import pandas as pd

def create_face_blendshapes_dataframe(face_blendshapes):
    data = []
    for face_blendshape_category in face_blendshapes:
        category_name = face_blendshape_category.category_name
        score = face_blendshape_category.score
        data.append({"Category": category_name, "Score": score})

    df = pd.DataFrame(data)
    return df

result_df = create_face_blendshapes_dataframe(detection_result.face_blendshapes[0])

# Redondear los valores a 4 decimales y formatear como cadena
result_df['Score'] = result_df['Score'].round(4).map('{:.4f}'.format)
```

```
# Transponer el DataFrame para tener una fila para Category y otra para Score
result_df_transposed = result_df.T
# Guardar el DataFrame transpuesto en un archivo CSV con comas como delimitador
csv_file = "resultados.csv"
result_df_transposed.to_csv(csv_file, sep=',', header=None)
```

Esta función da como resultado un archivo “resultados.csv” que contiene dos filas: la primera es ‘Category’, donde se muestran los 52 nombres de los coeficientes del modelo Blendshape; y la segunda es ‘Score’, donde aparecen los valores correspondientes a esos 52 coeficientes.
Se ha preparado el resultado de esta manera para poder copiar los valores y almacenarlos en otro archivo, denominado “Dataset.csv”, puesto que, al repetir todo el proceso explicado con cada imagen, se van actualizando los coeficientes en el primer archivo “resultados.csv”.

Un ejemplo de la solución que ofrece esta función es el siguiente, donde aparecen los 5 primeros coeficientes y sus valores del Blendshape:

```
Category,_neutral,browDownLeft,browDownRight,browInnerUp,browOuterUpLeft, …
Score,0.0000,0.1513,0.1683,0.0057,0.0132,…
```

Por otro lado, el archivo “Dataset.csv” contiene 55 columnas estructuradas de la siguiente forma:

H/M, Edad, [52 coeficientes Blendshape], Emoción

- H/M: Corresponde al género de la persona, a los hombres se les asigna ‘0’, mientras que a las mujeres ‘1’.
- Edad: Se han creado 3 categorías para indicar la edad:
  Joven: 0
  Adulto: 1
  Mayor: 2
  La evaluación de este apartado es subjetiva debido a la falta de datos precisos sobre la edad de cada participante, en consecuencia, esta distinción se ha realizado de manera visual y no en la edad real de los colaboradores.
- Seguido de la edad, se sitúan los 52 valores del modelo.
- Emoción: esta última columna corresponde con la expresión que muestra la imagen de cada persona y se clasifican de la siguiente forma:
  Neutralidad: 1
  Enfado: 2
  Felicidad: 3
  Tristeza: 4
  Distraído: 5

Un ejemplo, aplicado a la imagen de la figura 15, sería el siguiente:

```
H/M, Edad,_neutral,browDownLeft,browDownRight,browInnerUp,…, Emoción
1, 1, 0.0000,0.1513,0.1683,0.0057, …, 3
```

Una vez completado el archivo con 1002 imágenes, se realiza la preparación de los datos para su empleo en la herramienta Matlab.

Se han hecho 2 clasificaciones de los datos según el género y la edad, de forma que ahora tenemos:

- Un archivo “Datos.cvs” donde se ha eliminado las columnas H/M y Edad, de forma que todas las variables son continuas, salvo la variable salida que es categórica y corresponde con la columna Emoción.
- Primera clasificación: Género. Se han creado dos archivos: “Hombre.csv” donde solo se ha incluido los datos cuya columna H/M tenía el valor ‘0’, y “Mujer.csv”, para el valor ‘1’ de esa misma columna.
- Segunda clasificación: Edad. En esta parte tenemos 3 archivos: “Joven.csv”, “Adulto.csv” y “Mayor.csv” según el valor de la columna Edad, como sea mencionado anteriormente.

Por lo tanto, los 6 archivos tienen en común que tienen 53 columnas: las 52 del modelo Blendshape + la columna salida, Emoción.
Organizándolo de esta forma se consigue que todas las variables sean continuas, es decir, variables que toman cualquier valor numérico, mientras que solo tenemos una variable categórica, que corresponderá con la salida del modelo de clasificación que explicaremos más adelante.
Las variables categóricas hacen referencia a variables no numéricas, como las emociones, por lo que nos facilitan la clasificación de una serie de datos, empleando valores fijos, asociados a una categoría o cualidad concreta (Arias, 2021).

6. Herramientas para la clasificación de características y entrenamiento de modelos.

Puesto que ya tenemos todos los datos clasificados y preparados, se realizará la preselección de las características más relevantes, de entre las 52 que tenemos del modelo Blendshape y el posterior entrenamiento de los modelos de clasificación en Matlab.

Esta extracción de características se lleva a cabo con el objetivo de identificar qué relación hay entre lo que los humanos hacemos, subjetivamente, para poder identificar las expresiones faciales y lo que los métodos estadísticos, de selección de características, proponen como rasgos distintivos para reconocer una expresión.

Las personas llevamos a cabo una serie de procesos, inconscientemente, para lograr reconocer una emoción a partir de una expresión facial, que podríamos describir como: la percepción visual, lo que nos permite identificar el género y la edad de una persona; la observación de aquellos rasgos faciales que denotan mayor emoción, como pueden ser los ojos y la boca y, finalmente, la identificación de la emoción. (Cereceda, 2010)

Aun así, el reconocimiento de emociones es complicado incluso para las personas, puesto que no siempre concuerda la expresión facial que se muestra con el sentimiento interno. Por ejemplo, ante una situación incómoda, puede aparecer el rostro una sonrisa que no se identifica con el sentimiento de alegría, no obstante, un modelo de clasificación podría categorizar esta expresión como felicidad erróneamente.

Por lo tanto, vamos a investigar que coeficientes del Blendshape son más significativos, mediante un método estadístico, para reconocer las emociones, haciendo uso de la herramienta Classification Learner App de Matlab.

6.1. Matlab. Classification Learner.

Matlab es conocido por ser una "*plataforma de programación y cálculo numérico utilizada por científicos e ingenieros para analizar datos, desarrollar algoritmos y crear modelos.*" (MATLAB, s.f.)

Dentro de esta plataforma, encontramos la aplicación Classification Learner, que se emplea para el entrenamiento de modelos de clasificación de datos. Esta herramienta nos permite, a partir de nuestros datos, elegir las características con las que trabajar, seleccionar esquemas de validación, entrenamiento de modelos y analizar los resultados.
Entre los distintos modelos de clasificación que nos ofrece esta aplicación encontramos: árboles de decisión, redes neuronales, Naive Bayes, Ensemble, regresión logística, K-nearest Neighbors, entre otros. (MATLAB, s.f.)

Antes de describir el procedimiento de selección de características y entrenamiento de los modelos, se definirá las herramientas empleadas durante esta tarea.

6.2. Algoritmo de Mínima Redundancia y Máxima Relevancia.

Para la selección de características se empleará el algoritmo de MRMR (Mínima Redundancia y Máxima Relevancia).
El algoritmo tiene como objetivo hallar la características o variables que tienen un mayor peso para conseguir predecir la variable salida (relevancia), que en nuestro caso es 'Emoción'. Una vez recopiladas estas variables, se procede a eliminar aquellas que son 'relevantes' cuya aportación la puede ofrecer otra variable (redundancia). (González, 2022)

Se puede definir de la siguiente forma, donde Vc corresponde a la relevancia y Wc a la redundancia:

$$Vc(X_i) = \frac{1}{|S|} \sum_{X_i \in S} I(X_i, Y)$$

$$Wc(X_i, X_j) = \frac{1}{|S|^2} \sum_{X_i, X_j \in S} |I(X_i, X_j)|$$

En el que S es el conjunto de características seleccionadas, $|S|$ su cardinal e I es una función que mide el grado de relación entre dos características. (González, 2022)

La ventaja que obtenemos de emplear el MRMR es la eliminación de las variables poco relevantes y redundantes, obteniendo así, un algoritmo más simplificado y con mejor interpretabilidad.

6.3. Matrices de Confusión.

La matriz de confusión (ver Figura 17) es una herramienta que nos permite evaluar la precisión de un modelo de clasificación, donde la salida puede ser dos o más categorías. Esta matriz se dispone en forma de tabla donde las filas corresponden con las clases observadas, mientras que las columnas hacen referencia a las predicciones de cada categoría.

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 17. Matriz de Confusión.

Si visualizamos la tabla, los 'verdaderos positivos' y 'verdadero negativos' indican que la predicción coincide con el valor actual, mientras que los 'falsos positivos' y 'falsos negativos' son predicciones erróneas que no concuerdan con los valores actuales.

Puede comprenderse mejor con un ejemplo. Esto es, si queremos resolver un problema de clasificación para detectar si un paciente está enfermo o no, tendremos:

- Verdadero positivo (VP): un paciente es detectado correctamente que está enfermo.
- Falso Positivo (FP): indica que el paciente está enfermo cuando en la realidad no lo está.
- Falso Negativo (FN): el paciente está enfermo, pero es detectado como sano.
- Verdadero Negativo (VN): se detecta correctamente que el paciente está enfermo.

El caso ideal sería obtener 0 falsos negativos y 0 falsos positivos, no obstante, cualquier modelo de clasificación no tiene una precisión del 100%, por lo que buscaremos aquellos que tengan una menor tasa de falsos negativos y positivos en la predicción.

6.4. Curvas ROC.

La curva ROC (Receiver Operating Characteristics) es una representación gráfica que se emplea para valorar el rendimiento de un modelo de aprendizaje automático (ver Figura 18). En ella se muestra la sensibilidad, que corresponde con la proporción de verdaderos positivos, frente a la especificidad, que se identifica como proporción de falsos positivos. Ambos representados mediante valores del rango [0, 1] (Pérez & P.S. Pérez Martin, 2023).

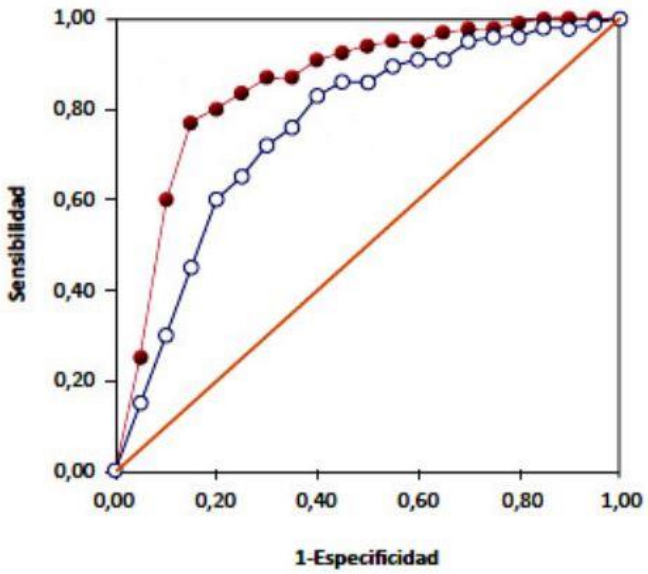

Figura 18. Curva ROC.

Esta curva se forma gracias a la unión de los distintos puntos de corte, que representan una sensibilidad y especificidad concreta. El punto de corte que presenta mayor especificidad y sensibilidad se situaría muy próximo al ángulo superior izquierdo de la gráfica, cera del punto (0,1), por lo que, este punto, que presentaría el mayor índice de Youden [Sensibilidad + Especificidad - 1], establece la mayor sensibilidad y especificidad conjunta. (Pérez & P.S. Pérez Martin, 2023)

Otra característica de la curva ROC es el AUC, o Área Bajo la Curva, que mide la capacidad discriminativa del modelo clasificador, es decir, la habilidad que posee el modelo para distinguir entre las distintas clases.
Por otro lado, la diagonal de referencia o línea de no-discriminación, es aquella línea que se traza desde el punto (0,0) hasta el (1,1), que representa un modelo incapaz de clasificar entre las distintas clases, es decir, sobre esta línea se encuentra el mismo número de falsos positivos y verdaderos positivos y, además, se corresponde con el valor AUC = 0.5. (Cerda & Cifuentes, 2012)

Si tomamos el mismo ejemplo de la matriz de confusión, un clasificador de pacientes enfermos y no enfermos, buscaríamos aquel modelo que presente una curva ROC lo más alejada posible de la diagonal de referencia con el objetivo de conseguir una mayor discriminación entre pacientes sanos y enfermos.
Además, cuanto más alejada esté la curva de esta línea, mayor será su AUC. Una curva con un AUC = 0.5 se considera no discriminatorio, mientras que AUC = 1 corresponde con la máxima discriminación, lo que supone el 100% de verdaderos positivos y ningún falso positivo.
Al igual que comentamos en la matriz de confusión, este caso sería el ideal, pero no es lo que sucede en la realidad, por lo que es preferible obtener un valor AUC mayor a 0.5, siendo el 0.75 el punto intermedio entre la discriminación y no discriminación, y un punto de corte lo más cercano posible a la esquina superior izquierda.

Obteniendo de esta forma, una mayor área bajo la curva (mayor capacidad discriminatoria) y un alto punto de corte (mayor sensibilidad y especificidad conjunta.)

6.5. Modelos de clasificación.

Se realizará una revisión de los modelos de clasificación disponibles en la aplicación Classification Learner.

6.5.1. Árboles de decisión.

Un árbol de decisión es un algoritmo característico del aprendizaje supervisado que se emplea tanto en problemas de clasificación, ofreciendo respuestas nominales como ‘Verdadero’ o ‘Falso’, como en regresión, cuyas respuestas son de tipo numérico (ver Figura 19).
En su estructura encontramos un nodo raíz desde el cual comienzan las ramas hacia otros nodos internos, o nodos de decisión. Estos nodos tienen la función de predecir clasificaciones mediante la comprobación del valor de un predictor (variable), hasta llegar a una solución para cada rama que se representa con un nodo hoja o terminal (Breiman, Friedman, Olshen, & Stone, 1984).

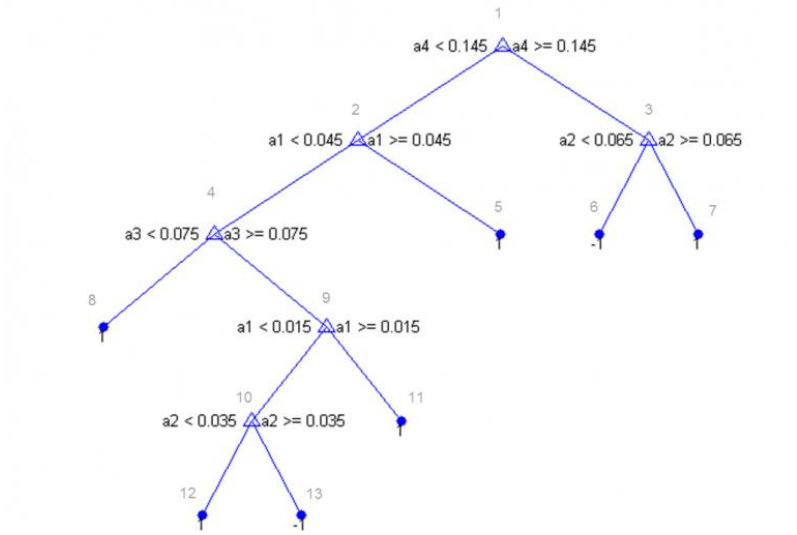

Figura 19. Estructura de un árbol de decisión.

El método que siguen los árboles de decisión para aprender se basa en la búsqueda de la división óptima de sus ramas, de forma repetida, hasta que todos los datos estén clasificados en distintas clases. Estos árboles pueden llegar a ser muy complejos, generando así, un sobreajuste. Es por ello que los árboles más sencillos, con pocas ramas, son los que mejores resultados ofrecen puesto que son capaces de conseguir nodos hoja puros, es decir, clasificaciones de los datos en una sola clase.
Cuando la complejidad del árbol es tan grande, se lleva a cabo un proceso denominado "poda", que consiste eliminar aquellas ramas que se dividen en características de poca relevancia (IBM, s.f.).

6.5.2. Análisis discriminante.

El método de clasificación de análisis discriminante es una técnica estadística que tiene el propósito de obtener funciones discriminantes que posibiliten la clasificación de datos dentro de categorías previamente definidas.

Estas funciones discriminantes son combinaciones lineales de aquellas variables independientes que consiguen una mejor distinción entre las diversas clases, mediante el análisis de la relación que existe entre una variable dependiente categórica y un conjunto de datos independientes (Morales, 2014).

6.5.3. Naïve Bayes.

El algoritmo Naïve Bayes se caracteriza por ser un clasificador probabilístico, que se basa en el Teorema de Bayes, que se emplea para llevar a cabo predicciones.
Este método supone la independencia entre los predictores, es decir, que el efecto de una característica es independiente a otra dentro del mismo conjunto de datos. Esta particularidad se llama independencia condicional de clase.

El proceso se define mediante la siguiente fórmula:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Donde:

- $P(h|D)$: Probabilidad de la hipótesis h dada los datos D.
- $P(D|h)$: probabilidad posterior de que se de h dado los datos D.
- $P(h)$: Probabilidad de h.
- $P(D)$: Probabilidad de D.

De forma que el proceso se lleva a cabo de la siguiente manera: se calcula la probabilidad previa para las estadísticas de las categorías; se determina la probabilidad con cada atributo para cada una de las clases y se calcula la probabilidad posterior aplicando estos valores en el teorema de Bayes.
La clase que obtenga el mayor valor será aquella a la que pertenezca el dato de entrada. (Roman, 2019)

6.5.4. K-nearest neighbors.

El algoritmo, no paramétrico, de aprendizaje supervisado K-nearest neighbors, K-NN o algoritmo de los K vecinos más próximos, lleva a cabo las clasificaciones basándose en la proximidad de los datos en un conjunto, es decir, sobre un conjunto de datos, se calcula la distancia existente entre un dato y el resto del conjunto (ver Figura 20).
Seguidamente se ordenan estos valores para conocer cuáles son los vecinos más cercanos a este punto o dato, en consecuencia, la clase predicha para este dato viene condicionada por la clase a la que pertenezcan los vecinos más próximos (Arbeloa, 2018).

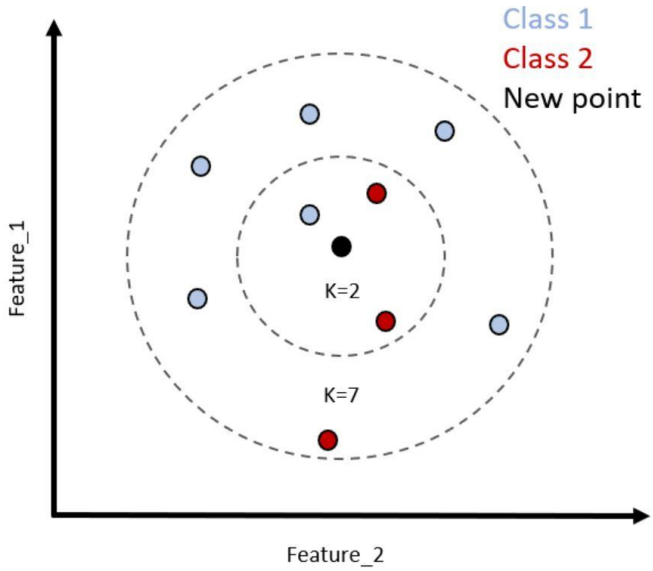

Figura 20. K-Nearest Neighbors

El número de vecinos viene dado por el valor 'k'.
En el ejemplo de la Figura 20, si k=2, al punto central (negro) se le asignará la clase 2, puesto que hay más puntos pertenecientes a esta clase que a la clase 1.
Por el contrario, si k=7, se designa la clase 1 a este punto, ya que hay una mayoría de puntos correspondientes a esta clase.

6.5.5. Máquinas de soporte vectorial.

Las máquinas de soporte vectorial (SVM) son algoritmos de aprendizaje estadístico supervisado muy empleados para las tareas de clasificación y regresión de datos. Estos se basan en la búsqueda de un hiperplano en un espacio de dimensionalidad muy alta.

Si tenemos, por ejemplo, un conjunto de datos que queremos clasificar en dos categorías (ver Figura 21), la máquina de vector de soporte tiene el objetico de construir un hiperplano que consiga la separación óptima, es decir, que logre maximizar la distancia de separación entre los puntos de datos de cada categoría (León, 2016).

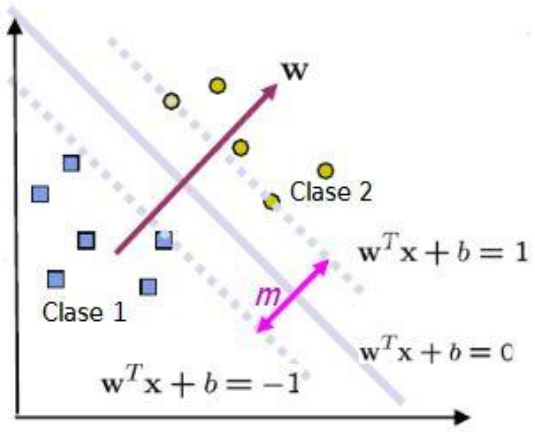

Figura 21. Representación Máquina de Vectores de Soporte

Cuando los datos no son linealmente separables, estos son transformados a un espacio dimensional superior donde pueda existir un hiperplano capaz de realizar esta separación. Esto se lleva a cabo mediante una técnica denominada “kernel trick”, ampliando así, las capacidades de las SVM (Josep, 2023).

6.5.6. Ensembles.

Un ensemble de clasificación es un modelo de predicción que se crea gracias a la combinación ponderada de diferentes modelos de clasificación para obtener mejores predicciones generales. Por tanto, este modelo no se construye a partir del entrenamiento de conjuntos de datos, sino a partir de las predicciones de distintos modelos ya entrenados.

Podemos explicar esta construcción mediante niveles, es decir, se entrenan una serie de modelos clásicos (primer nivel) y a partir de los resultados de este nivel se entrena un segundo modelo (segundo nivel).
Los modelos correspondientes al primer nivel suelen caracterizarse por ser específicos y distintos entre sí para obtener una baja correlación entre sus predicciones y aumentar su rendimiento. (Conde, 2022)

Dentro del Ensemble, encontramos distintas técnicas como el “bagging” y el “boosting”.

En bagging o bootstrap, dado un conjunto de datos se obtienen varias muestras de forma aleatoria y con ellas se entrenan los modelos de forma separada, siendo el resultado final la combinación de las predicciones de los modelos.
Boosting es un método de aprendizaje secuencial, en él se entrena un modelo con el conjunto de datos de entrenamiento al completo, de manera que los modelos posteriores se construyen ajustando los valores de error residuales del modelo inicial. El resultado será la ponderación de los valores de rendimiento de todos los modelos.

6.5.7. Redes Neuronales.

Las redes neuronales, como se ha explicado en el apartado '*5.3. Redes Neuronales*', son modelos computacionales que emulan el funcionamiento del cerebro humano para procesar información. La unidad básica de estas redes son las neuronas y se estructuran en capas, que van ajustando sus pesos en función de las predicciones que generen con la finalidad de mejorar su precisión.

La mejora del rendimiento de la red se lleva a cabo mediante el entrenamiento, mostrando a la red ejemplos de datos con resultados conocidos. Es en la retroalimentación, que recibe la red al comparar sus predicciones con los resultados, donde se consigue llevar a cabo el ajuste de los pesos y, por consiguiente, el aprendizaje de la red neuronal.

7. Descripción del proceso de selección de características y entrenamiento de los modelos de clasificación.

Como los datos están preparados y clasificados, tal como se explica en la sección 'Creación del Dataset', se importarán a Matlab para llevar a cabo el proceso de extracción de características y entrenamiento de los modelos de clasificación.

Comenzamos con el archivo de datos denominado 'Datos.csv' que contiene los 52 coeficientes del modelo Blendshape junto con la variable 'Emoción', y lo añadimos a Matlab. Como esté está en formato CSV, lo convertimos a la extensión MAT propia de Matlab, este tipo de archivos de datos se caracterizan por contener variables, matrices y funciones, entre otra información.
Para ello, seleccionamos 'Import Data' y lo guardamos con el mismo nombre, obteniendo: 'Datos.mat', tal y como se observa en la Figura 22.

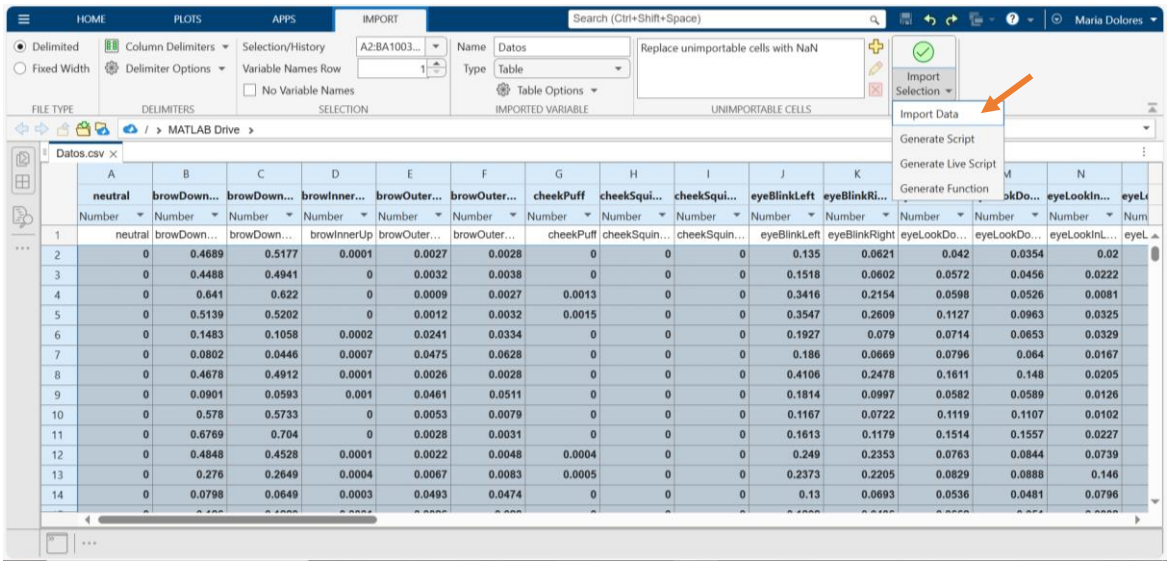

Figura 22. Importación de datos.

Una vez realizado el paso anterior, creamos un Script para cargar nuestro archivo y convertir la variable 'Emocion' en categórica mediante el siguiente código:

```
All = load("Datos.mat");
All = All.Datos;
All = convertvars(All,'Emocion','categorical');
names = All.Properties.VariableNames;
```

Una vez hecho, pulsamos el botón 'Run' y nos aparecerá las variables en la esquina izquierda inferior, en el apartado Workspace, donde hemos almacenado la tabla de datos en la variable 'All'.

A continuacion, hacemos uso de la aplicación Classification Learner, situada en la pestaña APPS, tal y como se indica en la Figura 23.

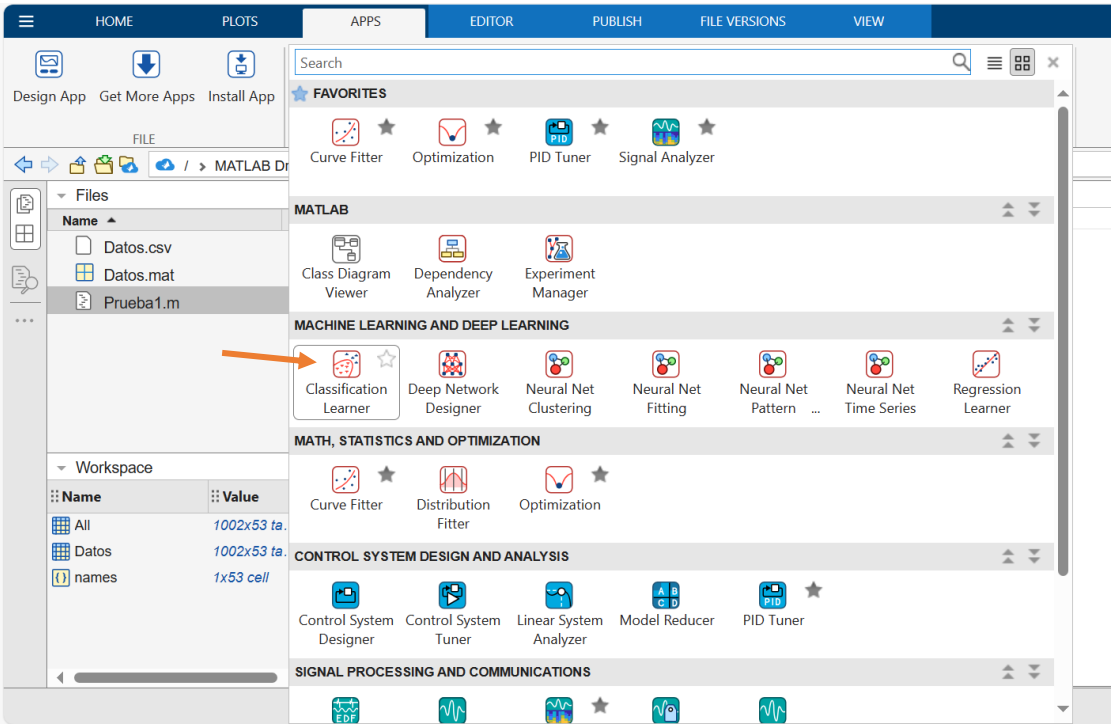

Figura 23. Selección de la aplicación en Matlab.

Una vez abierto, creamos una nueva sesión mediante el primer botón de la esquina superior izquierda 'New Session', y nos aparece una pestaña donde elegiremos las variables con las que queremos trabajar (ver Figura 24).

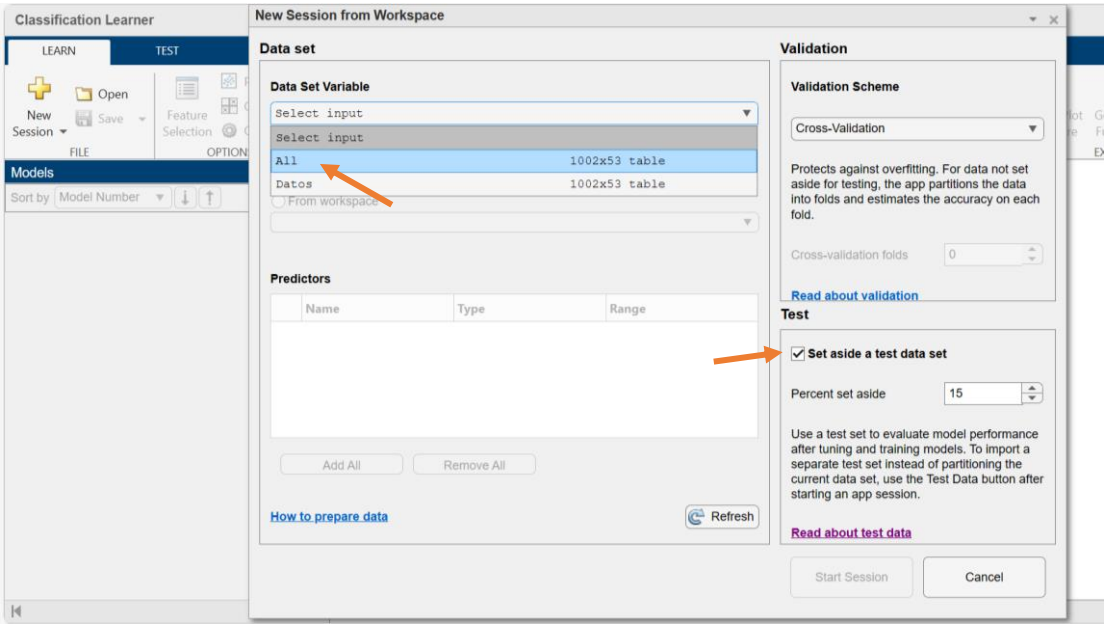

Figura 24. Creación de sesión para la aplicación de clasificación.

En el apartado 'Data Set Variable' seleccionamos nuestra variable 'All' y marcamos la casilla 'Set aside a test data set', esto significa que se reservará un porcentaje de datos para las pruebas que se realicen posteriormente.

Ya marcado, nos arecen todas las variables de nuestro conjunto y vemos como la variable 'Emocion' es categórica. Seguidamente, comenzamos la sesión en 'Start Session' tal y como se indica en la Figura 25.

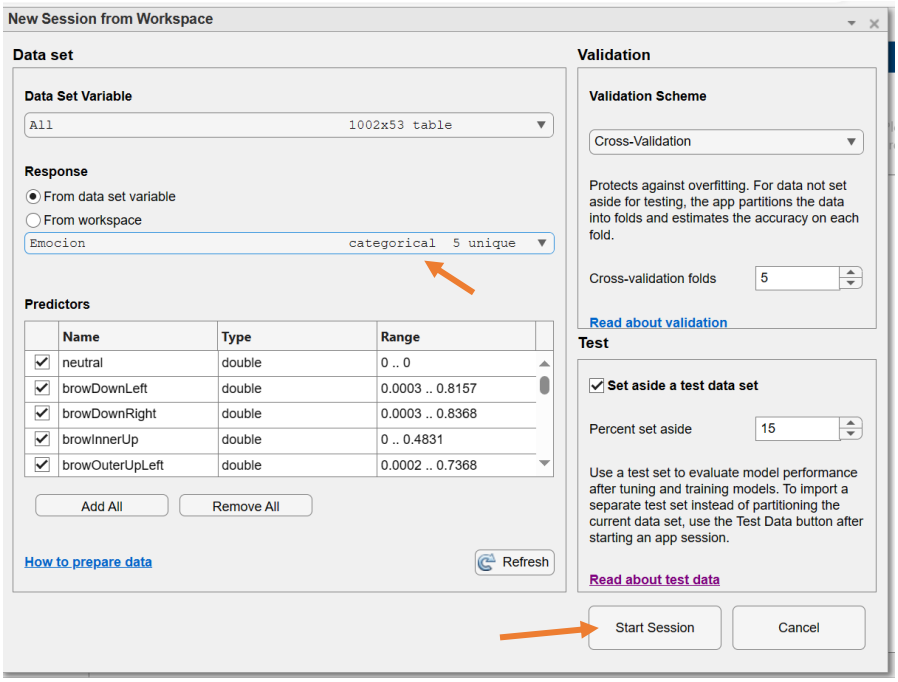

Figura 25. Configuración de la sesión para realizar el inicio de la misma.

Dentro de cada prueba se realizarán varios entrenamientos de los modelos para comparar el resultado según las características que seleccionemos.

7.1. Entrenamiento sin selección de características.

Empleando las 52 características, en el apartado 'MODELS' vamos a seleccionar todos los modelos disponibles. Si desplegamos esta pestaña, vemos como nos permite seleccionar todos los modelos, esto es por que únicamente tenemos una variable categórica, que es la salida, si tuviéramos numerosas variables categóricas, el número de modelos se reduciría debido a la forma de trabajo de este.

En nuestro caso, seleccionamos 'ALL' y pulsamos el botón 'Train All' para comenzar el entrenamiento (ver Figura 26).

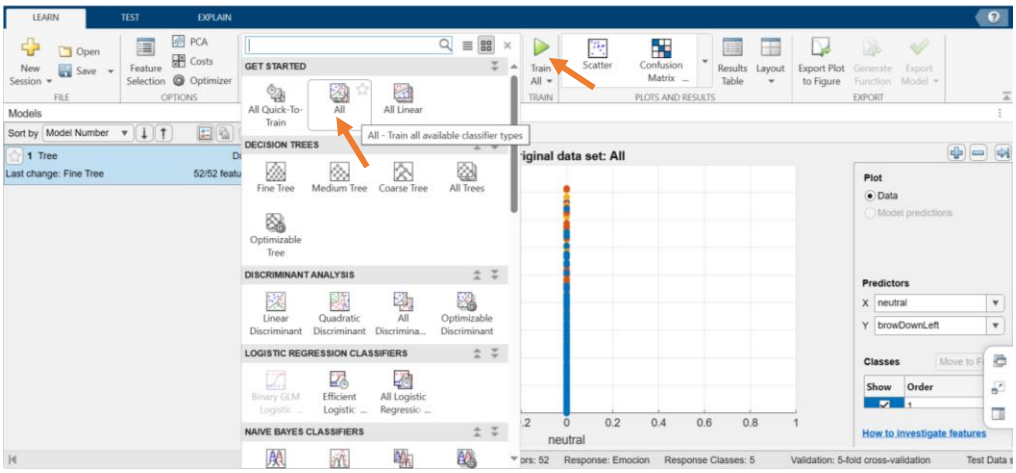

Figura 26. Selección de todos los posibles clasificadores para comparar, posteriormente, los modelos.

Este proceso de entrenamiento se completa tras un breve tiempo de espera.
Cuando ha finalizado, obtenemos todos los modelos entrenados en la columna izquierda, junto con un porcentaje que corresponde con la precisión que ha obtenido el modelo.

Aunque hemos ordenado los modelos según su precisión, debemos visualizar sus matrices de confusión y curvas ROC para decidir cuál es el que mejores resultados nos ofrece.
Para visualizar estas graficas debemos seleccionarlas en el apartado 'Plots and Results' (ver Figura 27).

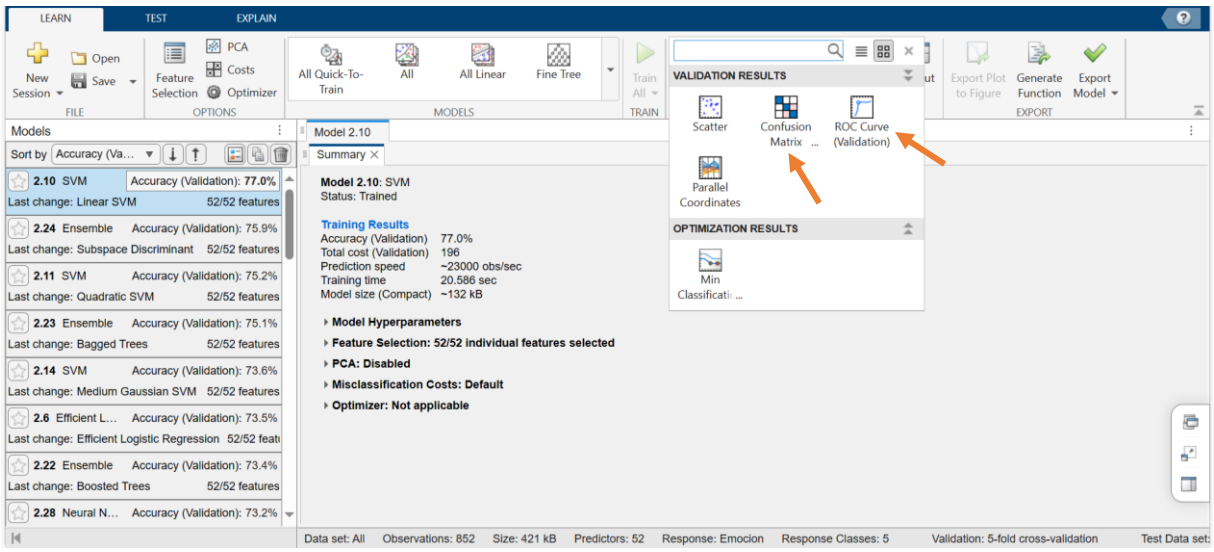

Figura 27. Proceso de validación de resultados a través de las matrices de confusión y las curvas ROC.

Para este caso, vamos a escoger tres modelos distintos y compararlos.

- <u>SVM (77%). Linear SVM.</u>

  Este modelo de máquinas de vectores de soporte es el que mayor precisión nos ofrece, siendo su matriz de confusión, su matriz de TPR y FNR y su curva ROC las de la Figura 28, Figura 29 y Figura 30, respectivamente.

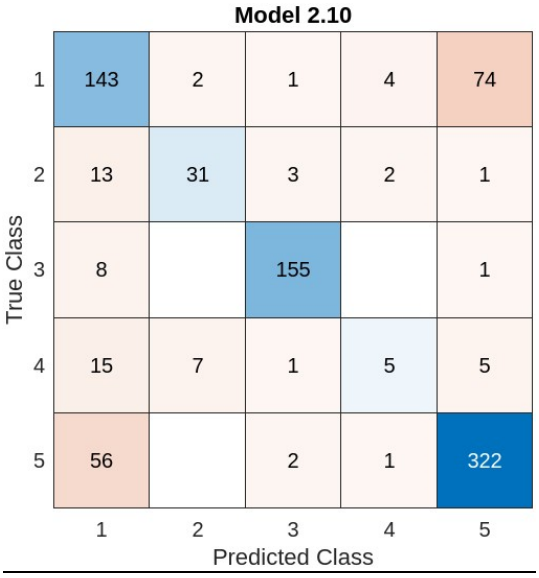

Figura 28. Matriz de Confusión SVM

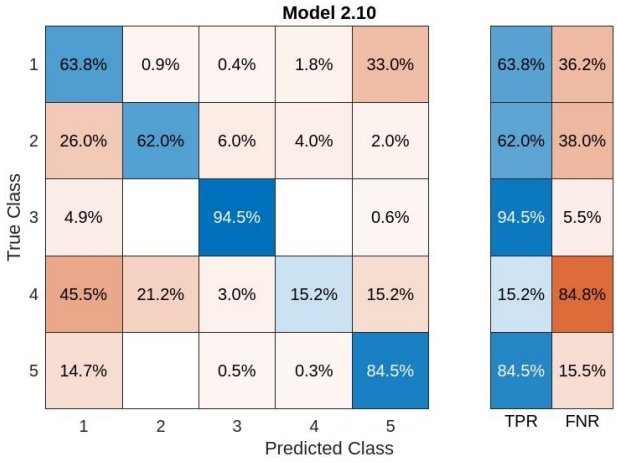

Figura 29. TPR y FNR.

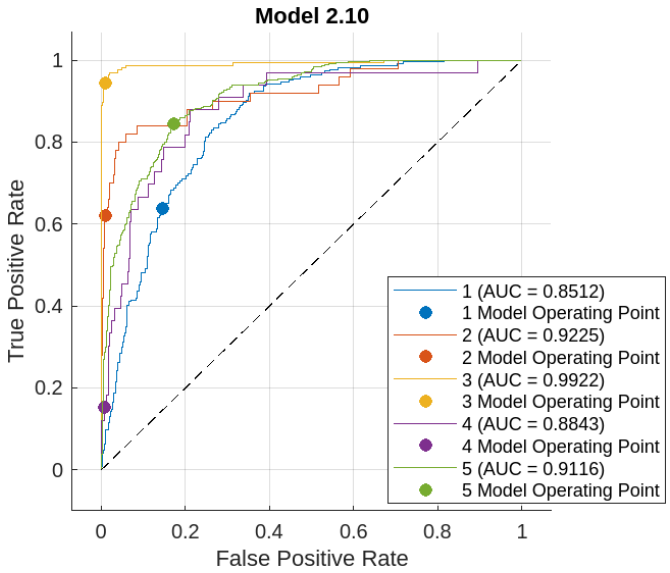

Figura 30. Curva ROC del modelo SVM.

- Ensemble (75.9%). Subspace Discriminant.
  En este modelo, se obtienen: la matriz de confusión, la curva ROC y la matriz de TPR y FNR (ver Figura 31, Figura 32 y Figura 33, respectivamente).

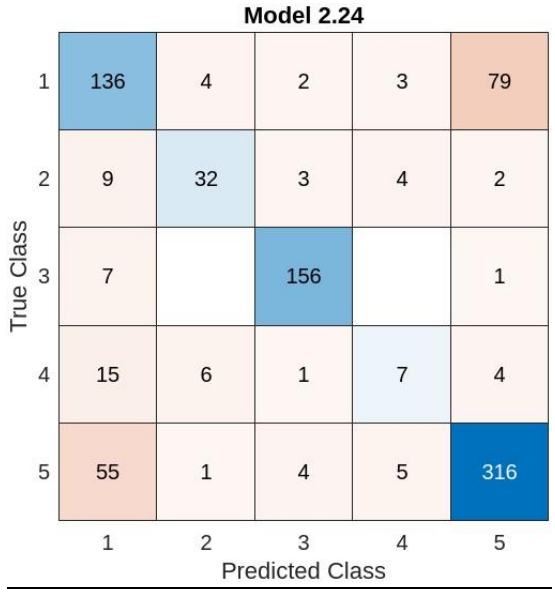

Figura 31. Matriz de confusión Ensemble.

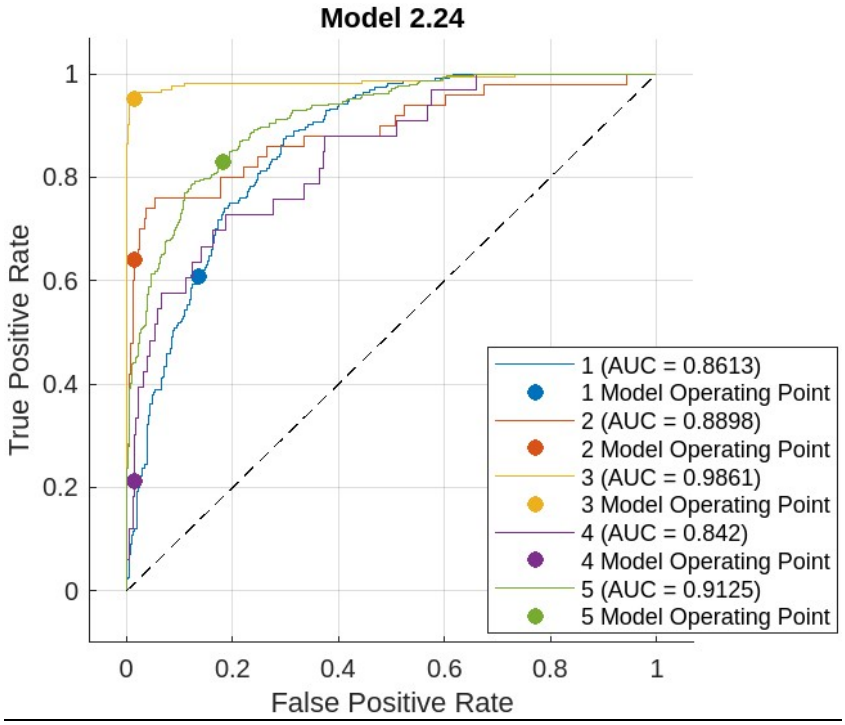

Figura 32. Curva ROC Ensemble.

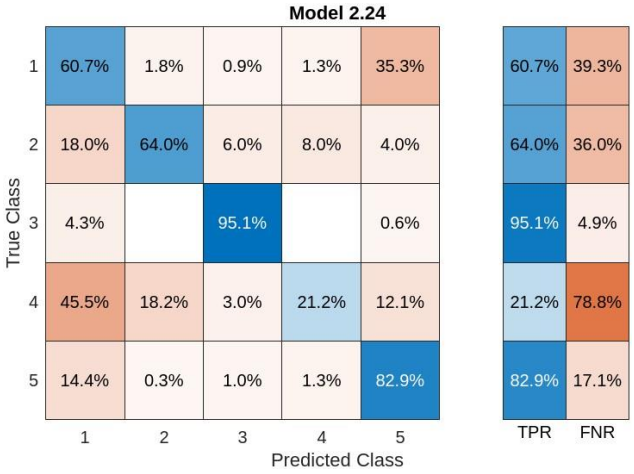

Figura 33. TPR y FNR Ensemble.

- <u>Neural Network (73.2%). Medium Neural Network.</u>
  En este modelo, se obtienen: la matriz de confusión, la curva ROC y la matriz de TPR y FNR (ver Figura 34, Figura 35 y Figura 36, respectivamente).

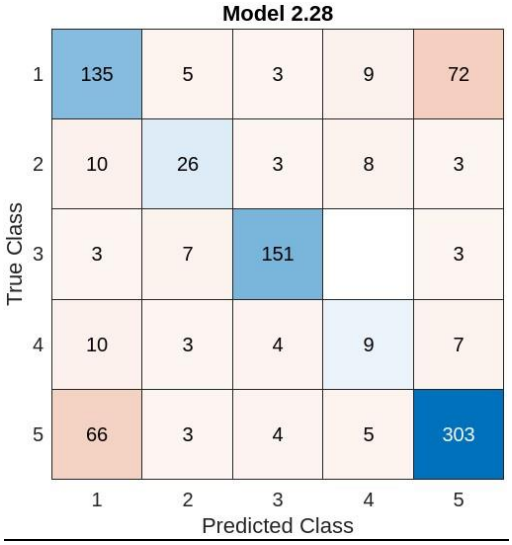

Figura 34. Matriz de Confusión NN.

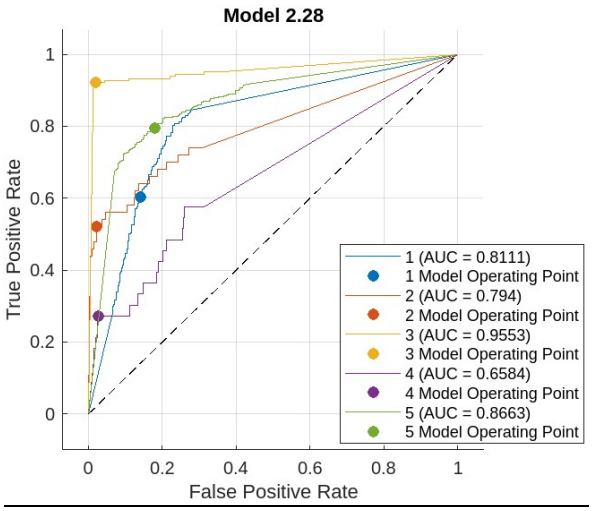

Figura 35. Curva Roc NN.

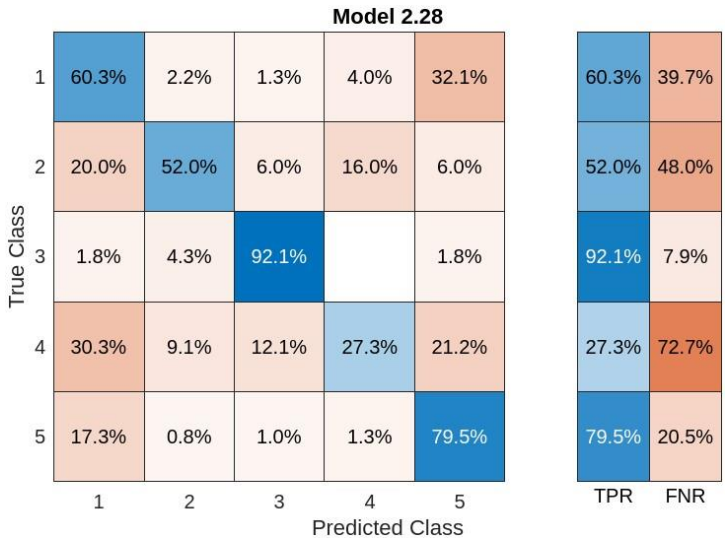

Figura 36. TPR y FNR de Neural Network.

Comparamos ahora los tres modelos seleccionados.

Visualizando las matrices y curvas ROC vemos como la clase 4 es la que da peores resultados, esto es debido a que esta categoría, que se corresponde con la emoción 'Enfado' es la que menos cantidad de muestras tiene, 40 imágenes en total hay en el dataset, por lo que es la clase con peor entrenamiento debido al bajo número de muestras. Esto también ocurre de forma parecida a la clase 2 ('Enfado'), compuesta por 58 muestras.

Por otra parte, las clases 1, 3 y 5 son las que dan mejores resultados, con mayores verdaderos positivos y altos AUC en las curvas ROC con.

En el modelo de redes neuronales, aunque consigue posicionar el punto de corte de la clase 4 más arriba que los otros dos, las curvas que presenta este modelo están muy próximas a la línea de no discriminación, lo que significa que las áreas bajo las curvas son menores. Por tanto, esto nos indica que tiene menos capacidad discriminativa que otros. Es por ello que descartaremos el uso de este modelo.

Por otro lado, si comparamos los modelos SVM y Ensemble, vemos que sus resultados son muy parecidos entre sí, con valores de AUC altos, por lo que podemos encontrar una mayor diferencia en los resultados de la categoría 4. Con el Ensemble se obtiene una tasa de verdaderos positivos mayor en esa clase, un 21.2%, mientras que con SVM se tiene un 15.2%, por tanto, aunque la tasa de verdaderos positivos en las tres primeras clases del SVM son mayores que en Ensemble, no son tan significativas como la que presenta la clase 4 en estos dos modelos.

Es por ello que, para esta prueba, donde se entrenaba modelos de clasificación todo el conjunto de muestras, es decir, sin ninguna clasificación de edad o género, y sin extracción de características, escogeremos el modelo Ensemble Subspace Discriminant.

7.2. Entrenamiento con selección de características.

Se repite el proceso anterior, pero incluyendo la clasificación de características.

Para ello, pulsamos en el botón 'Feature Selection' para poder elegir el algoritmo de extracción de características. Tenemos varios disponibles: MRMR, Chi2, ANOVA y Kruskall Wallis. Como se

explica en el apartado '7.2. Algoritmo de Mínima Redundancia y Máxima Relevancia', seleccionaremos el algoritmo MRMR (ver Figura 37).

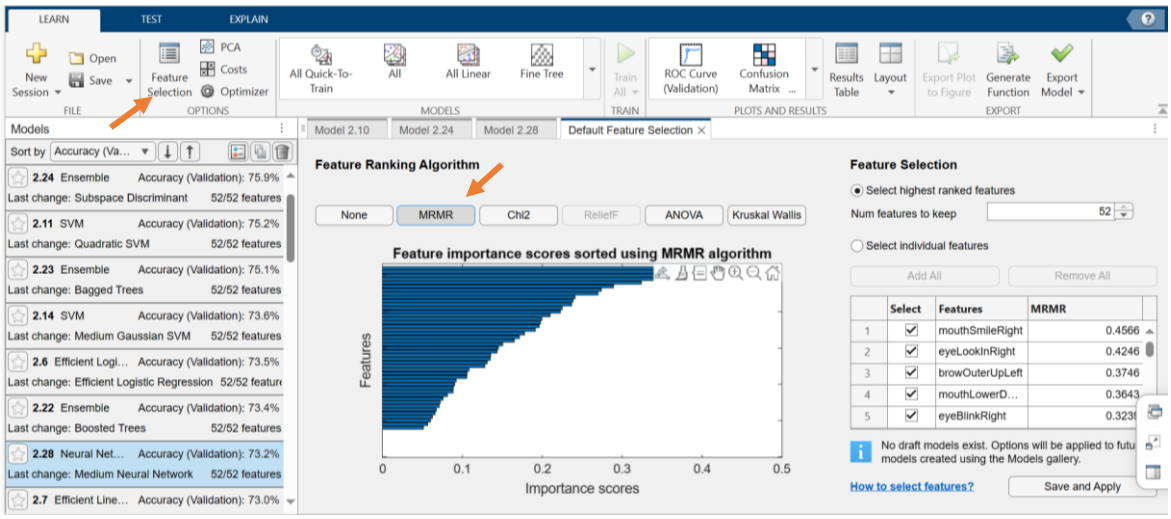

Figura 37. Aplicación de un algoritmo de selección de características específico, en concreto, MRMR.

Este algoritmo nos muestra una gráfica con las 52 características ordenadas de mayor a menor relevancia, puesto que el descenso de importancia se lleva a cabo de forma muy gradual, decidimos seleccionar aquellas que superen el valor de 0.15 en el gráfico, descartando las de valores inferiores.

El número de características se modifica en el apartado 'Num features to keep'. Disminuimos este número hasta que se hayan desmarcado las todas las características inferiores a 0.15 en el cuadro inferior.

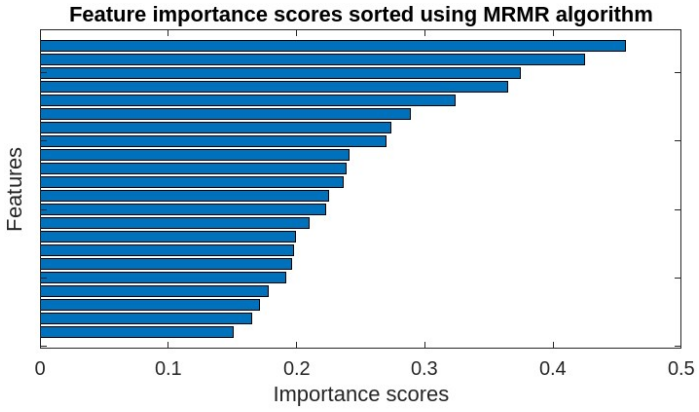

Figura 38. Características seleccionadas.

Puesto que esta partición la hemos hecho sobre el valor de la característica y no en función del número de características a seleccionar, nos queda un total de 22 datos para este caso. Este número de características que se seleccionen variará según los tipos de datos que tengamos.

Estas son las 22 características que el modelo ha seleccionado:

1. 'mouthSmileRight'
2. 'eyeLookInRight'
3. 'browOuterUpLeft'
4. 'mouthLowerDownLeft'

5. 'eyeBlinkRight'
6. 'mouthRollLower'
7. 'mouthUpperUpLeft'
8. 'eyeSquintRight'
9. 'mouthFrownRight'
10. 'mouthFunnel'
11. 'mouthStretchRight'
12. 'cheekPuff'
13. 'eyeWideLeft'
14. 'mouthSmileLeft'
15. 'mouthDimpleRight'
16. 'mouthUpperUpRight'
17. 'eyeSquintLeft'
18. 'mouthLowerDownRight'
19. 'brownInnerUp'
20. 'mouthRollUpper'
21. 'eyeBlinkLeft'
22. 'eyeLookInLeft'

Hecha la selección, pulsamos 'Save and Apply', y repetimos el proceso de elegir todos los modelos de clasificación y comenzamos el entrenamiento.
Comparamos las matrices de confusión y curvas de ROC de todos los modelos para decidir cuales dan mejores resultados, siendo estos los siguientes:

- SVM (73.1%). Linear SVM.
  En este caso, la Figura 39 muestra la matriz de confusión, la Figura 40 muestra la curva ROC y la Figura 41 muestra la matriz de TPR y FNR.

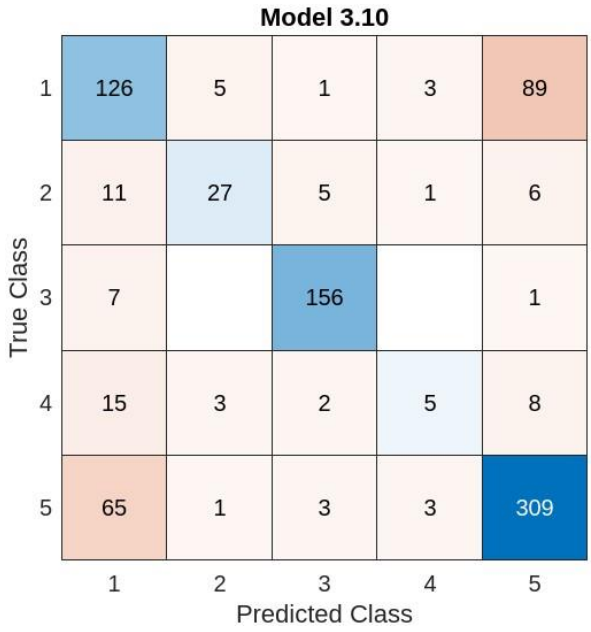

Figura 39. Matriz de confusión SVM.

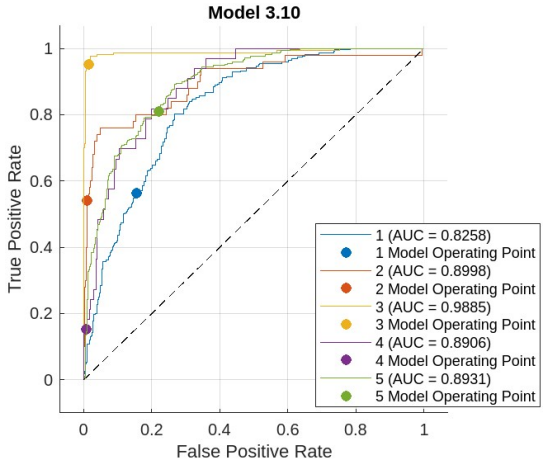

Figura 40. Curva ROC SVM.

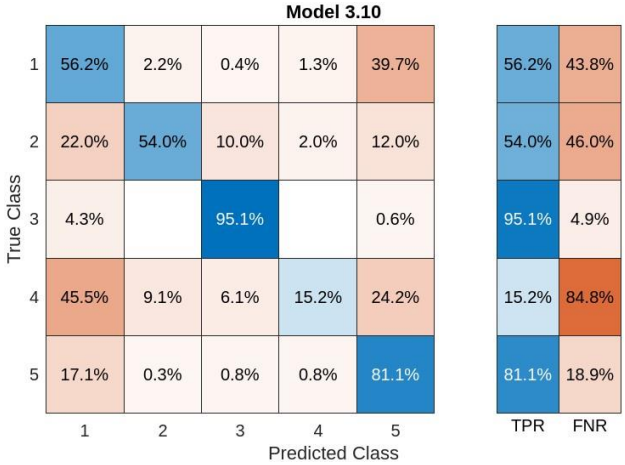

Figura 41. TPR y FNR de SVM.

- <u>Ensemble (72.4%). Boosted Trees.</u>
  En este caso, la Figura 42 muestra la matriz de confusión, la Figura 43 muestra la curva ROC y la Figura 44 muestra la matriz de TPR y FNR.

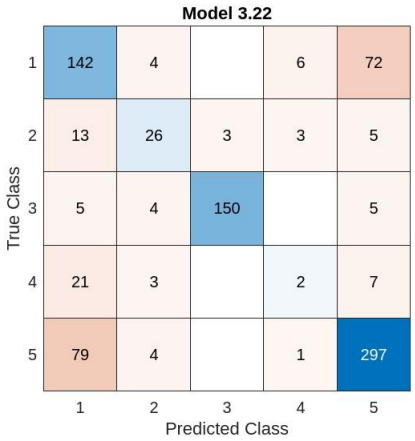

Figura 42. Matriz de confusión Ensemble.

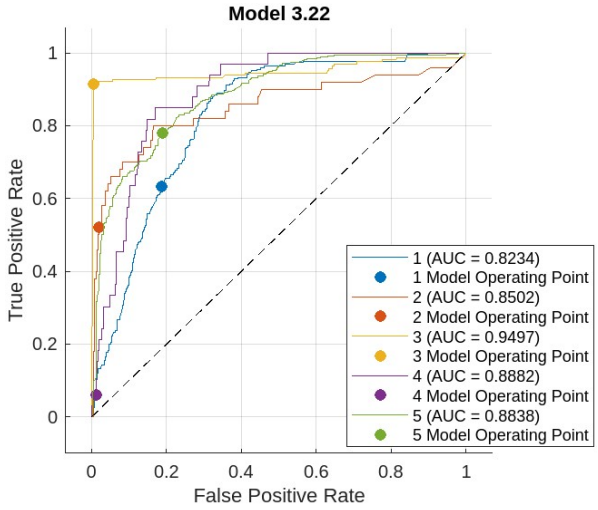

Figura 43. Curva Roc Ensemble.

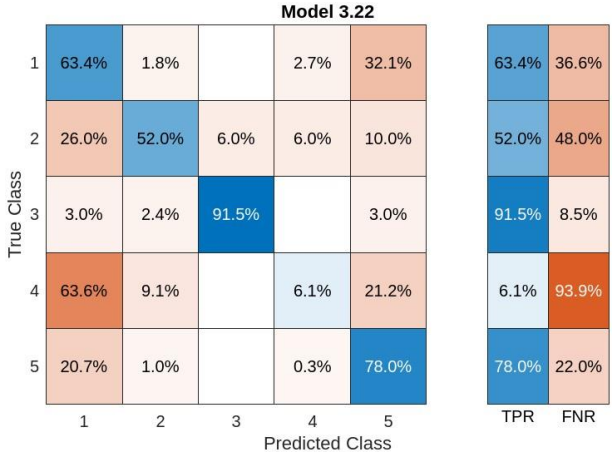

Figura 44. TPR y FNR Ensemble.

- <u>Linear Discriminant (71.5%).</u>
  En este caso, la Figura 45 muestra la matriz de confusión, la Figura 46 muestra la curva ROC y la Figura 47 muestra la matriz de TPR y FNR.

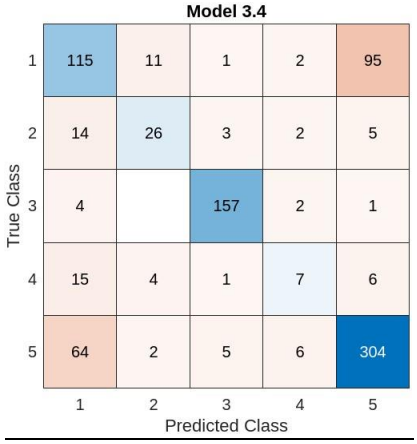

Figura 45. Matriz de confusión de LD.

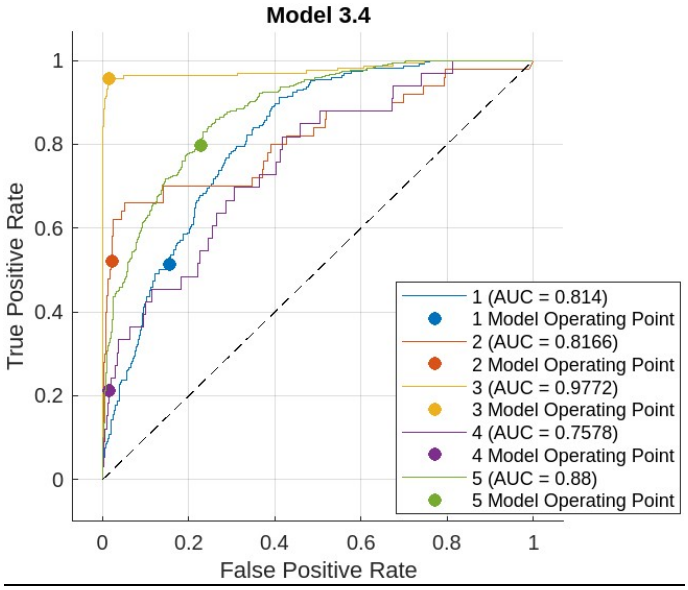

Figura 46. Curva ROC de LD.

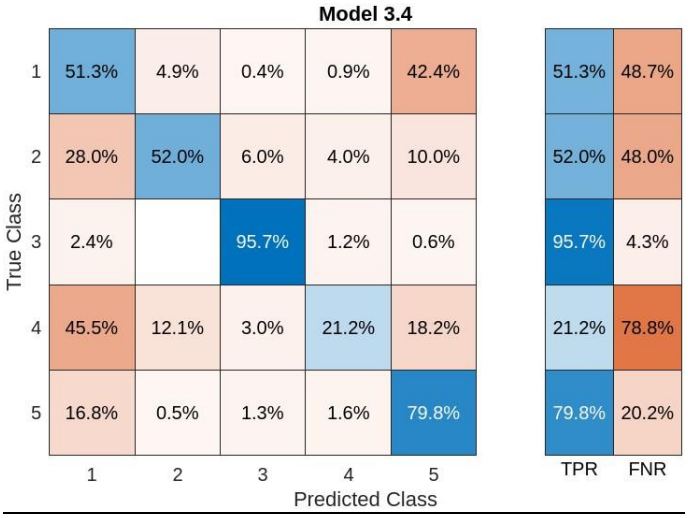

Figura 47. TPR y FNR de LD.

Podemos observar cómo los rendimientos de estos tres modelos son inferiores que los modelos sin extracción de características.

Por otra parte, las curvas ROC son similares en los tres, no obstante, es, de nuevo, en el punto de corte de la clase 4 donde se observa mayor diferencia. En el modelo Ensemble, este punto se encuentra muy próximo al [0,0], lo que quiere decir que el número de verdaderos positivos es mínimo. Esto también viene reflejado en la matriz de confusión, teniendo un 93.9% de tasa de falsos negativos. Descartaremos, entonces, el modelo Ensemble.

En cuanto a los dos modelos restantes, visualmente vemos que los cambios en las curvas ROC son muy pequeños, incluso en las tasas falsos negativos y verdaderos positivos tampoco se aprecia gran disparidad, es por ello que, al igual que sucedía en el apartado anterior, la diferencia es ligeramente superior en estas tasas de la categoría 4, aumentando de un 15.2% a un 21.2% en el modelo Linear Discriminant.

8. Resultados.

En el apartado anterior se ha descrito el proceso a seguir para llevar a cabo en entrenamiento, y su posterior comparación, de los modelos de clasificación. A continuación, repetiremos este mismo proceso para las distintas clasificaciones de edad y género que hemos aplicado en las pruebas. Teniendo en cuenta, también, el uso de todas las características y su selección.

8.1. Prueba 1. Conjunto de variables completo.

Este apartado corresponde con el proceso explicado anteriormente, donde hemos llegado a la conclusión de que para el entrenamiento con todas las características se ha seleccionado el modelo Ensemble Subspace Discriminant. Mientras que, en la segunda repetición, seleccionando 21 características, se escoge el modelo Linear Discriminant. En consecuencia, vamos a analizar los resultados de estos dos modelos para hallar cuál de los dos se ajusta mejor al tipo de datos que tenemos en esta primera prueba.

Lo primero que observamos al seleccionar los modelos es el porcentaje de precisión que presenta, teniendo el primero un 75.9% y un 71.5% el segundo. Esta pequeña diferencia también podemos observarla en las gráficas.

Visualizando las curvas ROC, la gráfica del SVM tiene valores de AUC ligeramente superiores y, además, la mayoría de sus puntos de corte también están levemente más arriba. Esto se ve reflejado a su vez, en las matrices de confusión, donde la categoría 4 está igualadas pero el resto cuenta con un mayor número de aciertos.

Por tanto, podemos determinar que, para este conjunto de datos, donde incluimos imágenes de hombres y mujeres de todas las edades, el mejor modelo para clasificar las emociones de los participantes es la Máquina de soporte vectorial sin extracción de características.
Además, aunque la diferencia es baja entre los dos clasificadores, podemos llegar a la conclusión de que cuando se tiene un dataset general, sin ninguna distinción ni clasificación de los participantes, puede ser conveniente incluir todas las características disponibles para el entrenamiento del modelo por dos razones:

- Tras ver la gráfica de la Figura 38 donde se ordenan las características de mayor a menor relevancia, vemos como esta disminución es progresiva y no hay un cambio abrupto en la importancia que presentan estas.
  Esto puede indicarnos que al ser un conjunto que, en cierta medida, intenta representar a la población, es más complicado encontrar características del rostro tan relevantes y concretas en todas las personas por igual.
  En cambio, en datasets más específicos, creados únicamente con mujeres u hombres, podríamos apreciar una gráfica de características que presente mayor diferencia entre los rasgos relevantes y los poco relevantes, para identificar emociones.

  Por lo tanto, no descartar ninguna característica, en este caso, puede suponer una mejora en la información que emplee el modelo clasificador para conseguir detectar las emociones, puesto que le proporciona más referencias para lograr a cabo esta tarea.

- Ligado al anterior, la estructura facial de las personas es muy dispar, pero, cuando observamos a una persona, identificamos si es hombre o mujer y estimamos su edad. Esto es gracias a que reconocemos, inconscientemente, aquellos rasgos comunes que presenta cada género y edad.
  Este mismo proceso no lo lleva a cabo un modelo de selección, por lo que puede ser una tarea más complicada conseguir identificar los rasgos característicos de forma común entre todas las personas.

  Por ello, la selección de características en datasets divididos y su posterior entrenamiento, puede llegar a simplificar el reconocimiento de emociones en personas.

8.2. Prueba 2. Clasificación por género: Hombre.

Repetimos el proceso selección de características y entrenamiento de modelos según el apartado '*8. Descripción del proceso de selección de características y entrenamiento*.' Señalando directamente que modelos se han escogido para los tipos de entrenamientos.

Recordemos que el archivo de datos 'Hombre.csv' es el que emplearemos para esta prueba, donde, tenemos las 52 variables Blendshape y la variable categórica de salida 'Emoción', con un total de 583 filas en el conjunto. Los datos presentes en este conjunto corresponden únicamente a hombres sin distinción de edad.

Entrenamiento sin selección de características.

Tras comparar los resultados de este entrenamiento, se opta por el modelo Medium Neural Network (78%). Siendo su matriz de confusión y curvas ROC las que se muestran en la Figura 48 y la Figura 49, respectivamente. Por su parte, la Figura 50 muestra la matriz de TPR y FNR.

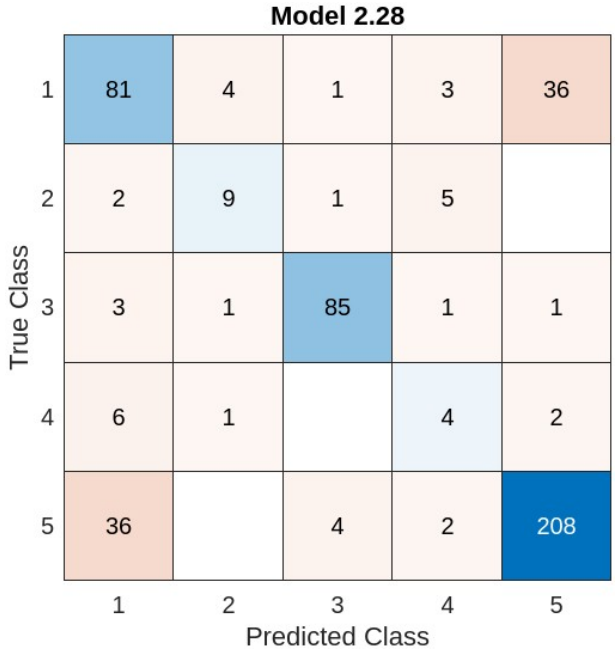

Figura 48. Matriz de confusión NN (P2).

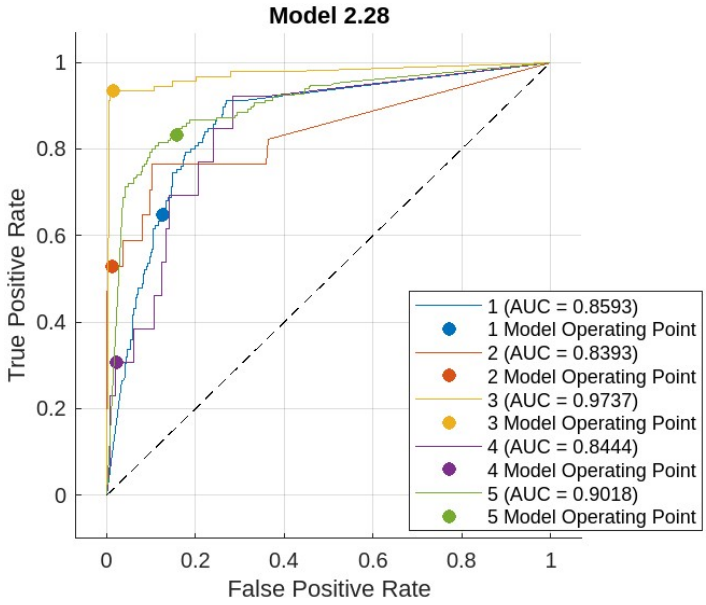

Figura 49. Curva ROC NN (P2).

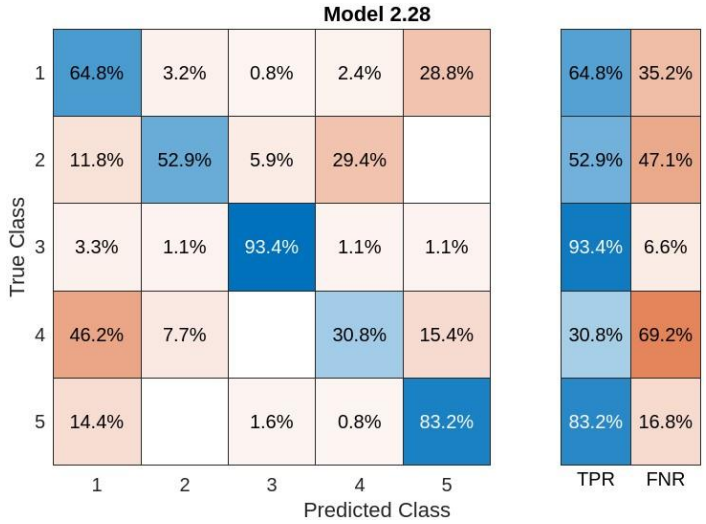

Figura 50. TPR y FNR de NN (P2).

Aunque este modelo no es el que mayor porcentaje de precisión presenta, consigue situar al punto de corte de la clase 4 en una posición más elevada que el resto, reflejándose esto en la tasa de verdaderos positivos, consiguiendo un porcentaje del 30.8% frente al 22% de otros modelos.

Entrenamiento con selección de características.

Aplicando el algoritmo MRMR en la en Classification Learner App, obtenemos la gráfica de los rasgos ordenados por relevancia (ver Figura 51).

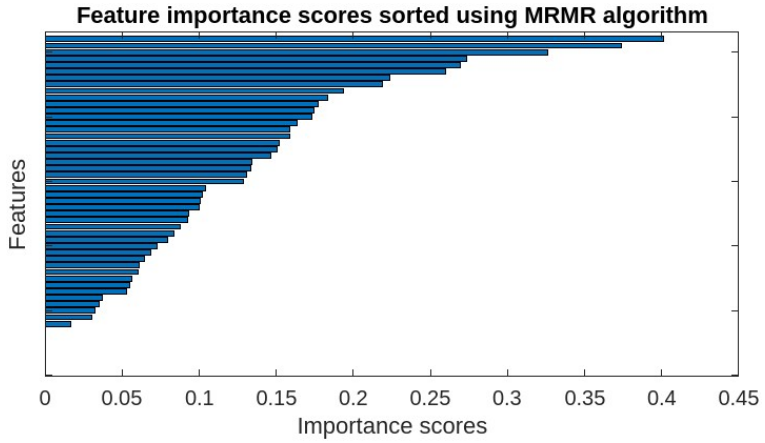

Figura 51. Gráfica MRMR prueba 2.

En esta gráfica podemos ver como el descenso de relevancia sigue siendo gradual, pero resaltan más las 8 primeras columnas en comparación con el resto. Si seguimos el mismo procedimiento anterior, donde establecíamos el umbral en el 0.15 y tomamos los valores superiores a este, obtendríamos 18 características que cumplen este requisito, resultando la gráfica de la Figura 52.

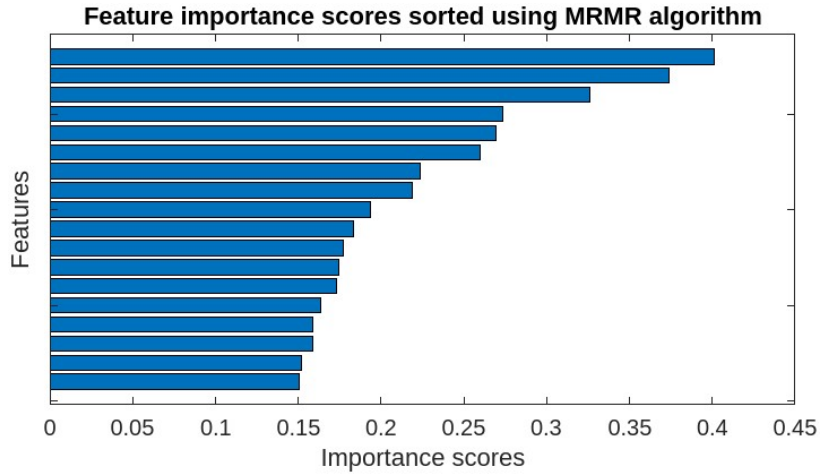

Figura 52. Gráfica MRMR 18 características.

El listado de las 18 características es:

1. 'mouthSmileRight'
2. 'eyeLookOutLeft'
3. 'browDownLeft'
4. 'eyeSquintRight'
5. 'eyeLookDownRight'
6. 'mouthShrugLower'
7. 'mouthUpperUpLeft'
8. 'eyeBlinkRight'
9. 'mouthLowerDownRight '
10. 'mouthFrownLeft'
11. 'mouthRollUpper '
12. 'mouthFunnel'
13. 'mouthSmileLeft '
14. ' eyeBlinkLeft '

15. ' brownInnerUp'
16. 'CheekPuff'
17. ' mouthUpperUpRight'
18. ' eyeLookOutRight'

El modelo escogido es el Ensemble Boosted Trees (76.2%). La Figura 53 muestra la matriz de confusión y la Figura 54 las curvas ROC asociadas a dicho modelo.

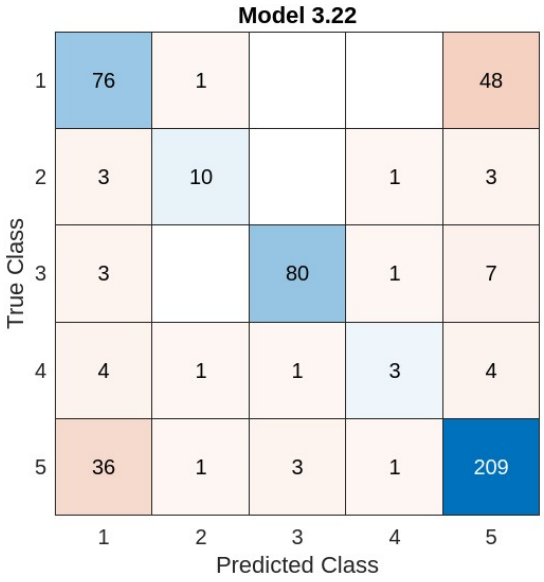

Figura 53.Matriz de confusión de Ensemble (P2).

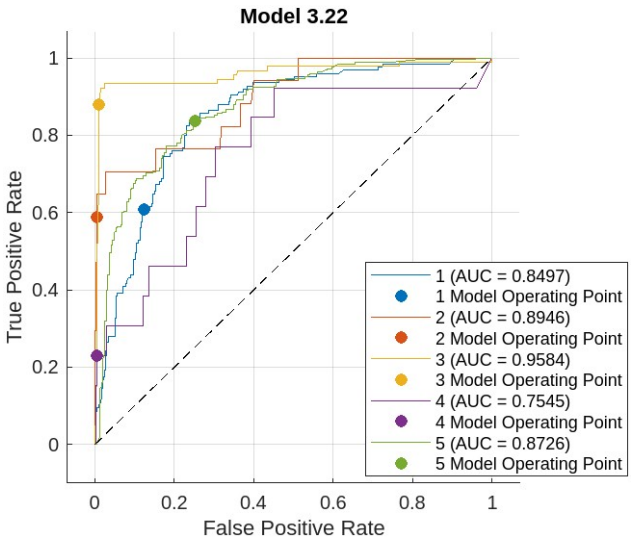

Figura 54. Curvas ROC Ensemble (P2).

Los dos modelos seleccionados ofrecen resultados muy parecidos entre sí, siendo el primero de ellos ligeramente más preciso.

En este caso no se encuentra tanta diferencia entre los dos entrenamientos, como ocurría en la prueba 1, con el conjunto de datos al completo.
Esto podemos relacionarlo con la uniformidad del dataset, al trabajar con datos exclusivamente de un género, la gráfica de representación de rasgos relevantes destaca menos estas

características, acentuando, por su parte, las tres primeras columnas que hacen referencia a: 'mouthSmileRight', 'eyeLookOutLeft' y 'browDownLeft'.
Por tanto, debido a la poca diversidad de los datos, las diferencias de los rasgos faciales entre los participantes puede no ser tan notoria como en la prueba 1, ya que sucedería lo comentado anteriormente: el proceso que realizamos las personas, inconscientemente, de identificar a los participantes como hombres o mujeres, es lo que hemos llevado a cabo con el conjunto de datos al hacer las clasificaciones previas.

Esto puede dar lugar a que haya una menor complejidad y dispersion de los rasgos faciales, por lo que la diferencia entre las características relevantes y poco relevantes no es tan pronunciada. En consecuencia, podemos observar que no hay una mejora significativa en los resultados de los modelos entrenados con todos los rasgos frente a los que emplean la clasificación de estos.

8.3. Prueba 3. Clasificación por género: Mujer.

Para esta prueba hacemos uso del archivo 'Mujer.csv' que detallamos en el apartado '*6. Creación del Dataset*'. En este archivo solo encontramos los datos correspondientes a las mujeres de todas las edades. Este archivo tiene un tamaño de 419 filas x 53 columnas.

Entrenamiento sin selección de características.

Siguiendo el criterio anterior, el modelo seleccionado en este entrenamiento es el Ensemble Subspace Discriminant (74.5%). La Figura 55 muestra la matriz de confusión y la Figura 56 las curvas ROC asociadas a dicho modelo.

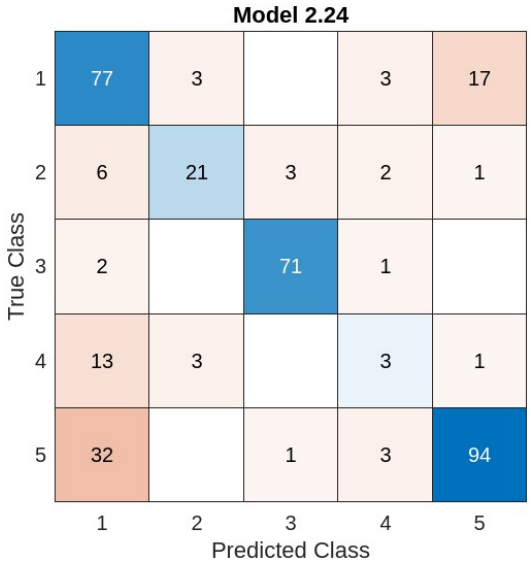

Figura 55. Matriz confusión Ensemble (P3)

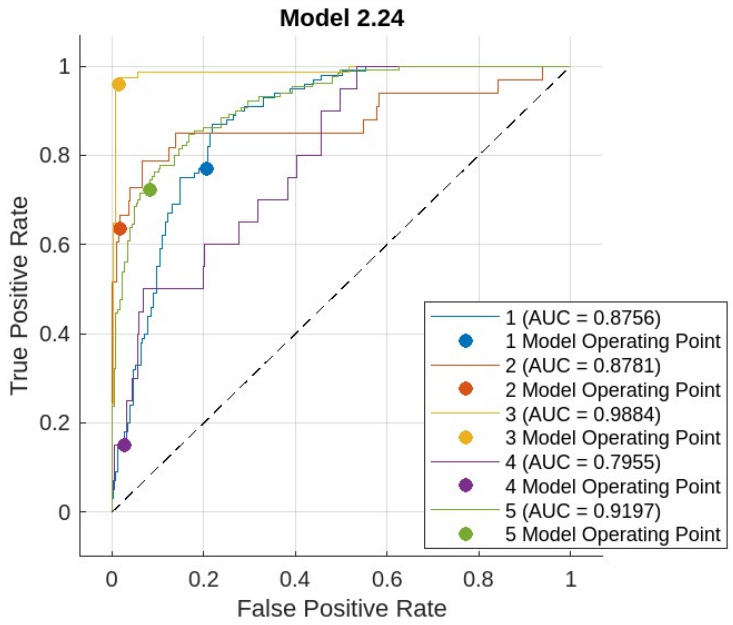

Figura 56. Curva ROC (P3).

Entrenamiento con selección de características.

Las características ordenadas quedan según se indica en la Figura 57.

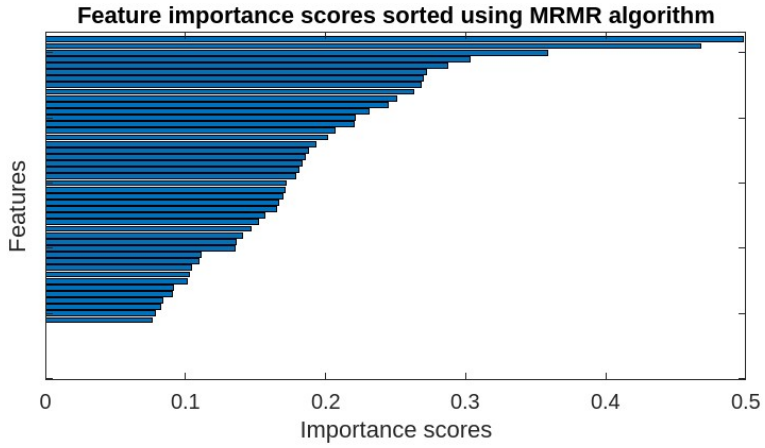

Figura 57. Gráfica MRMR prueba 3.

Con los datos de mujeres, el rasgo de mayor valor es 0.4985, muy próximo a 0.5, mientras que en el de los hombres, el equivalente a este sobrepasaba ligeramente el 0.4. Además, también se aprecia una gran diferencia en las 4 primeras columnas frente al resto, pertenecientes a: 'mouthSmileLeft', 'eyeWideLeft', 'browInnerUp' y 'mouthLowerDownLeft' (ver Figura 58).

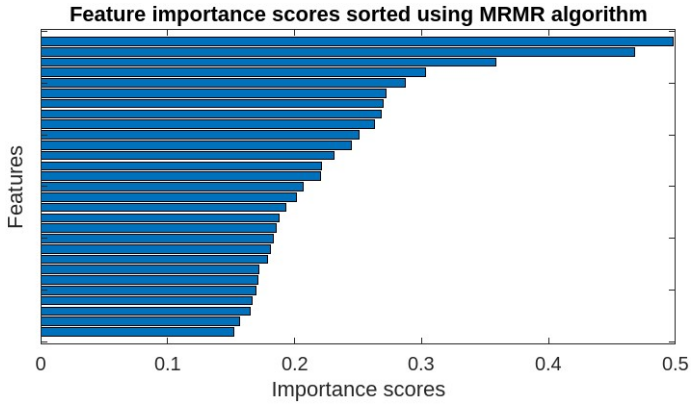

Figura 58. Gráfica MRMR 29 características.

Se obtienen las siguientes características aplicando el umbral de 0.15, seleccionando 29:

1. 'mouthSmileLeft'
2. 'eyeWideLeft'
3. 'brownInnerUp'
4. 'mouthLowerDownLeft'
5. 'mouthDimpleRight '
6. 'mouthUpperUpRight'
7. 'JawRight'
8. 'eyeSquintLeft'
9. 'mouthSretchRight'
10. 'eyeBlinkRight'
11. ' browOuterUpLeft'
12. 'mouthFunnel'
13. 'mouthRight'
14. 'mouthLowerDownRight '
15. 'mouthSmileRight'
16. 'eyeLookOutRight'
17. 'eyeBlinkLeft'
18. 'CheekPuff'
19. 'mouthUpperUpLeft'
20. 'browOuterUpRight'
21. 'mouthFrownRight'
22. 'eyeSquintRight'
23. 'mouthPressLeft'
24. 'mouthRollLower '
25. 'eyeLookDownLeft'
26. 'JawOpen'
27. 'mouthSretchLeft'
28. 'mouthShrugLower'
29. 'mouthPressRight'

Para el modelo seleccionado en Wide Neural Network (73.1%), tenemos los resultados que se observan en la Figura 59 y en la Figura 60.

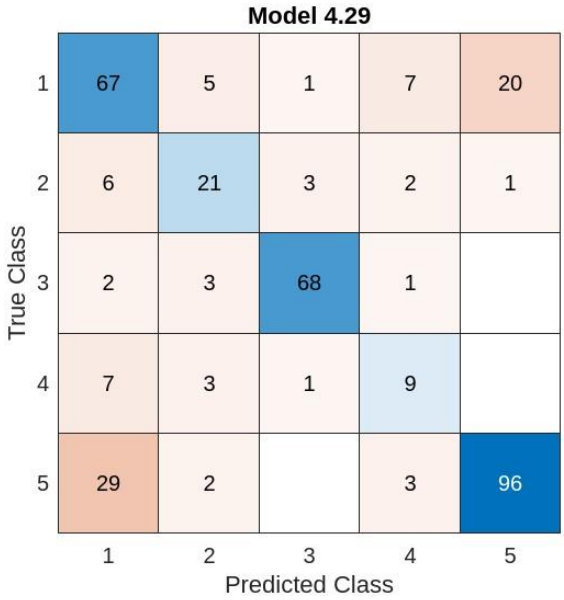

Figura 59. Matriz de confusión NN (P3)

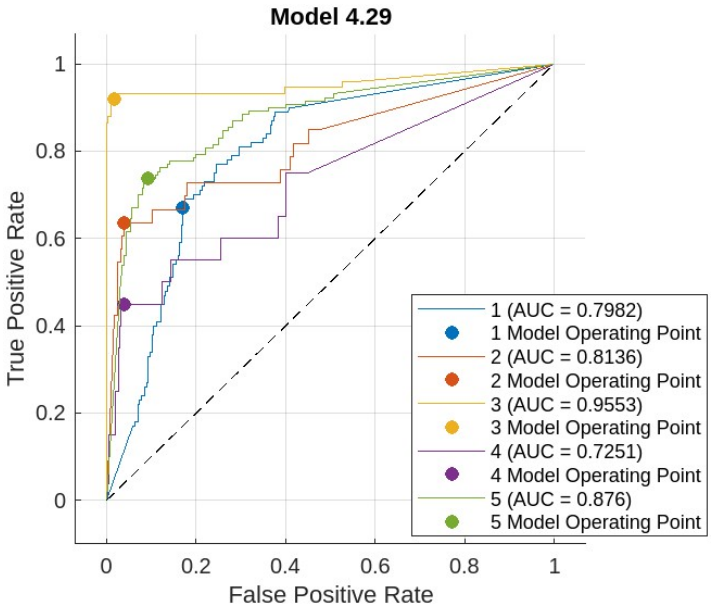

Figura 60. Curva ROC NN (P3)

En esta prueba 3, ocurre lo mismo que en que el apartado anterior.

Por tanto, con un conjunto específico de mujeres, el hecho de tomar todas las características o hacer una clasificación de estas, no supone una gran diferenciación en el rendimiento de los clasificadores.

8.4. Prueba 4. Clasificación por edad: Joven.

Tomaremos el archivo 'Joven.csv' para llevar a cabo esta prueba. Aquí se incluye tanto hombres como mujeres que hemos categorizado visualmente como jóvenes. Este está formado por 177 imágenes.

Entrenamiento sin selección de características.

Se selecciona el modelo Ensemble (77.5%) Bagged Trees y se obtienen los resultados que se muestran en la Figura 61 y en la Figura 62.

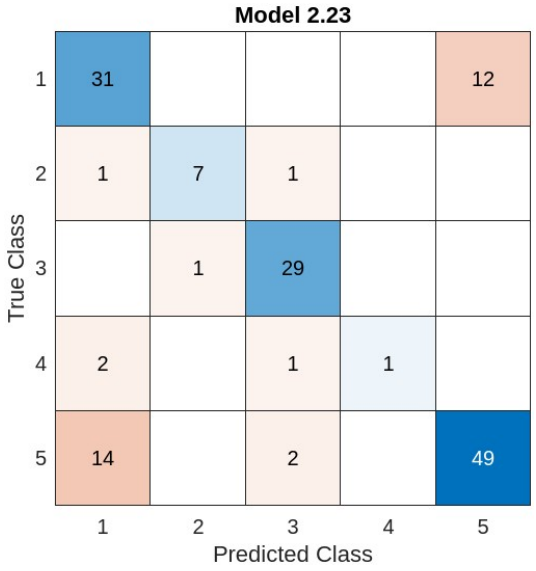

Figura 61. Matriz de confusión Ensemble (P4)

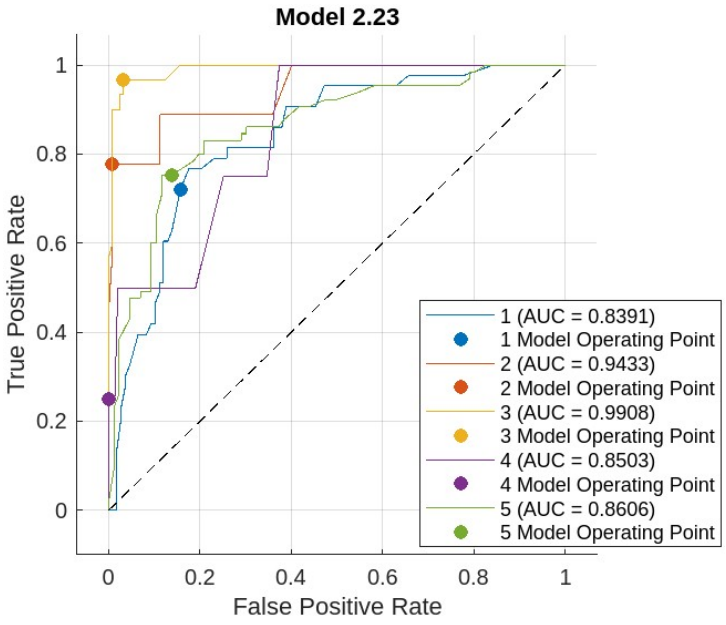

Figura 62. Curvas ROC Ensemble (P4)

Entrenamiento con selección de características.

Ordenamos las 52 características y aplicamos el mismo umbral de 0.15, obteniéndose la clasificación de la Figura 63 y observando específicamente 13 características en la Figura 64.

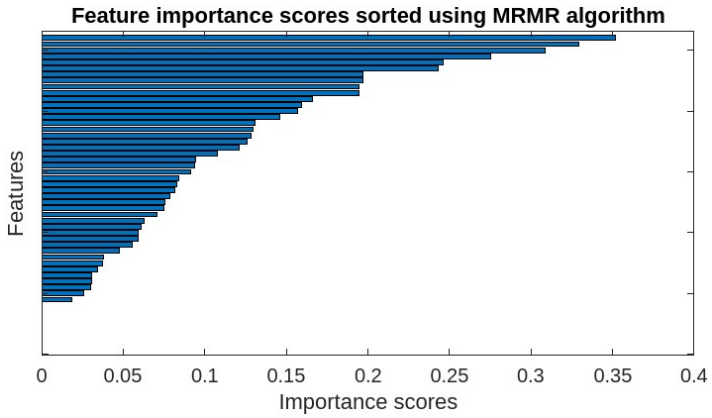

Figura 63. Gráfica MRMR (P4)

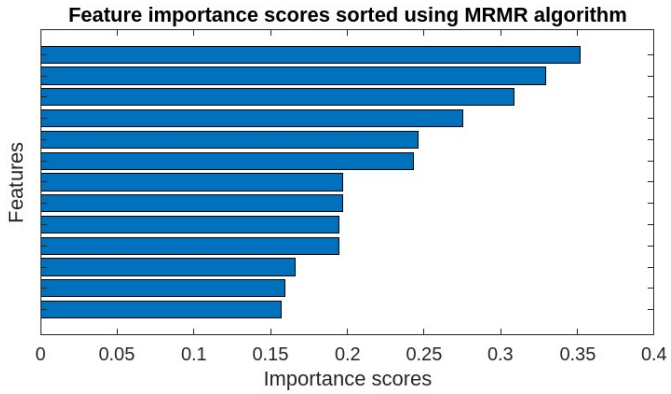

Figura 64. Gráfica MRMR 13 características.

Tras aplicar el umbral, esta vez nos han quedado solo 13 características, menos en comparación con las pruebas anteriores, donde obteníamos entre 20 y 30. Además, en este caso, el máximo valor no ha llegado a 0.4.

El listado de las seleccionadas es el siguiente:

1. 'mouthLowerDownRight '
2. 'eyeWideLeft'
3. 'eyeLookInRight'
4. 'eyeSquintLeft'
5. 'mouthSmileRight'
6. 'mouthRight'
7. 'mouthUpperUpLeft'
8. 'mouthFrownLeft'
9. 'mouthDimpleRight'
10. 'mouthFunnel'
11. 'eyeBlinkLeft'
12. 'mouthSmileLeft'
13. 'eyeSquintRight'

Realizamos el entrenamiento con esta selección y escogemos el modelo Ensemble (74.2%) Bagged Trees, cuyos resultados se muestran en la Figura 65 y en la Figura 66.

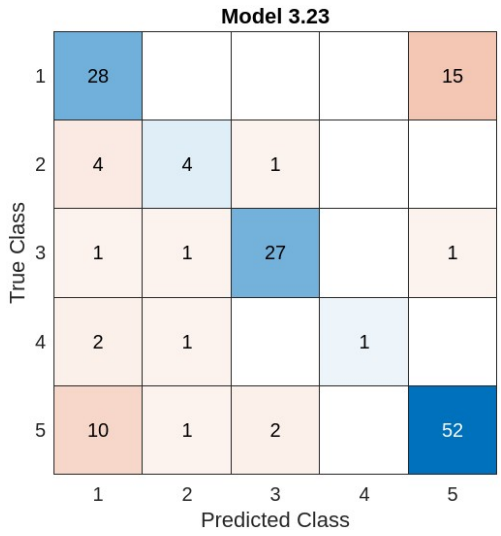

Figura 65. Matriz de confusión Ensemble (P4)

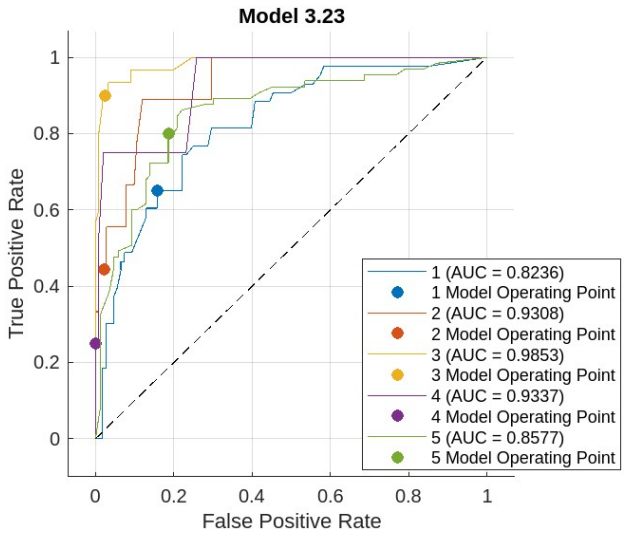

Figura 66. Curvas ROC Ensemble (P4)

Al igual que en las pruebas 2 y 3 la principal diferencia en los participantes era el género, ahora podemos expresar que la disparidad que tenemos en esta prueba respecto a las siguientes es la edad de los usuarios. Y, por tanto, hay rasgos faciales que tienen en común las personas en esta categoría.

Con el mismo ejemplo de antes, se ha efectuado una división según la apariencia de los participantes, refiriéndonos a la edad. Estas características propias de las personas jóvenes, como el aspecto de la piel con pocas o ninguna arruga, es lo que todos los participantes tienen en común.

8.5. Prueba 5. Clasificación por edad: Adulto.

Seguimos con la realización de la siguiente prueba con el archivo 'Adulto.csv', compuesto por 752 imágenes. Este es el más amplio en esta clasificación por edad.

Entrenamiento sin selección de características.

Elegimos el modelo Ensemble (76.2%) Subspace Discriminant y se obtienen los resultados que se muestran en la Figura 67 y en la Figura 68.

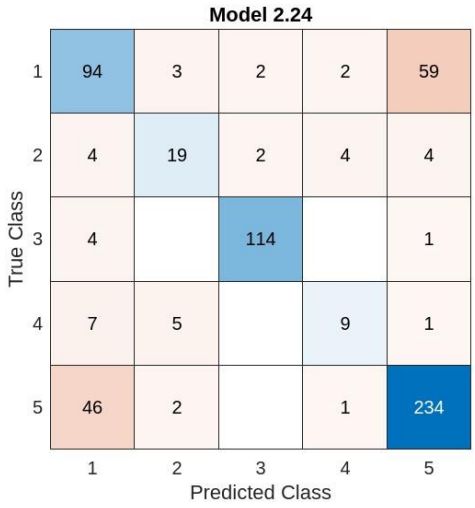

Figura 67. Matriz de confusión (P5)

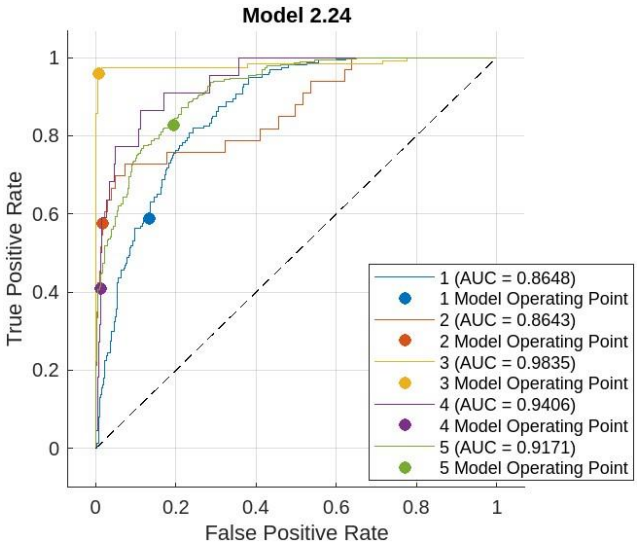

Figura 68. Curvas ROC (P5)

Entrenamiento con selección de características.

Realizamos la selección de los rasgos relevantes mediante un umbral del 0.15 tal y como muestra la Figura 69 y la Figura 70.

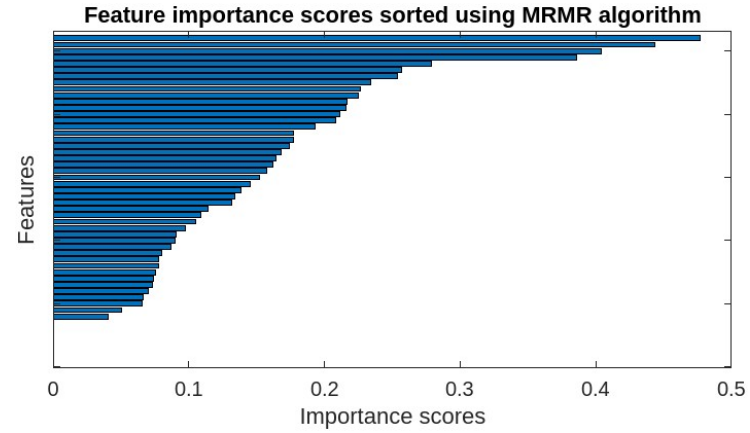

Figura 69. Gráfico del algoritmo MRMR

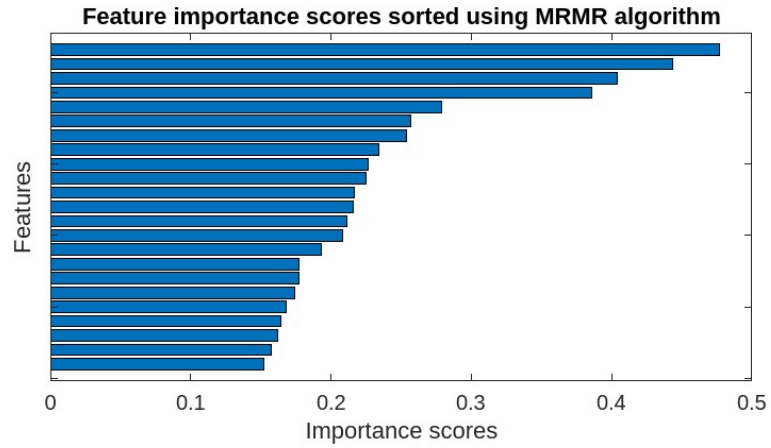

Figura 70. Gráfico MRMR 23 características.

El listado de características es:

1. 'mouthSmileLeft'
2. 'browDownRight'
3. 'eyeLookInRight'
4. 'eyeBlinkLeft'
5. 'mouthShrugLower'
6. 'mouthLowerDownLeft'
7. 'mouthRollUpper'
8. 'mouthUpperUpRight'
9. 'eyeSquintLeft'
10. 'mouthSretchRight'
11. 'mouthFunnel'
12. 'mouthSmileRight'
13. 'eyeSquintRight'
14. 'eyeLookDownLeft'
15. 'JawRight'
16. 'CheekPuff'
17. 'eyeWideLeft'
18. 'mouthUpperUpLeft'
19. 'mouthDimpleRight '
20. 'browOuterUpLeft'
21. 'mouthLowerDownRight'

22. 'eyeBlinkRight'
23. 'mouthRollLower '

Escogemos el modelo SVM (73.9%) Linear SVM y se obtienen los resultados de la Figura 71 y de la Figura 72.

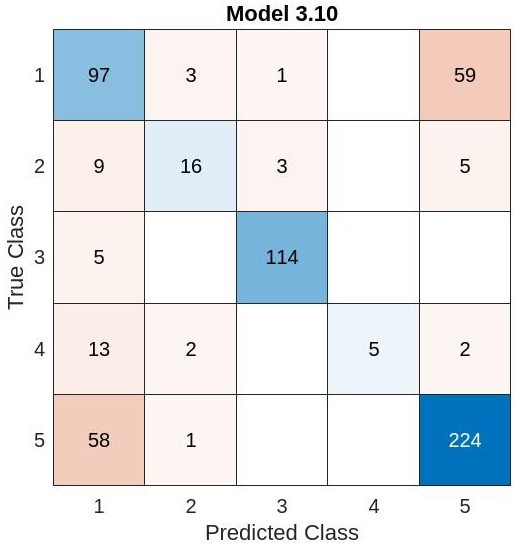

Figura 71. Matriz de confusión (P5)

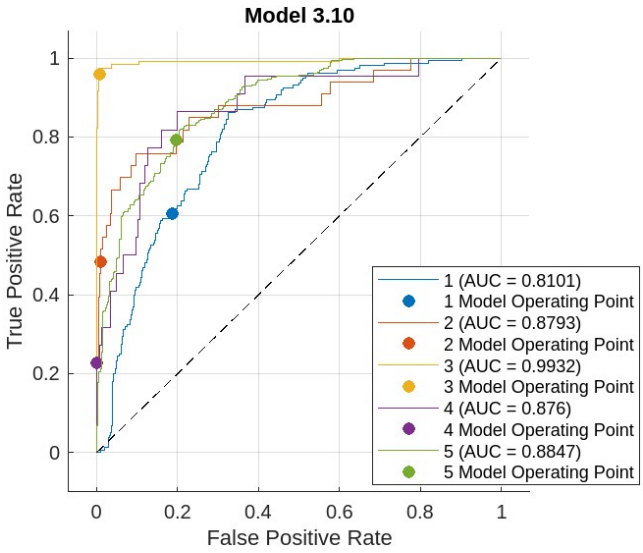

Figura 72. Curvas ROC (P5)

Comparando las curvas ROC, las correspondientes al primer modelo tiene mejores AUC y punto de corte ligeramente más elevados.

8.6. Prueba 6. Clasificación por edad: Mayor.

Por último, llegamos al conjunto de datos 'Mayor.csv' compuesto por 100 imágenes de personas mayores. Este es el conjunto más escaso debido a la falta de participantes correspondientes en esta categoría.

Entrenamiento sin selección de características.

Tomamos el modelo Ensemble (69.4%). Subspace KNN y se obtienen los resultados de la Figura 73 y de la Figura 74.

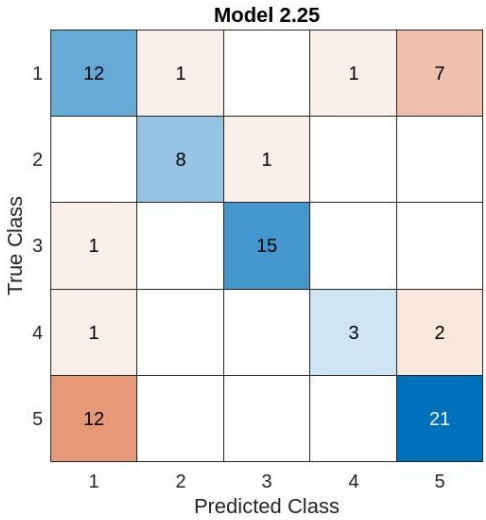

Figura 73. Matriz de confusión (P6)

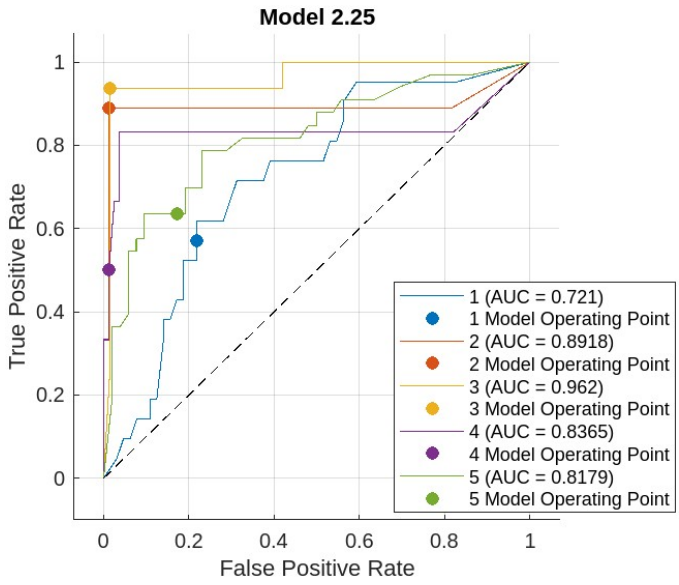

Figura 74. Curvas ROC (P6)

Entrenamiento con selección de características.

Aplicamos el mismo proceso de selección de características, con el umbral 0.15. No obstante, solo 3 rasgos superan este valor, tal y como se observa en la Figura 75 y en la Figura 76.

1. 'mouthLowerDownLeft'
2. 'eyeLookOutRight'
3. ' browInnerUp'

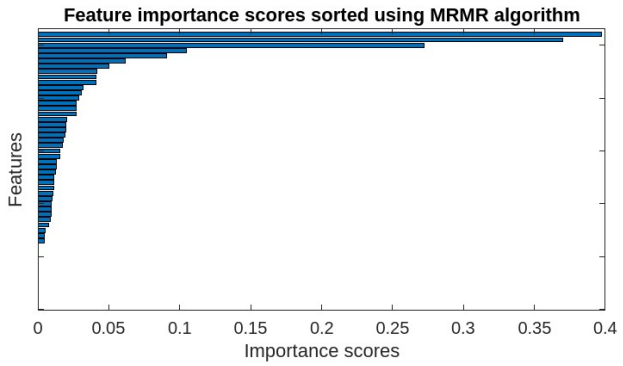

Figura 75. Gráfico MRMR

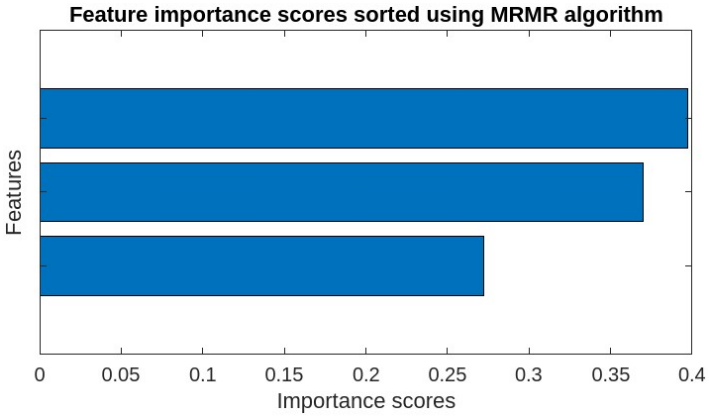

Figura 76. Gráfico MRMR 3 características.

Tomamos el modelo Tree (54.1%). Fine Tree y se obtienen los resultados de la Figura 77 y de la Figura 78.

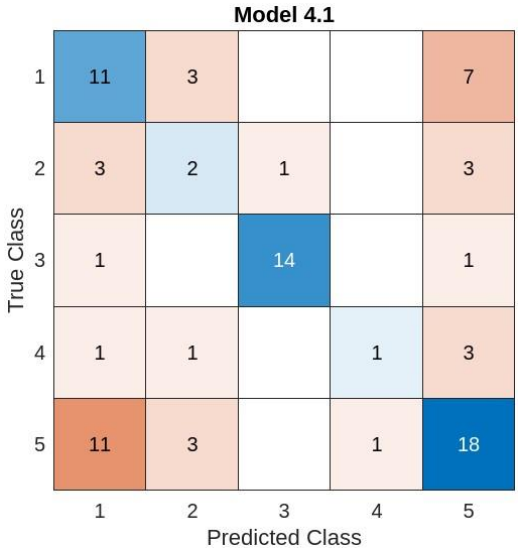

Figura 77. Matriz de confusión (P6)

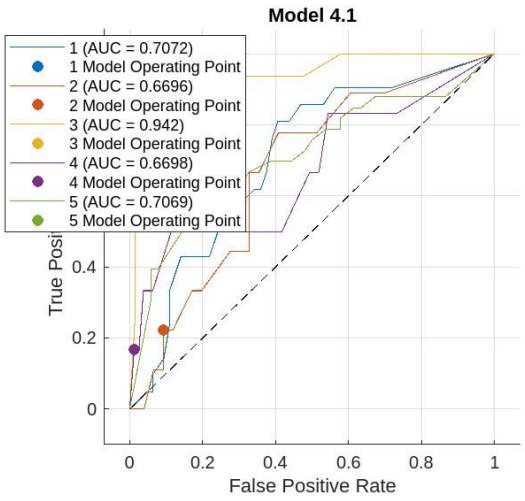

Figura 78. Curvas ROC (P6)

Como el resultado de este modelo no es tan bueno debido a que solo dispone de tres características para realizar la clasificación de emociones, vamos a realizar otro entrenamiento en el que hemos disminuido el umbral a 0.04 para obtener 10 características (ver Figura 79).

1. 'mouthLowerDownLeft'
2. 'eyeLookOutRight'
3. 'browInnerUp'
4. 'mouthPressLeft'
5. 'eyeBlinkRight'
6. 'JawOpen'
7. 'mouthSretchLeft'
8. 'eyeSquintRight'
9. 'mouthLowerDownRight'
10. 'mouthRollLower'

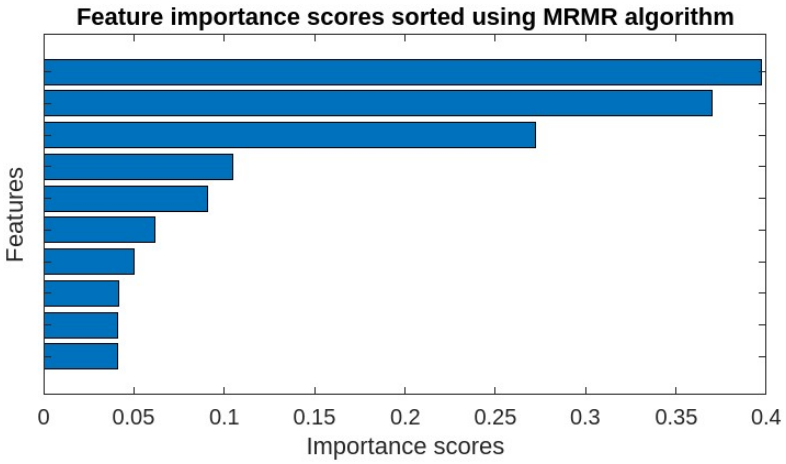

Figura 79. Gráfica MRMR 10 características.

Y seleccionamos el modelo Ensemble (64.7%). RUSBoosted Trees, obteniéndose los resultados mostrados en la Figura 80 y en la Figura 81.

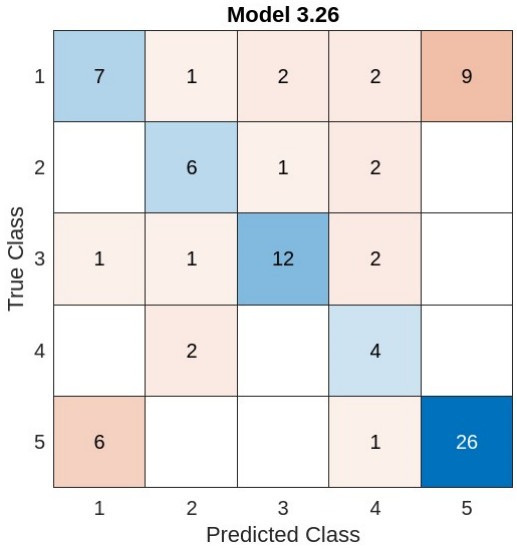

Figura 80.Matriz de confusión (P6)

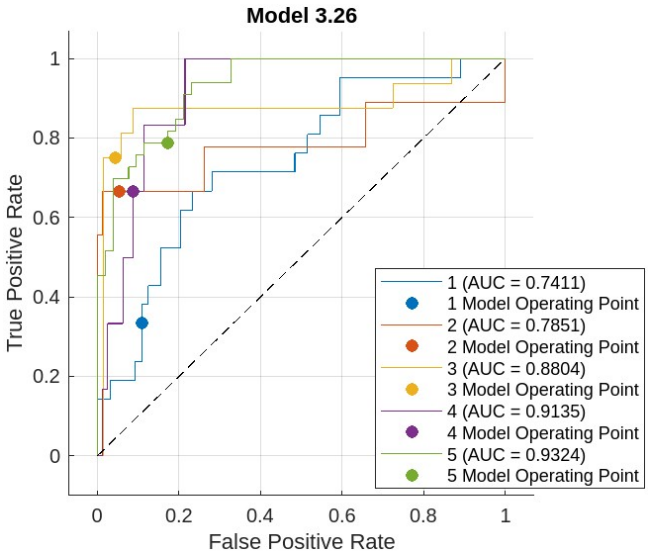

Figura 81. Curvas ROC (P6)

Este conjunto es el más reducido de todos, por lo que encontramos limitaciones en este aspecto. Tener un número pequeño de muestras afecta al rendimiento del modelo, puesto que puede limitar la capacidad de reconocimiento de patrones complejos y representativos. En consecuencia, será menos preciso a la hora de clasificar las emociones en imágenes con personas que ''no ha visto'' previamente.

Por otra parte, el uso de únicamente 3 características relevantes ha dado como resultado curvas ROC muy próximas a la diagonal, incluso sobrepasándola en un tramo. Esto se ocasiona porque, al limitar tanto el número de características, se produce una pérdida significativa de información que el modelo necesita para aprender a detectar los patrones comunes.
Al aumentar este número a 10, vemos una mejora en el rendimiento del modelo, ya que consigue capturar una gama más amplia de información de los datos de muestra, permitiéndole realizar predicciones más acertadas sobre las clases 2, 4 y 5.

8.7. Comparación de características relevantes.

Podemos comparar las características relevantes que hemos obtenido en cada prueba según la distinción de los datos y comprender, así, que aspectos faciales son los más importantes para los modelos de clasificación.
Para ello, vamos a tomar las 10 primeras características de cada prueba, obteniendo:

- Género: Hombres y Mujeres (ver Figura 82).

Hombre	Mujer
mouthSmileRight	mouthSmileLeft
eyeLookOutLeft	eyeWideLeft
browDownLeft	brownInnerUp
eyeSquintRight	mouthLowerDownLeft
eyeLookDownRight	mouthDimpleRight
mouthShrugLower	mouthUpperUpRight
mouthUpperUpLeft	jawOpen
eyeBlinkRight	eyeSquintLeft
mouthLowerDownRight	mouthSretchRight
mouthFrownLeft	eyeBlinkRight

Figura 82. Tabla comparación por género.

Vemos que, en el caso de las mujeres, el algoritmo toma las características correspondientes a la boca como más relevantes, mientras que, para los hombres, las referentes a la boca y a los ojos están más igualadas.
Esto podría deberse a distintas razones como, evidentemente, las diferencias en la estructura facial de los hombres y las mujeres, que puede influir en la forma de expresar las emociones ambos géneros y, por otra parte, los aspectos culturales, es decir, culturalmente, en comparación con los hombres, las mujeres tienden a expresar más las emociones y puede que a la hora de mostrarlo, se enfatice la expresión, en mayor medida, a través del gesto de la boca.

- Edad (ver Figura 83):

Joven	Adulto	Mayor
mouthLowerDownRight	mouthSmileLeft	mouthLowerDownLeft
eyeWideLeft	browDownRight	eyeLookOutRight
eyeLookInRight	eyeLookInRight	browInnerUp
eyeSquintLeft	eyeBlinkLeft	mouthPressLeft
mouthSmileRight	mouthShrugLower	eyeBlinkRight
mouthRight	mouthLowerDownLeft	JawOpen
mouthUpperUpLeft	mouthRollUpper	mouthSretchLeft
mouthFrownLeft	mouthRollUpper	eyeSquintRight
mouthDimpleRight	eyeSquintLeft	mouthLowerDownRight
mouthFunnel	mouthSretchRight	mouthRollLower

Figura 83. Tabla comparación por edad.

En esta comparación podemos ver como en la categoría Mayor encontramos más características que hacen referencia al término 'Lower Down', es decir, 'hacia abajo' en comparación con las otras dos, y en cuanto a los ojos, la dos últimas categorías comparten los términos 'Blink' y 'Squint', que hacen referencia al parpadeo de los ojos y la posición entrecerrada de estos.
Es por ello que, en estas características, se ve reflejada la estructura facial según la edad de los participantes, ya que, a una edad avanzada, es más evidente el cambio del rostro con la aparición de arrugas y perdida de firmeza en la piel.

Apreciamos lo contrario en la categoría Joven, donde aparecen más términos como 'Wide' o 'Upper', que significan 'hacia arriba' o 'amplio'.

9. Conclusiones y trabajo futuro.

Tras la realización de las 6 pruebas, con el uso de diferentes conjuntos de datos, llevaremos a cabo un análisis de los resultados que hemos obtenido.

Como ya hemos comentado, el reconocimiento de emociones es una tarea muy complicada para los modelos de clasificación e incluso también supone una dificultad para las personas.
Esto se debe a la gran variedad de rostros, que supone un obstáculo importante en la tarea de clasificación, en consecuencia, los modelos deben adaptarse a un gran número de características para conseguir predicciones precisas.

Un tema que hemos tratado durante el trabajo es la extracción de características, que se emplea con el objetivo de reducir el coste computacional, al trabajar con conjuntos más pequeños, y eliminar el posible ruido que se pueda aplicar durante el proceso de detección, ya que se consigue suprimir aquellas características que no aporten suficiente información para la detección de patrones.

No obstante, en la mayoría de las pruebas no ha habido una mejora significativa en el rendimiento del modelo al usar una selección de características ya que, como hemos comentado, en esta tarea es posible que, el mayor número de información que podamos ofrecer a los modelos de clasificación ayude al buen rendimiento de este.
Esto lo vemos reflejado en la prueba 1, con un dataset muy general, sin embargo, este planteamiento puede cambiar si modificamos el enfoque de la clasificación, es decir, si buscamos especializar a un detector en reconocer únicamente una emoción, el número de características podría reducirse.
Por ejemplo, si quisiéramos detectar la felicidad, sería conveniente no tomar las 52 características de Blendshape y solo incluir aquellas que hacen referencia a la boca, ya que se asociaría la felicidad con una sonrisa en las personas.

Por otra parte, se ha visto que se obtienen distintas características relevantes en hombres y mujeres, por lo que la distinción por género también puede repercutir en el rendimiento del modelo.

Respecto a los modelos seleccionados en cada prueba, el que más se ha repetido ha sido el Ensemble que, como ya hemos visto anteriormente, este se forma gracias a la combinación de distintos modelos de clasificación con el objetivo de mejorar el rendimiento final. No obstante, no podemos asegurar que este modelo sea el mejor para el reconocimiento de emociones, ya que dependerá de otros factores como la naturaleza del conjunto de datos.

En cuanto a los conjuntos de datos, afectará positivamente que esté compuesto por una gran variedad de imágenes con el objetivo de intentar representar lo máximo posible a la población, dentro de la medida de lo posible, ya que un dataset de estas características conseguirá disminuir el sesgo del conjunto y aumentar la precisión del modelo.
Asimismo, es igual de importante conseguir un número equilibrado entre las distintas emociones, puesto que, si solo se tiene una pequeña cantidad de alguna clase, como nos ocurría para la categoría 4, será más difícil identificar esa emoción y podría confundir, por ejemplo, las expresiones de enfado con neutralidad, sobre todo, si éstas no son posadas.

Finalmente, aunque la detección de emociones sea una tarea complicada para los modelos de clasificación, e incluso para las personas, con este trabajo nos hemos aproximado un poco a este objetivo empleando los coeficientes Blendshape gracias a las soluciones ofrecidas por MediaPipe, en vez de imágenes directamente.

Este trabajo, de hecho, no es el final del camino, sino más bien el principio, ya que se podría ampliar utilizando un conjunto de datos mucho mayor, entrenando los modelos de forma personalizada, esto es, si se utiliza un reconocedor de la persona, se podría aplicar un reconocedor de gestos (que definen emociones), específico para dicha persona.

En el contexto de identificación de emociones en personas mayores, el trabajo es relevante, ya que estimar automáticamente el estado de ánimo a través de la expresividad del rostro de dichas personas puede ser interesante desde el punto de vista de la salud. Así, los terapeutas ocupacionales, auxiliares, cuidadores, familiares, psicólogos y personal sanitario, en general, que habitualmente trabajan en centros de día o residencias, podrían utilizar esta información para llevar a cabo tareas de prevención de trastornos que suelen afectar a esta edad: ansiedad, depresión, desnutrición o falta de hidratación, disminución de los niveles de ciertos elementos esenciales por el seguimiento de una dieta pobre o falta de estimulación cognitiva, entre otros.

La identificación de las personas por parte, por ejemplo, de un robot asistencial y la evaluación del estado de ánimo a través de un sistema de reconocimiento facial, a través de una herramienta como MediaPipe, es interesante, ya que no se requiere el almacenamiento de la fotografía de la persona, sino el conjunto de landmarks (lo que protege la privacidad, pues no permite la identificación directa del individuo). Este aspecto es de especial relevancia, desde el punto de vista de la seguridad del sistema, ya que protege la identidad de los usuarios. Además, MediaPipe ofrece soluciones bien probadas, con una ficha clara de los procesos de validación que se han llevado a cabo en la implementación de cada solución. Los modelos de clasificación obtenidos a partir de estas entradas de landmarks podrían ser “reentrenables”, si se consiguen nuevas muestras en los datasets, aplicando técnicas conocidas de Machine Learning.

Por otra parte, MediaPipe puede usarse con imágenes, vídeos y cámaras convencionales, por lo que el uso de un sensor RGB-D es completamente factible como fuente de datos. Una cámara de este tipo, además, permitiría diferenciar si la persona es real o se trata de una fotografía, simplemente comprobando la nube de puntos generada.

En resumen, este trabajo se puede extender:

- Extendiendo y mejorando los datasets adaptándolos a su uso con personas mayores.
- Aplicando modelos específicos para cada individuo si se obtienen suficientes patrones para cada conjunto de gestos que suelen definir la expresión de una emoción.
- Combinando los resultados de los modelos con la información proveniente de otros sensores, como por ejemplo, interfaces cerebrales, que permitan estimar el grado de atención o de estrés, entre otros.
- Migrando los algoritmos escritos en Matlab a otros lenguajes para que el sistema pueda ser usado, en tiempo real, en un robot asistencial.

Bibliografía

Abdollahi, H., Mahoor, M. H., Zandie, R., Siewierski, J., & & Qualls, S. H. (2023). *"Artificial Emotional Intelligence in Socially Assistive Robots for Older Adults: A Pilot Study".* doi:https://doi.org/10.1109/taffc.2022.3143803

Arbeloa, G. B. (2018). *Implementación del algoritmo de los k vecinos más cercanos (k-NN) y estimación del mejor valor local de k para su cálculo*. Obtenido de Unavarra.es: https://academica-e.unavarra.es/xmlui/bitstream/handle/2454/29112/Memoria.pdf?sequence=2

Arias, E. R. (2021). *Variable categórica*. Obtenido de Economipedia: https://economipedia.com/definiciones/variable-categorica.html

Bequir, S. (06 de 05 de 2022). *Residencia Argaluza*. Obtenido de El aislamiento social en el adulto mayor: nuestra gran batalla: https://residencia-argaluza.com/blog/el-aislamiento-social-en-el-adulto-mayor-nuestra-gran-batalla/

Bisogni, C., Cimmino, L., Marsico, M. D., Hao, F., & Narducci, F. (2023). *Emotion recognition at a distance: The robustness of machine learning based on hand-crafted facial features vs deep learning models,*. 104724, ISSN 0262-8856. doi:https://doi.org/10.1016/j.imavis.2023.104724.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees.* Biometrics, 40, 874. Obtenido de https://api.semanticscholar.org/CorpusID:29458883

Cerda, J., & Cifuentes, L. (2012). *Uso de curvas ROC en investigación clínica: Aspectos teórico-prácticos.* doi:10.4067/s0716-10182012000200003

Cereceda, S. P. (2010). *Reconocimiento de emociones: estudio neurocoginitivo.* Praxis: revista de psicología, (18), 29.

Conde, M. (2022). *Machine Learning Ensembles*. Obtenido de Data Science: https://www.es100x100datascience.com/2017/02/22/machine-learning-ensembles-i/

Domínguez, T. (28 de 04 de 2022). *Trainontech*. Obtenido de Deep Learning o IA vs Visión artificial clásica: https://trainontech.com/deep-learning-o-ia-vs-vision-artificial-clasica/

EKCIT, T. E. (5 de 4 de 2021). *TIC PORTAL*. Obtenido de Deep learning: ¿se pueden programar las máquinas para pensar como humanos?: https://www.ticportal.es/glosario-tic/deep-learning-dl

FaceMesh. (s.f.). Obtenido de mediapipe: https://chuoling.github.io/mediapipe/solutions/face_mesh.html

Franco. (10 de 09 de 2023). *Redes Neuronales: Aprendizaje y Resolución de Problemas*. Obtenido de ThePower Business School: https://www.thepowermba.com/es/blog/redes-neuronales

González, F. V. (2022). *Métodos de Selección de Variables en Modelos de Regresión.* Universidad de Santiago de Compostela. Obtenido de

https://minerva.usc.es/xmlui/bitstream/handle/10347/30047/2021_TFG_Matem%c3%a1ticas_Vicente_Gonz%c3%a1lez_Variables.pdf?sequence=1

Gonzalez, J. L. (13 de 7 de 2020). *SoldAI*. Obtenido de Tipos de aprendizaje automático: https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2

Gonzalez, L. (21 de 01 de 2020). *Diferencia entre Machine Learning y la Programación Tradicional*. Obtenido de Aprende IA: https://aprendeia.com/diferencia-entre-machine-learning-y-la-programacion-tradicional/

Grisales, J. A. (12 de 10 de 2022). *Geostrategy*. Obtenido de Visión artificial en la medicina: https://www.geostrategydata.com/vision-artificial-en-la-medicina/

IBM. (s.f.). *¿Qué es un árbol de decisión?* Obtenido de IBM: https://www.ibm.com/es-es/topics/decision-trees

Jiménez, A. (22 de 06 de 2022). *¿ Por que usar python para machine learning ?* Obtenido de El Blog de Python: https://elblogpython.com/machine-learning/por-que-python-para-machine-learning/

Josep. (2023). *Máquinas de soporte vectorial: la guía definitiva.* Obtenido de Conectando ideas: https://conectandoideas.net/maquinas-de-soporte-vectorial/

León, E. C. (2016). *Introducción a las máquinas de vector soporte (SVM) en aprendizaje supervisado.* Obtenido de Universidad Zaragoza: https://zaguan.unizar.es/record/59156/files/TAZ-TFG-2016-2057.pdf

López Barbosa, R. R. (2015). *Application of Sentiment Analysis to Data Generated in Social Media*. Obtenido de 1Library.co: https://1library.co/article/stanford-corenlp-rntn-an%C3%A1lisis-de-sentimientos.1y97rprq

Lozano, R. G. (25 de 11 de 2020). *"Sistema de Reconocimiento Facial con Deep Learning".* Obtenido de Universidad Carlos III Madrid: https://e-archivo.uc3m.es/bitstream/handle/10016/34486/TFG_Ruben_Gonzalez_Lozano.pdf?sequence=1

Lucey, P. a. (2010). 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. En *The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression* (págs. 94-101). San Francisco. doi:10.1109/CVPRW.2010.5543262

Lucey, P. a. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. En *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops* (págs. 94-101). San Francisco, CA, EE. doi:10.1109/CVPRW.2010.5543262

Marrero, I. N. (2022). *Detección de anomalías cardíacas mediante algoritmos de inteligenci artificial.* Cartagena: Universidad Politécnica de Cartagena.

Martínez, J. (01 de 2020). *DataSmarts* . Obtenido de ¿Qué es un Optimizador y Para Qué Se Usa en Deep Learning?: https://datasmarts.net/es/que-es-un-optimizador-y-para-que-se-usa-en-deep-learning/

Matich, D. J. (03 de 2001). *"Redes Neuronales: Conceptos Básicos y.* Obtenido de Universidad Tecnológica Nacional de Rosario, Rosario,.

MATLAB. (s.f.). *Mathworks.com*. Obtenido de https://es.mathworks.com/products/matlab.html

MediaPipe. (s.f.). Obtenido de Google for Developers: https://developers.google.com/mediapipe

Michael J, L., Kamachi, M., & Gyoba, J. (s.f.). *The Japanese Female Facial Expression (JAFFE) Dataset*. doi:10.5281/zenodo.3451524

Michel F. Valstar, M. P. (s.f.). *Induced Disgust, Happiness and Surprise: an Addition to the MMI Facial*. Obtenido de https://mmifacedb.eu/

Morales, Y. A. (2014). *Clasificación de datos en el análisis discriminante*. Obtenido de Revista Vinculando.: https://vinculando.org/articulos/analisis-discriminante.html

Mukhiddinov, M., Djuraev, O., Akhmedov, F., Mukhamadiyev, A., & Cho, J. (2023). *Masked Face Emotion Recognition Based on Facial Landmarks and Deep Learning Approaches for Visually Impaired People.* doi:https://doi.org/10.3390/s23031080

Pareto Boada, J. (04 de 2022). *Robótica social asistencial. Implicaciones y desafíos éticos.* Obtenido de Upc.edu: https://www.iri.upc.edu/files/scidoc/2625-Rob%C3%B3tica-social-asistencial.-Implicaciones-y-desaf%C3%ADos-%C3%A9ticos.pdf

Parra Barrero, E. T. (2015). *idUS.* Obtenido de Universidad de Sevilla: https://idus.us.es/bitstream/handle/11441/30319/TrabajoFinGrado.pdf?sequence=1&isAllowed=y

Pérez, J. M., & P.S. Pérez Martin. (2023). *La curva ROC.* doi:https://doi.org/10.1016/j.semerg.2022.101821

Porta, O. (20 de 01 de 2020). *Historia de la visión artificial: así ha evolucionado esta tecnología*. Obtenido de INFAIMON: https://infaimon.com/blog/vision-2d-3d/historia-evolucion-vision-artificial/

Python. (s.f.). *What is python? Executive summary*. Obtenido de Python.org: https://www.python.org/doc/essays/blurb/

Quintal, S. (10 de 09 de 2023). *Redes neuronales: Descubriendo el potencial de la inteligencia artificial*. Obtenido de FUNIBER. Fundación Universitaria Iberoamericana: https://blogs.funiber.org/tecnologias-informacion/2023/10/09/redes-neuronales

Ríos, M. M. (04 de 2022). *ESIC*. Obtenido de ¿Qué es el deep learning y para qué sirve? : https://www.esic.edu/rethink/tecnologia/que-es-deep-learning-para-que-sirve

Rodriguez, A. H. (12 de 12 de 2021). *Mlearning Lab*. Obtenido de Vehículos autónomos, una carrera hacia la IA general: https://mlearninglab.com/2021/12/12/vehiculos-autonomos-una-carrera-hacia-la-ia-general/

Rodríguez, Y. (1 de 08 de 2023). *Guía sobre Datasets.* Obtenido de https://www.ironhack.com/es/blog/una-guia-sobre-datasets-que-son-como-se-utilizan-y-donde-encontrarlos

Roman, V. (2019). *Algoritmos Naive Bayes: Fundamentos e Implementación*. Obtenido de Ciencia y Datos: https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f

Rouhiainen, L. P. (2018). *Inteligencia Artificial 101 Cosas que debes saber hoy sobre nuestro futuro.* Barcelona: Editorial Planeta, S.A.

Roy, K. (2021). *Extended Cohn-Kanade Dataset (CK+)*. Obtenido de https://www.researchgate.net/profile/Koushik-Roy-6

Sarmiento-Ramos, J. L. (10 de 2020). *Revista UIS Ingenierías.* doi:https://doi.org/10.18273/revuin.v19n4-2020001

Thomaz, C. E. (s.f.). *FEI Face Database*. Obtenido de Centro Universitario da FEI, São Bernardo do Campo, São Paulo, Brazil: https://fei.edu.br/~cet/facedatabase.html

Zago Canal, F., Rossi Müller, T., Cristine Matias, J., Gino Scotton, G., Reis de Sa Junior, A., Pozzebon, E., & Sobieranski, A. (7 de 10 de 2021). *A survey on facial emotion recognition techniques: A state-of-the-art literature review.* Obtenido de Science Direct: https://doi.org/10.1016/j.ins.2021.10.005