



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

GRADO EN INGENIERÍA INFORMÁTICA DE GESTIÓN Y SISTEMAS DE INFORMACIÓN

TRABAJO FIN DE GRADO

MODELO PREDICTIVO DE RECURRENCIA DE FIBRILACIÓN AURICULAR: PRAFAI

Alumno/Alumna: Valdelvira, Vázquez, Iker

Director/Directora (1): Atutxa, Salazar, Aitziber

Director/Directora (2): Gojenola, Galletebeitia, Koldobika

Curso: 2021-2022

Fecha: Bilbao, 03, febrero, 2022

Resumen

La fibrilación auricular (FA) es una de las arritmias más comunes en el mundo hoy en día. Aparte de la anticoagulación, no se ha definido hasta el momento un tratamiento óptimo de control de frecuencia o ritmo. Una herramienta con un algoritmo predictivo capaz de predecir la recurrencia de fibrilación auricular basándose en las características del paciente sería de gran ayuda para tomar las decisiones terapéuticas. Por este motivo ha surgido el Proyecto de Investigación PRAFAI (Prediction of Recurrence of Atrial Fibrillation using Artificial Intelligence), y este documento tiene como objetivo explicar el planteamiento, ejecución y resultados del mismo.

El tema de investigación es la predicción de recurrencia de fibrilación auricular tras un primer episodio de la arritmia haciendo uso de herramientas de Inteligencia Artificial. Por lo tanto, el objetivo principal del estudio es diseñar una herramienta tecnológica con técnicas de aprendizaje automático para el cálculo de riesgo de recurrencia o permanencia en fibrilación auricular en pacientes de la OSI (Organización Sanitaria Integrada) de un hospital de la red de salud vasca que presentan un primer evento documentado de FA.

En primer lugar, se ha recibido una base de datos con pacientes diagnosticados con fibrilación auricular y un conjunto de datos relacionados con esos pacientes que contiene variables demográficas, antecedentes personales generales, antecedentes cardiovasculares específicos, aspectos hospitalarios, pruebas de laboratorio, procedimientos realizados e intervenciones realizadas. La primera fase ha consistido en la unión de los datos necesarios de cada paciente para la creación del conjunto de datos con el que se entrenará y evaluará la herramienta predictiva. Las siguientes fases del proyecto han sido las relacionadas con tareas de Inteligencia Artificial y Machine Learning: preprocessado de datos, elección de los algoritmos de aprendizaje supervisado, implementación y generación de los modelos predictivos, evaluación de los resultados y análisis de las variables predictoras más importantes.

El resultado del proyecto es la obtención del modelo predictivo óptimo para el cálculo automático del riesgo de recurrencia de fibrilación auricular. Es un modelo de clasificación binaria que diferencia entre pacientes que presentan recurrencia en la arritmia y pacientes sin recurrencia. Su finalidad, en primer lugar, es clasificar correctamente los pacientes y obtener la probabilidad de la predicción y, en segundo lugar, identificar las características más relevantes para el éxito de la predicción.

Palabras clave: fibrilación auricular, riesgo de recurrencia, Inteligencia Artificial, Machine Learning, aprendizaje supervisado, clasificación binaria.

Laburpena

Fibrilazio aurikularra (FA) gaur egungo arritmia arruntenetako bat da munduan. Antikoagulazioaz gain, orain arte ez da definitu maiztasuna edo erritmoa kontrolatzeko tratamendu optimorik. Pazientearen ezaugarrien arabera fibrilazio aurikularren errepikapena iragartzeko gai den algoritmo prediktiboa duen tresna oso lagungarria litzateke erabaki terapeutikoak hartzeko. Arrazoi horregatik sortu da PRA-FAI (Prediction of Recurrence of Atrial Fibrillation using Artificial Intelligence) ikerketa-proiektua, eta dokumentu honen helburua bere planteamendua, gauzatzea eta emaitzak azaltzea da.

Ikerketaren gaia, arritmien lehen gertaera baten ondoren, fibrilazio aurikularra berriz agertuko den iragartzea da, Adimen Artifizialeko tresnak erabiliz. Beraz, ikerlanaren helburu nagusia ikasketa automatikoko teknikekin tresna teknologiko bat diseinatzea da, euskal osasun-sareko ospitale bateko ESI-ko (Erakunde Sanitario Integratua) pazientei fibrilazio aurikularra berriro agertzeko edo irauteko arriskua kalkulatzeko, FAren lehen gertaera dokumentatua aurkezten badute.

Hasteko, fibrilazio aurikularra diagnostikatu zaien pazienteei buruzko datu-basea jaso da, baita gaixo horiei buruzko datuak ere: aldagai demografikoak, aurrekari pertsonal orokorrak, aurrekari kardiobaskular espezifikoak, ospitale-alderdiak, laborategiko probak, egindako prozedurak eta ebakuntzak. Lehenengo fasean, paziente bakoitzaren datuak batu dira, erreminta prediktiboa entrenatu eta ebaluatzeko datu-multzoa sortzeko. Proiektuaren hurrengo faseak Adimen Artifizial eta Machine Learning atazen ingurukoak izan dira: datuen aurreprozesaketa, gainbegiratutako ikasketa-algoritmoen hautaketa, eredu iragarleen implementazioa, emaitzen ebaluazioa eta aldagai iragarle garrantzitsuenen azterketa.

Proiektuaren emaitza fibrilazio aurikularra berragertzeko arriskua automatikoki kalkulatzen duen eredu iragarle optimoa da. Arritmian errekurrentzia duten pazienteak eta errekurrentziarik ez duten pazienteak bereizten dituen sailkapen bitarreko eredua da. Horren helburua, lehenik eta behin, pazienteak behar bezala sailkatzea eta iragarpenaren probabilitatea lortzea da, eta, bigarrenik, iragarpenak arrakasta izateko aldagai garrantzitsuenak identifikatzea.

Hitz gakoak: fibrilazio aurikularra, errekurrentzia arriskua, Adimen Artifiziala, Machine Learning, ikasketa gainbegiratua, sailkapen bitarra.

Abstract

Atrial fibrillation (AF) is one of the most common arrhythmias in the world nowadays. Apart from anticoagulation, an optimal rate or rhythm control therapy has not been defined so far. A tool with a predictive algorithm able to predict the recurrence of atrial fibrillation based on the characteristics of the patient would be helpful in making therapeutic decisions. For this reason PRAFAI (Prediction of Recurrence of Atrial Fibrillation using Artificial Intelligence) Research Project was created, and this document aims to explain its approach, execution and results.

The research topic is the prediction of atrial fibrillation recurrence after a first episode of the arrhythmia using Artificial Intelligence tools. Therefore, the main objective of the study is to design a technological tool with Machine Learning techniques to calculate the risk of recurrence or permanence in atrial fibrillation in patients from a HMO (Health Maintenance Organization) hospital of the Basque health network who present a first documented event of AF.

First of all, a database with patients diagnosed with atrial fibrillation and a dataset related to these patients were received. Related data contains demographic variables, general personal history, specific cardiovascular history, hospital aspects, laboratory tests, procedures performed and interventions. The first phase consisted in the union of the necessary data of each patient for the dataset creation with which the predictive tool will be trained and evaluated. The next phases of the project were those related to Artificial Intelligence and Machine Learning tasks: data pre-processing, choice of supervised learning algorithms, implementation and generation of predictive models, results evaluation and most important predictive variables analysis.

The result of the project is the optimal predictive model to calculate automatically the risk of recurrence of atrial fibrillation. It is a binary classification model that differentiates between patients with arrhythmia recurrence and patients without recurrence. Its purpose, firstly, is to correctly classify the patients and obtain a probability for the prediction and, secondly, to identify the most relevant features for succeeding the prediction.

Keywords: atrial fibrillation, risk of recurrence, Artificial Intelligence, Machine Learning, supervised learning, binary classification.

Índice general

1. Introducción	14
1.1. Contexto	14
1.2. Origen del proyecto	15
1.3. Objetivos	15
1.3.1. Hipótesis del estudio	15
1.3.2. Motivación del proyecto	15
1.3.3. Objetivo principal	16
1.3.4. Objetivos secundarios	16
1.4. Beneficios que aporta el trabajo	16
1.4.1. Impacto esperado	16
1.4.1.1. Valor añadido	16
1.4.2. Desarrollo de nuevos productos y servicios	16
1.4.2.1. Mejora de la salud	16
1.4.3. Posibilidad de generalización de los resultados	17
2. Marco teórico y Estado del arte	18
2.1. Terminología básica	18
2.1.1. IA en el presente proyecto	19
2.2. Desafíos en ML tradicional y DL	19
2.3. Antecedentes en el área médica	20
2.4. Antecedentes de IA en fibrilación auricular	23
3. Planificación y análisis del proyecto	26
3.1. Arquitectura	26
3.2. Herramientas utilizadas	28
3.3. Alcance del proyecto	29
3.3.1. Planificación y gestión	31
3.3.2. Análisis	32
3.3.3. Diseño	36
3.3.4. Implementación y desarrollo	41
3.3.5. Documentación	44
3.3.6. Alcance de la tarea secundaria: ECG-PRAFAI	45
3.3.6.1. Diseño	47
3.3.6.2. Implementación y desarrollo	50
3.4. Cronograma / Diagrama de Gantt	52
3.5. Evaluación económica	56
3.5.1. Costes por mano de obra	56
3.5.2. Costes en hardware	56
3.5.3. Costes en software	56
3.5.4. Costes indirectos	57
3.5.5. Costes totales	57
3.6. Análisis de riesgos	57

4. Descripción, diseño y desarrollo de la solución	60
4.1. Descripción del corpus	60
4.1.1. Criterio de inclusión	60
4.1.2. Criterio de exclusión	60
4.1.3. Tamaño de la muestra	60
4.2. Estudio de variables/rasgos	61
4.2.1. Demográficas	61
4.2.2. Historia médica personal	62
4.2.3. Historia cardiovascular específica	63
4.2.4. Hospitalizaciones	65
4.2.5. Pruebas de laboratorio	65
4.2.6. Tratamientos (códigos ATC)	65
4.2.7. Procedimientos	66
4.2.8. Intervenciones	66
4.3. Diseño y desarrollo de la solución	66
4.3.1. Extracción de datos	70
4.3.2. Software diseñado	71
4.3.2.1. Creación del <i>dataset</i>	71
4.3.2.2. División del conjunto de datos y preprocesado	74
4.3.2.2.1. División del <i>dataset</i>	75
4.3.2.2.2. Gestión de <i>missing values</i>	75
4.3.2.2.3. Discretización de alguna variable numérica	80
4.3.2.2.4. Estandarización / Normalización de variables numéricas	80
4.3.2.3. Selección de variables / Reducción de dimensiones	81
4.3.2.3.1. <i>Feature selection</i> (selección de variables)	82
4.3.2.3.2. <i>Dimension reduction</i> (reducción de dimensiones)	89
4.3.2.4. Importancia/Correlación de las variables	89
4.3.2.4.1. <i>Feature importance</i> (importancia de las variables)	90
4.3.2.4.2. Correlación de las variables (<i>heatmaps</i>)	92
4.3.2.5. Entrenamiento de modelos	93
4.3.2.6. Evaluación de modelos	103
4.3.2.6.1. Esquema de evaluación: <i>k-fold Cross Validation</i>	103
4.3.2.6.2. Métricas de evaluación	104
4.3.2.7. Selección del modelo óptimo	106
5. Resultados	108
5.1. Resultados de las evaluaciones de los modelos	108
5.2. Resultados del mejor modelo predictivo	113
5.2.1. Modelo óptimo con mejoras	115
5.2.1.1. Resultados en el conjunto Test	117
5.2.1.2. Variables predictoras seleccionadas	119
6. Tarea secundaria: PRAFAI en base a ECGs	126
6.1. Introducción y objetivos	126
6.2. Estado del arte	127
6.2.1. Bases de datos: AF Termination Challenge Database y Long Term AF Database	128
6.2.1.1. AF Termination Challenge Database [45]	128
6.2.1.2. Long Term AF Database Petruti <i>et al.</i> [54]	130
6.3. Tareas previas al desarrollo de la solución	130
6.3.1. Herramienta de conversión de ECGs en formato MUSE-XML	131
6.3.2. Elección de un conjunto de datos compatible con los ECGs de la OSI del hospital	131
6.3.2.1. Base de datos: <i>ECG recording from Chapman University and Shaoxing People's Hospital</i> [79]	132
6.3.2.2. Inconveniente	132
6.3.2.3. Solución propuesta	133
6.4. Diseño y desarrollo de la solución	133
6.4.1. Baseline: <i>XGBoost</i>	134
6.4.2. Red neuronal	135

6.4.2.1. Red neuronal recurrente (prueba)	139
6.4.3. Experimento: reducción de ítems	141
6.5. Resultados	142
7. Conclusiones	146
7.1. Análisis de los resultados y contratiempos	146
7.2. Análisis de la planificación estimada y real	148
7.3. Trabajo futuro	150
7.4. Reflexión personal	150
Anexo I: Documento de requisitos	151
Anexo II: Últimos estudios relacionados con modelos predictivos de FA	154
- Scores clínicos para los resultados del control del ritmo o la progresión de la arritmia en pacientes con fibrilación auricular: una revisión sistemática	154
- Evaluación de un modelo predictivo para el desarrollo de fibrilación auricular en un repositorio de historias médicas electrónicas (EMRs)	158
- Modelo predictivo de riesgo de fibrilación auricular incidente en la población japonesa general (The Hisayama Study)	158
- Modelo predictivo para predecir la mortalidad en pacientes con fibrilación auricular (The BASIC-AF risk score)	159
- CHARGE-AF en una base de datos de EMRs de atención primaria de rutina en los Países Bajos: validación del riesgo de fibrilación auricular a 5 años e implicaciones para la selección de pacientes en el cribado de fibrilación auricular	159
- Peso, altura, cambio de peso y riesgo de fibrilación auricular incidente en hombres y mujeres de mediana edad	160
- Rendimiento de modelos predictores de riesgo de fibrilación auricular sobre 4 millones de personas	160
- Desarrollo de un score de riesgo de fibrilación auricular en adultos con diabetes mellitus (ACCORD Study)	161
- Modelos de predicción de fibrilación auricular aplicables en la comunidad: revisión sistemática y metaanálisis	161
- Evaluación de un modelo de aprendizaje automático aplicado a datos de registros médicos electrónicos (EMRs) armonizados para la predicción de fibrilación auricular incidente	162
Anexo III: Últimos estudios relacionados con modelos predictivos de FA utilizando ECGs	163
- AFibNet: An Implementation of Atrial Fibrillation Detection with Convolutional Neural Network	163
- Classification of 12-lead ECGs: the PhysioNet / Computing in Cardiology Challenge 2020 . .	166
- <i>prna</i> team	167
- <i>Between_a_ROC_and_a_heart_place</i> team	168
- JAMA Cardiology — Performance of a Convolutional Neural Network and Explainability Technique for 12-Lead Electrocardiogram Interpretation	169
- Circulation: Deep Neural Networks Can Predict NewOnset Atrial Fibrillation From the 12-Lead ECG and Help Identify Those at Risk of Atrial Fibrillation–Related Stroke	169
- Convolutional Neural Network and Rule-Based Algorithms for Classifying 12-lead ECGs . . .	171
Bibliografía	171

Índice de figuras

2.1. Resultados del estudio <i>Chen et al.</i> [12]	21
2.2. Resultados cambiando el tamaño de la muestra de entrenamiento del estudio CRS-PSC [10]	21
3.1. Arquitectura del software	27
3.2. Bloques principales del diagrama EDT	29
3.3. Diagrama EDT	30
3.4. Diagrama EDT del bloque de planificación y gestión	31
3.5. Diagrama EDT del bloque de análisis	33
3.6. Diagrama EDT del bloque de diseño	36
3.7. Diagrama EDT del bloque de implementación y desarrollo	41
3.8. Diagrama EDT del bloque de documentación	44
3.9. Diagrama EDT de la tarea secundaria: ECG-PRAFAI	46
3.10. Diagrama EDT del bloque de diseño (ECG-PRAFAI)	47
3.11. Diagrama EDT del bloque de implementación y desarrollo (ECG-PRAFAI)	50
3.12. Diagrama de Gantt	55
4.1. Diseño de la solución (resumido en bloques de tareas)	68
4.2. Diseño de la solución (detallado)	69
4.3. Extracción de datos	70
4.4. Software diseñado	71
4.5. Diseño de la creación del <i>dataset</i>	71
4.6. Distribución de la clase	73
4.7. Distribución de la variable <i>edad</i>	74
4.8. Diseño del preprocesado y división Train/Dev/Test	75
4.9. Distribución de <i>missing values</i> por variables	78
4.10. Distribución de <i>missing values</i> por variables (porcentajes)	79
4.11. Fórmulas para la estandarización mediante <i>StandardScaler</i>	81
4.12. Fórmula para la estandarización mediante <i>MinMaxScaler</i>	81
4.13. Fórmula para la estandarización mediante <i>MaxAbsScaler</i>	81
4.14. Fórmula para la estandarización mediante <i>Normalizer</i>	81
4.15. Diseño de la selección de variables / reducción de dimensiones e importancia/correlación de las variables	82
4.16. <i>Filter methods</i> para selección de variables	82
4.17. <i>Wrapper methods</i> para selección de variables	82
4.18. <i>Embedded methods</i> para selección de variables	83
4.19. Métrica AUC respecto al número de variables seleccionadas en el <i>dataset</i> Train/Dev_Mean_StandardScaler	84
4.20. Coeficientes de regularización L1 (Lasso) en las variables del <i>dataset</i> Train/Dev_Mean_StandardScaler	86
4.21. Coeficientes de impureza de Gini en las variables del <i>dataset</i> Train/Dev_Mean_StandardScaler	87
4.22. Coeficientes de la prueba χ^2 en las variables del <i>dataset</i> Train/Dev_Mean_MinMaxScaler .	88
4.23. Porcentaje de varianza explicada dependiendo del número de componentes principales seleccionados	89
4.24. Importancia de las variables en un modelo de <i>LogisticRegression</i> sobre el <i>dataset</i> Train/- Dev_Mean_StandardScaler	91
4.25. <i>Heatmap</i> de correlación del <i>dataset</i> Train/Dev_MissingValues_StandardScaler	92
4.26. Correlación de algunas variables con la clase <i>sigue_fa</i> en el <i>dataset</i> Train/Dev_MissingValues_StandardScaler	93
4.27. Diseño del entrenamiento de modelos	94
4.28. Ejemplo del algoritmo <i>DecisionTree</i>	95

4.29. Ejemplo del algoritmo <i>KNN</i>	95
4.30. Ejemplo del algoritmo <i>LogisticRegression</i>	96
4.31. Ejemplo del algoritmo <i>Perceptron</i>	97
4.32. Ejemplo del algoritmo <i>SVM</i>	98
4.33. Ejemplo del algoritmo <i>RandomForest</i>	99
4.34. Ejemplo del algoritmo <i>MLP</i>	100
4.35. Ejemplo del algoritmo <i>Bagging</i>	101
4.36. Ejemplo del algoritmo <i>AdaBoost</i>	102
4.37. Diseño de la evaluación de los modelos	103
4.38. Ejemplo <i>4-fold Cross Validation</i>	104
4.39. Matriz de confusión con 2 etiquetas de clase	105
4.40. Ejemplo de curva ROC y AUC	106
4.41. Diseño de la selección del modelo óptimo	107
4.42. Ejemplo de predicciones en 6 instancias del conjunto Test	107
5.1. Mejores AUC por algoritmo tras <i>feature selection</i> con <i>RFE</i>	111
5.2. Mejores AUC por algoritmo sobre los conjuntos de datos <i>Raw</i>	112
5.3. Mejores AUC por algoritmo tras reducción de dimensionalidad con <i>PCA</i>	113
5.4. Ejemplo de hiperplano de separación creado con <i>SVM</i>	114
5.5. Curva ROC y AUC tras evaluación con <i>5-fold Cross Validation</i> en Train/Dev	115
5.6. Curva ROC y AUC tras evaluación con <i>10-fold Cross Validation</i> en Train/Dev	117
5.7. Curva ROC y AUC tras evaluación en Test	119
5.8. Predicción sobre 10 instancias de Test	119
5.9. Métrica AUC respecto al número de variables seleccionadas	120
5.10. Coeficientes (absolutos) de importancia de las variables en un modelo <i>SVM</i> lineal	122
5.11. Coeficientes de importancia de las variables en un modelo <i>SVM</i> lineal	123
5.12. Distribución de la variable <i>hipertiroidismo</i>	124
5.13. Distribución de la variable <i>c04</i>	124
5.14. Métrica AUC respecto al número de variables seleccionadas (I)	125
5.15. Métrica AUC respecto al número de variables seleccionadas (II)	125
6.1. Ejemplo de onda de un ECG y los segmentos de la derivación II que presentan un ciclo cardíaco normal.	126
6.2. ECG de dos canales del conjunto N (persistencia/recurrencia de FA)	129
6.3. ECG de dos canales del conjunto S (terminación de FA un minuto después del final del registro)	129
6.4. ECG de dos canales del conjunto T (terminación inmediata de FA después del final del registro)	130
6.5. 12-lead ECG convertido a formato CSV	131
6.6. Ejemplo de registro ECG de un paciente en FA, en el que desaparece la onda P y la distancia entre los QRS varía.	133
6.7. Curva ROC y AUC del modelo <i>aVR</i> sobre el conjunto Dev	135
6.8. Arquitectura de la CNN FCN + MLP (I)	138
6.9. Arquitectura de la CNN FCN + MLP (II)	138
6.10. Ejemplo neurona recurrente	139
6.11. Arquitectura de la RNN con dos capas LSTM bidireccionales	140
6.12. Curva ROC y AUC del modelo RNN (LSTM) sobre el conjunto Dev	141
6.13. AUC respecto al número de ítems de entrenamiento	143
6.14. <i>Accuracy</i> respecto al número de ítems de entrenamiento	143
6.15. <i>Loss</i> respecto al número de ítems de entrenamiento	144
6.16. <i>Accuracy</i> y AUC respecto al número de ítems de entrenamiento sobre el conjunto Test	144
7.1. Comparativa de planificaciones en el diagrama de Gantt	149
7.2. Diseño de la CNN del estudio <i>AFibNet</i>	165
7.3. Arquitectura de la CNN del estudio <i>AFibNet</i>	166
7.4. Arquitectura de la red neuronal del estudio del equipo <i>prna</i>	167
7.5. Arquitectura de la red neuronal del estudio del equipo <i>Between_a_ROC_and_a_heart_place</i>	168
7.6. Bloque residual (<i>ResB</i>) y bloque SE de la red neuronal del equipo <i>Between_a_ROC_and_a_heart_place</i>	168

7.7. Segmentos de la derivación V2 de un ECG resaltados por LIME como más importantes para la clasificación de fibrilación auricular	169
7.8. Arquitectura de la DNN del estudio de <i>Circulation</i>	170
7.9. Rendimiento de los modelos del estudio de <i>Circulation</i> en cuanto a las métricas AUROC y AUPRC	170

Índice de cuadros

2.1. Conjuntos de datos del estudio [2]	20
3.1. Reuniones con el cliente/data-manager	31
3.2. Reuniones con el grupo de trabajo	32
3.3. Objetivos y alcance	32
3.4. Beneficios que aporta	32
3.5. Cronograma / Diagrama de Gantt	32
3.6. Captura de requisitos	33
3.7. Marco teórico	34
3.8. Estado del arte	34
3.9. Análisis de los datos de entrada	34
3.10. Selección del <i>framework</i> para el desarrollo del software	35
3.11. Selección de recursos	35
3.12. Análisis de los cambios y mejoras propuestas	35
3.13. Diseño del <i>dataset</i>	37
3.14. División del conjunto de datos en Train/Dev/Test	37
3.15. Métodos de imputación de <i>missing values</i>	37
3.16. Métodos de discretización de algunas variables numéricas	38
3.17. Estandarización/Normalización de variables numéricas	38
3.18. Métodos de selección de variables y reducción de dimensiones	38
3.19. Métodos de importancia de variables y <i>heatmaps</i> de correlación	39
3.20. Algoritmos	39
3.21. Hiperparámetros a optimizar	39
3.22. Esquema de evaluación	40
3.23. Métricas de evaluación	40
3.24. Creación del <i>dataset</i>	41
3.25. Preprocesado de los datos	42
3.26. Selección de variables y reducción de dimensiones	42
3.27. Importancia de variables y <i>heatmaps</i> de correlación	42
3.28. Entrenamiento de modelos predictivos con algoritmos de ML	43
3.29. Evaluación de los modelos	43
3.30. Selección del mejor modelo	43
3.31. Implementación de los cambios y mejoras propuestas	43
3.32. Búsqueda de material de apoyo	44
3.33. Creación de diagramas y tablas	44
3.34. Redacción de la memoria	44
3.35. Preparación de la defensa	45
3.36. Herramienta de conversión de ECGs en formato MUSE-XML	47
3.37. Elección del <i>dataset</i> compatible con los ECGs de la OSI	48
3.38. División del conjunto de datos en Train/Dev/Test	48
3.39. Estandarización de los datos	48
3.40. Baseline	48
3.41. Modelo de <i>deep learning</i>	49
3.42. Esquema de evaluación	49
3.43. Métricas de evaluación	49
3.44. Diseño del experimento: reducción de ítems	49

3.45. Implementación del conversor con ECGs en formato MUSE-XML	50
3.46. Obtención <i>dataset</i> compatible con los ECGs de la OSI	50
3.47. Preprocesado de los datos	51
3.48. Entrenamiento de los clasificadores	51
3.49. Evaluación de los modelos	51
3.50. Implementación del experimento: reducción de ítems	51
3.51. Planificación del tiempo (I)	52
3.52. Planificación del tiempo (II)	53
3.53. Costes totales del proyecto	57
3.54. Riesgo 1	57
3.55. Riesgo 2	58
3.56. Riesgo 3	58
3.57. Riesgo 4	58
3.58. Riesgo 5	59
3.59. Riesgo 6	59
3.60. Riesgo 7	59
3.61. Riesgo 8	59
4.1. Categorización del IMC	62
5.1. Mejores resultados en AUC tras <i>feature selection</i> con RFE (I)	109
5.2. Mejores resultados en AUC tras <i>feature selection</i> con RFE (II)	110
5.3. Resultados tras evaluación con <i>5-fold Cross Validation</i> en Train/Dev (I)	115
5.4. Resultados tras evaluación con <i>10-fold Cross Validation</i> en Train/Dev (II)	117
5.5. Resultados tras evaluación en Test	118
6.1. Evaluación del modelo <i>aVR</i> sobre los conjuntos Dev y Test	135
6.2. Evaluación de las redes neuronales sobre los conjuntos Dev y Test mediante <i>Hold-Out</i>	137
6.3. Evaluación del modelo FCN + MLP sobre los conjuntos Dev mediante <i>10-fold Cross Validation</i>	137
6.4. Evaluación del modelo RNN (LSTM) sobre el conjunto Dev	141
6.5. Evaluaciones de los modelos FCN + MLP al reducir el tamaño del conjunto de entrenamiento	142
7.1. Requisito 1	151
7.2. Requisito 2	151
7.3. Requisito 3	151
7.4. Requisito 4	151
7.5. Requisito 5	152
7.6. Requisito 6	152
7.7. Requisito 7	152
7.8. Requisito 8	152
7.9. Requisito 9	152
7.10. Requisito 10	153
7.11. Requisito 11	153

Acrónimos

- **FA:** Fibrilación auricular
- **PRAFAI:** Prediction of Recurrence of Atrial Fibrillation using Artificial Intelligence
- **OSI:** Organización Sanitaria Integrada
- **IA:** Inteligencia Artificial
- **ML:** Machine Learning (aprendizaje automático)
- **TIC:** Tecnologías de la Información y la Comunicación
- **ECG:** Electrocardiograma
- **EMR:** Electronic Medical Record (expediente médico electrónico)
- **ROC:** Receiver Operating Characteristic (característica operativa del receptor)
- **AUC:** Area Under Curve (área bajo curva)
- **IMC:** Índice de Masa Corporal
- **FEVI:** Fracción de Eyección del Ventrículo Izquierdo
- **PCA:** Principal Component Analysis (análisis de componentes principales)
- **SVM:** Support Vector Machines (maquinas de vectores de soporte)
- **MLP:** Multilayer Perceptron (perceptrón multicapa)
- **KNN:** k-Nearest Neighbors (k vecinos más cercanos)
- **EDT:** Estructura de Descomposición del Trabajo
- **RFE:** Recursive Feature Elimination (eliminación de características recursivas)
- **SFS:** Sequential Feature Selection (selección secuencial de características)
- **CNN:** Convolutional Neural Network (red neuronal convolucional)
- **FCN:** Fully Convolutional Network (red totalmente convolucional)
- **RNN:** Recurrent Neural Network (red neuronal recurrente)
- **LSTM:** Long Short-Term Memory (red de memoria a largo y corto plazo)

1. Introducción

1.1. Contexto

La fibrilación auricular (FA) es una de las arritmias más comunes en el mundo y se relaciona con una alta tasa de morbilidad cardiovascular. Su tratamiento depende de la anticoagulación y de la elección entre revertir el ritmo de FA o ajustar la frecuencia cardiaca, ya que esta arritmia puede provocar taquicardia. La gestión óptima del tratamiento no se ha definido todavía y un algoritmo capaz de predecir la recurrencia de fibrilación auricular teniendo en cuenta las características del paciente será una herramienta de gran ayuda para guiar las decisiones terapéuticas. Hay que tener en cuenta que a día de hoy, en la era del Big Data, se comienza a tener datos suficientes como para poder hacer real la medicina personalizada.

Aunque no existe una definición universalmente aceptada, el Grupo Asesor de Horizonte 2020 define la medicina personalizada como un modelo médico que utiliza la caracterización de los fenotipos y genotipos de las personas (por ejemplo, perfiles moleculares, imágenes médicas, datos de estilo de vida) para adaptar la estrategia terapéutica adecuada para la persona adecuada, en el momento adecuado, y/o para determinar la predisposición a la enfermedad y/o para brindar una prevención oportuna y específica. Esta definición también fue utilizada por los ministros de Sanidad de la UE en sus conclusiones del Consejo sobre medicina personalizada para pacientes, publicadas en diciembre de 2015 [1]. Asimismo, la Comisión Europea presenta las herramientas tecnológicas y soluciones digitales para la salud y la asistencia, en particular la medicina personalizada, como uno de los ámbitos de intervención para mejorar la competitividad de la industria europea.

El presente proyecto supone un paso en dirección hacia la medicina personalizada, ya que su finalidad es diseñar una herramienta basada en Inteligencia Artificial para el cálculo automático del riesgo de recurrencia de FA en determinado paciente con un primer evento registrado.

La fibrilación auricular es la arritmia más prevalente en el mundo y afecta a más del 4 % de la población española por encima de los 40 años, una prevalencia que se prevé que aumente [22]. Para el tratamiento de esta arritmia se utilizan dos estrategias. La primera, consiste en la anticoagulación. El tratamiento anticoagulante disminuye la incidencia de eventos cardioembólicos. La segunda, es la elección entre control de frecuencia o ritmo (*rate or rhythm control*) de la arritmia, la cual no está tan clara.

Las Guías Americanas de 2014 dicen que hay una mayor frecuencia de complicaciones cuando se opta por la estrategia de control de ritmo (*rhythm control*), y en la actualización de 2019, no hay cambios acerca de las recomendaciones en estas estrategias [30]. En las Guías Europeas recientemente publicadas se dice que las estrategias de control de frecuencia o ritmo (*rate or rhythm control*) no se diferencian en cuanto a la mortalidad [26].

Aunque algunos meta-análisis y revisiones sistemáticas se posicionaron a favor del control de frecuencia (*rate control*), ya que daba menor frecuencia de complicaciones [61], estudios más recientes muestran que la ablación por catéter para el control de ritmo (*rhythm control*) podría tener ventajas comparado con los tratamientos médicos en pacientes con insuficiencia cardiaca y FA. Además, la comparación entre la ablación por catéter y la terapia médica para el control del ritmo (*rhythm control*) en pacientes con insuficiencia cardiaca no demuestra una mejora significativa en el grupo de ablación [51].

Por ello, teniendo en cuenta los discretos resultados favorables de la estrategia de control de ritmo (*rhythm control*) en el análisis por protocolo del ensayo CABANA y en la muestra seleccionada EAST-AFNET 4, las dos estrategias serían válidas, lo que causa grandes dudas en la práctica clínica a la hora de elegir el tratamiento óptimo para el paciente con un diagnóstico de FA.

Como la recurrencia de FA a pesar de la estrategia de control de ritmo (*rhythm control*) es muy frecuente (30-50 % tras la ablación), la decisión terapéutica es aún más compleja. Por lo tanto, la decisión sobre la gestión terapéutica depende de los síntomas del paciente y la ‘suposición’ sobre la probabilidad de éxito del control de ritmo (*rhythm control*).

La predicción de la probabilidad de recurrencia de la arritmia ya ha sido probada en otros *scores* clínicos. La mayoría han sido estudiados en poblaciones o intervenciones muy específicas, por lo que la aplicación en la práctica clínica es limitada [17]. Las herramientas basadas en la Inteligencia Artificial pueden mejorar la selección de variables con mayor valor predictivo, generando así la posibilidad de plantear soluciones terapéuticas personalizadas.

Ninguno de los proyectos revisados ha investigado el riesgo de permanencia o recurrencia de FA a partir de un primer episodio basados en la historia clínica electrónica, por lo que se considera de gran interés en la práctica clínica. Como conclusión, se espera que el uso del aprendizaje automático supervisado ayude a la identificación de patrones ocultos en los datos, para obtener una óptima solución al identificar el riesgo de recurrencia de FA y, como consecuencia, tomar mejores decisiones terapéuticas.

1.2. Origen del proyecto

El proyecto ha sido propuesto por Aitziber Atutxa y Koldobika Gojenola, docentes del departamento de Lenguajes y Sistemas Infomáticos y miembros del grupo de investigación IXA en la Escuela de Ingeniería de Bilbao de la Universidad del País Vasco (UPV/EHU). Es un proyecto coordinado con investigadores del grupo de cardiología clínica de Biocruces - Hospital Bilbao-Basurto, de donde parte la idea y la necesidad de desarrollar la herramienta. Ellos son los responsables del Proyecto de Investigación PRAFAI (Prediction of Recurrence of Atrial Fibrillation using Artificial Intelligence) y también son los encargados de proporcionar los datos y características de los pacientes con un primer evento de FA. El hecho de colaborar con investigadores de un grupo de cardiología lo convierte en un proyecto con aplicación ‘real’, que pueda ser funcional fuera del marco universitario y que combina los ámbitos de la medicina y la informática con el objetivo de mejorar y facilitar las decisiones clínicas.

1.3. Objetivos

A continuación se presentan las razones y motivaciones que han impulsado la realización de este proyecto, así como los objetivos a alcanzar esperados tras la finalización del mismo.

1.3.1. Hipótesis del estudio

La Inteligencia Artificial hace posible ofrecer un modelo predictivo capaz de reducir el tiempo y facilitar la toma de decisiones en la gestión terapéutica de los pacientes con un primer diagnóstico de fibrilación auricular. Esta tecnología se puede traducir en una herramienta fácil de usar para los sanitarios a la hora de predecir una recurrencia en fibrilación auricular basándose en las características del paciente.

1.3.2. Motivación del proyecto

La fibrilación auricular es la arritmia más prevalente en el mundo y se relaciona con una alta tasa de morbilidad cardiovascular. No se conocen modelos predictivos de recurrencia de FA a partir de un primer episodio basados en la historia clínica electrónica que hayan utilizado aprendizaje automático, técnica mediante la cual se ha demostrado poder predecir el desarrollo de FA a partir de historias electrónicas o de electrocardiogramas en pacientes sin arritmia.

1.3.3. Objetivo principal

Diseñar una herramienta tecnológica basada en Machine Learning (Aprendizaje Automático) para la predicción automática de riesgo de recurrencia en pacientes de la OSI (Organización Sanitaria Integrada) de un hospital de la red de salud vasca que han presentado un primer episodio de FA, con el objetivo de facilitar la gestión terapéutica en esos pacientes.

1.3.4. Objetivos secundarios

- Generar una base de datos depurada y estructurada de pacientes con un primer evento de FA de la OSI del hospital a partir de su historia clínica electrónica para el diseño y validación de modelos predictivos, tanto para la hipótesis de este estudio como para hipótesis futuras. No es solo generar una base de datos, es generar el primer corpus que existe de estas características.
- Realizar un análisis de las variables predictoras más influyentes en las predicciones de recurrencia de FA realizadas por el modelo predictivo diseñado.
- Tarea secundaria (ECG-PRAFAI) con el objetivo de realizar un mismo modelo predictivo de recurrencia o persistencia de FA añadiendo las características de los propios electrocardiogramas de los pacientes con un primer evento de FA.

1.4. Beneficios que aporta el trabajo

Se documentará una estimación precisa de incidencia y recurrencia de FA en un periodo de dos años en la OSI del hospital con el que se trabaja desde enero de 2015 hasta septiembre de 2018. Además, se espera obtener una relación con las variables predictoras más influyentes para la predicción de recurrencia de FA. Esto nos llevará a crear una herramienta informática con la capacidad de guiar un tratamiento terapéutico adecuado prediciendo el riesgo de recurrencia. A continuación se pasa a presentar el impacto esperado de este proyecto, junto con la posibilidad de generalización de los resultados obtenidos partiendo de los datos de la OSI del hospital a otros pacientes.

1.4.1. Impacto esperado

1.4.1.1. Valor añadido

El uso de la Inteligencia Artificial en la cardiología es innovador. El uso de las Tecnologías de la Información y la Comunicación (TIC) en conjunto con las condiciones externas de los entornos en los que se trabaja son capaces de formular una estrategia efectiva, robusta y sostenible basada en la innovación y tecnología.

La principal utilidad práctica de la herramienta a desarrollar se basa en los siguientes puntos:

- Apoyar la toma de decisiones basada en evidencias como las características básicas de los pacientes y su relación con la recurrencia de fibrilación auricular.
- Permitir que estos procedimientos se incorporen a la formación de los profesionales en cardiología y puedan evaluar virtualmente su aprendizaje y mejorarlo.

1.4.2. Desarrollo de nuevos productos y servicios

El desarrollo de *scores* informáticos y herramientas de apoyo a decisiones clínicas es uno de los principales avances de la medicina personalizada. La utilización de algoritmos basados en Machine Learning puede ser un complemento muy útil a los *scores* clásicos para la valoración de recurrencia de FA.

1.4.2.1. Mejora de la salud

La definición de las variables relacionadas con un mayor riesgo de recurrencia de FA puede inspirar un mayor control de los factores de riesgo modificables. La medicina predictiva tiene el mayor impacto en la mejora de la salud de la población.

La primera versión de la herramienta informática (después de ser validada en un futuro estudio) podría ser utilizada a nivel clínico para optimizar la gestión de los diagnósticos y tratamientos de pacientes con una nueva aparición de FA. Las aplicaciones basadas en herramientas de Aprendizaje Automático pueden resultar muy útiles para complementar los *scores* clásicos.

1.4.3. Posibilidad de generalización de los resultados

La implementación de la herramienta de predicción podrá ser utilizada por la OSI del hospital al que pertenecen los pacientes estudiados y otros centros serían invitados a facilitar sus datos con el objetivo de alimentar el algoritmo de aprendizaje automático y hacerlo más competente en una población más heterogénea. En una segunda fase, podría ser implementado tanto en centros de la red de salud vasca como, posteriormente, en la red de salud española.

2. Marco teórico y Estado del arte

El proyecto se sustenta en ciertas consideraciones teóricas que es importante aclarar. Esta sección está dedicada a ello. Así, se comienza por introducir terminología y definiciones básicas referentes al área de la Inteligencia Artificial. También se presentarán trabajos del área médica que han empleado tanto técnicas tradicionales de Machine Learning como técnicas de Deep Learning, lo que nos dará pie a poder discutir sobre las ventajas e inconvenientes que presenta cada una de esa alternativas. Por último, se presentará una serie de *scores* clínicos utilizados para la predicción de recurrencia de FA, con los que podremos hacernos una idea de las variables predictoras más determinantes y resultados obtenidos en esos modelos, para así realizar una comparativa tras finalizar nuestra tarea.

2.1. Terminología básica

La **Inteligencia Artificial (IA)** es el procesamiento de los datos basado en máquinas para lograr objetivos que normalmente requieren la función cognitiva humana. En la era moderna, la IA es capaz de analizar conjuntos de datos densos y proporcionar el potencial para clasificar patrones complejos y representaciones novedosas de datos más allá de la interpretación humana directa.

El **Machine Learning (ML)** es una subdisciplina de la IA y emplea algoritmos para aprender patrones empíricamente desde los datos. ML es capaz de identificar relaciones no lineales e interacciones de grado alto entre variables, lo cual puede ser complejo para las estadísticas tradicionales.

El **Deep Learning (DL)** es una forma de ML capaz de trabajar con grandes conjuntos de datos y ofrecer un gran poder computacional para tomar decisiones eficientes sobre datos complejos. Ya se ha observado el éxito de ML y DL en múltiples disciplinas como el procesamiento del lenguaje, gaming, visión por computador, diseño, industria y áreas científicas. En el ámbito médico, se ha usado DL para el análisis de imágenes médicas, genómicas computacionales, análisis de señales fisiológicas, representación de datos médicos y predicción de enfermedades.

Features (Rasgos/Características)

Para que la IA sea capaz de interpretar los datos es necesario que estén formados por vectores de características (*feature vectors*). Estos rasgos (*features*) tienen que ser interpretables matemáticamente para que puedan ser computados por los algoritmos de ML. Los rasgos pueden ser de diferentes tipos: numéricos, categóricos, binarios... En estudios de ML tradicional que no usan DL, los rasgos son escogidos a mano y están diseñados por humanos para medir atributos específicos y relevantes del conjunto de datos, como puede ser las variables clínicas almacenadas en la historia médica electrónica. Sin embargo, en DL moderno, es posible computar y seleccionar automáticamente los rasgos relevantes del conjunto de datos en crudo.

En ML tradicional se requiere un conocimiento de dominio significativo y una experiencia de procesamiento de datos para entrenar los modelos. En DL, sin embargo, se aprenden rasgos abstractos directamente de los datos brutos.

En DL es muy habitual el uso de redes neuronales artificiales, que consisten en diferentes capas de nodos (capa de entrada, capas ocultas y capa de salida) que manipulan y transforman los datos de entrada, creando una representación de los datos, que finalmente conecta los datos de entrada con las salidas.

Diferentes capas de las redes neuronales aprenden automáticamente rasgos abstractos representativos de los datos. Una red bien diseñada y bien entrenada puede producir resultados robustos en muchas aplicaciones, sin la necesidad de un conocimiento de dominio significativo. Muchas tareas que se creían limitadas a la operación del humano debido a la complejidad de los datos son ahora automatizadas mediante métodos de DL.

2.1.1. IA en el presente proyecto

El presente proyecto es una tarea de ML supervisado. El conjunto de entrenamiento con el que se entrena el algoritmo de ML consiste en vectores de rasgos de entrada (*input*) y su correspondiente etiqueta (*output*). Es decir, se quiere predecir la variable de salida (*output*) de nuevos datos y, para ello, se entrena un modelo con datos en los que la variable de salida ya es conocida. Concretamente, se va a realizar una clasificación es binaria, ya que se predice si los pacientes continúan con un diagnóstico de fibrilación auricular (1) o no (0).

2.2. Desafíos en ML tradicional y DL

A continuación se va a realizar una lista de los desafíos que proponen las diferentes metodologías de aprendizaje automático supervisado: Machine Learning tradicional y Deep Learning. El objetivo de esta sección es dar a conocer las ventajas y limitaciones de estas dos técnicas, y llegar a unas conclusiones que nos permitan decidir el método más adecuado para nuestra tarea. Esta sección se complementará con ejemplos de estudios que realizan comparativas entre estas técnicas, los cuales se encuentran en apartado *Antecedentes en el área médica*.

- Todos los ítems del conjunto de datos tienen que ser previamente etiquetados, lo que puede ser muy laborioso y limitar la cantidad de datos disponibles para el análisis.
- Puede ocurrir que la proporción de las clases no esté balanceada. Por lo tanto, el modelo predictivo podría tener una gran exactitud (*accuracy*) en predicciones de la clase predominante en los datos de entrenamiento, pero peores resultados en el resto. En estos casos, las métricas como AUC o *F-score* pueden ayudar más en la evaluación. Otra alternativa es hacer un muestreo para balancear las clases, ya sea eliminando ejemplos de las clases mayoritarias (*undersampling*) o replicando casos de las minoritarias (*oversampling*).
- Puede ocurrir sobreajuste (*overfitting*), cuando el modelo aprende ruido y fluctuaciones aleatorias como relaciones falsas entre los datos de entrenamiento, lo cual dará mejores resultados con los datos de entrenamiento, pero peores con los de test. El *overfitting* puede ocurrir cuando no hay ejemplos de entrenamiento suficientes, hay excesivo ruido en los datos de entrenamiento o la complejidad del modelo es excesiva. Se puede combatir mejorando el conjunto de entrenamiento en cuanto a mayor número de ejemplos, mayor diversidad o calidad en los datos, o reduciendo la complejidad del modelo.
- En ML tradicional, puede resultar un problema la complejidad del modelo, causada por el espacio de rasgos de grandes dimensiones, el cual requiere una gran cantidad de datos para representar adecuadamente la variabilidad de las combinaciones de rasgos. Aunque el número óptimo de variables depende mucho del problema, un ratio 10:1 de ejemplos por variables es un buen punto de partida. Alternativamente, se puede reducir el espacio de rasgos antes del entrenamiento con la selección de rasgos.
- En DL, se puede simplificar la complejidad del modelo reduciendo el número de nodos o capas en la red neuronal. También, se puede dejar de entrenar la red cuando el error en el conjunto de validación comienza a aumentar (*early stopping*), para que el modelo no aprenda ruido.
- En problemas de complejidad no muy alta o con conjuntos de datos pequeños, los algoritmos de Deep Learning no tienen por qué ofrecer mejores resultados que un regresor tradicional con variables correctamente seleccionadas. DL es apropiado cuando se tiene un gran número de datos con muchos rasgos donde el reconocimiento de patrones es complejo.
- Una limitación de ML, y sobre todo de DL, es la dificultad de interpretar la contribución de los rasgos en el modelo (*black box nature*). La mala interpretabilidad es generalmente aceptable en

tareas sin ramificaciones clínicas, como la segmentación automatizada de imágenes. Sin embargo, la interpretabilidad es especialmente importante cuando la toma de decisiones se basa en los resultados de un modelo de IA. Es importante tener una herramienta interpretable si se trata con toma de decisiones médicas o nuevas perspectivas de enfermedades.

- Otra desventaja significativa de DL es que necesita un gran almacenamiento para los datos y una gran infraestructura computacional para aprender modelos de manera eficiente. A medida que los modelos se vuelven más complejos para incorporar información, por ejemplo, al haber falta de datos, el poder computacional crece en lugar de reducirse. Por ello, a pesar del gran aumento en las capacidades computacionales y la disminución de costos, puede suponer un gran esfuerzo financiero desarrollar y mantener la infraestructura computacional que requiere entrenar modelos de DL.

2.3. Antecedentes en el área médica

En esta sección se va a presentar el trabajo *Chen et al. [12]*, el cual recoge varios estudios del área médica en los que se emplean ML tradicional y DL. En él se tratan los puntos mencionados en el apartado anterior (*Desafíos en ML tradicional y DL*) y su objetivo es hacer una comparativa de estas dos técnicas en busca de la más apropiada para herramientas de IA en el ámbito clínico. El análisis de los antecedentes en estudios médicos nos ayudará a sacar conclusiones para la elección de los métodos más adecuados para nuestra tarea.

En el trabajo *Chen et al. [12]*, se han revisado los siguientes estudios:

- Predicción de sangrado postoperatorio después de cirugía colorrectal (CRS-PSC) [10].
- Predicción del diagnóstico, remisión y reaparición del asma infantil (A-DRR) [74].
- Predicción del tiempo hasta el primer tratamiento para los pacientes diagnosticados con leucemia linfocítica crónica (CLL-TFT) [11].
- Predicción de la mortalidad en la UCI (ICU-M) utilizando un conjunto de datos público [74] / [63].
- Predicción del uso indebido de opioides (Opioid) [9].

En la siguiente tabla se muestra para cada uno de los estudios el tamaño del conjunto de datos y el número de variables:

Conjunto de datos	# de ejemplos	# de variables
CRS-PSC	13399	117
A-DRR	4013	51
CLL-TFT	737	31
ICY-M	4000	41
Opioid	142377	836

2.1. Cuadro: Conjuntos de datos del estudio [2]

En cada caso, se han comparado diferentes algoritmos tradicionales de ML como regresión lineal (LR), redes Bayesianas (BN), *support vector machine* (SVM), *random forest* (RF) y *gradient boosting machine* (GBM) con redes neuronales DL incluyendo el perceptrón multicapa (MLP) y *long short-term memory* (LSTM). Los hiperparámetros de cada modelo han sido ajustados usando *Cross-Validation*. En los modelos MLP, específicamente, se han ajustado el número de capas ocultas y el número de nodos en cada capa oculta. Y para los modelos MLP y LSTM se han ajustado hiperparámetros como la tasa de aprendizaje (*learning rate*), la función de activación, la función de pérdida (*loss function*) y el número de iteraciones de entrenamiento (*epochs*). La métrica a optimizar en los modelos MLP y LSTM ha sido el *accuracy*. El área bajo la curva AUC se ha utilizado para evaluar el rendimiento de los diferentes modelos.

Table 2. Predictive power of each model and their associated training time			
App	Model	AUROC	Training time (s)
CRS-PSC	LR	0.735 ± 0.004	5.3
	BN	0.765 ± 0.004	48.6
	SVM	0.781 ± 0.003	33.1
	RF	0.812 ± 0.002	9.8
	GBM	0.822 ± 0.001	13.8
	MLP	0.795 ± 0.003	219
	LSTM	0.805 ± 0.004	703
	LR	0.875 ± 0.044	5.3
A-DRR	SVM	0.884 ± 0.011	23.1
	RF	0.947 ± 0.012	10.8
	GBM	0.945 ± 0.011	23.8
	LSTM	0.845 ± 0.034	301.4
CLL-TFT	LR	0.795 ± 0.038	2.3
	SVM	0.843 ± 0.036	5.7
	RF	0.924 ± 0.004	3.3
	GBM	0.817 ± 0.009	5.4
ICU-M	LSTM	0.805 ± 0.017	46.5
	LR	0.476 ± 0.074	4.3
	SVM	0.556 ± 0.051	12.7
	RF	0.618 ± 0.048	9.8
Opioid	GBM	0.681 ± 0.037	16.4
	LSTM	0.646 ± 0.043	243.5
	LR	0.907 ± 0.002	NA
	SVM	0.904 ± 0.002	NA
Opioid	RF	0.875 ± 0.003	NA
	MLP	0.909 ± 0.002	NA
	LSTM	0.909 ± 0.002	NA
	Bolded values indicate best achieved metric for each project. Training times were not measured during the Opioid project		

2.1. Figura: Resultados del estudio *Chen et al.* [12]

Como se puede ver en la Figura 2.1, en la mayoría de los casos, los métodos de ML tradicionales ofrecen un mejor rendimiento comparado con las alternativas de DL. Esto puede atribuirse al tamaño pequeño del conjunto de datos de entrenamiento en la mayoría de los modelos. Cuando aumenta la cantidad de datos, los métodos de DL se convierten más competitivos, aunque solo en el conjunto de datos *Opioid* (más de 100.000 datos) un algoritmo de DL da mejores resultados que uno de ML tradicional y tampoco puede superar significativamente a los algoritmos tradicionales.

Tamaño de la muestra

Table 3. Predictive power of each model based on percentage of data used to train						
% of Data (N)	LR	BN	SVM	RF	GBM	MLP
10 (1340)	0.728	0.748	0.731	0.776	0.804	0.742
20 (2680)	0.727	0.749	0.758	0.797	0.811	0.762
40 (5360)	0.730	0.753	0.773	0.798	0.809	0.778
60 (8039)	0.732	0.759	0.772	0.801	0.812	0.786
80 (10,719)	0.731	0.764	0.779	0.808	0.818	0.791
100 (13,399)	0.735	0.765	0.781	0.812	0.822	0.795

2.2. Figura: Resultados cambiando el tamaño de la muestra de entrenamiento del estudio CRS-PSC [10]

También se demuestra lo anterior en la Figura 2.2, donde se comparan los resultados obtenidos con el modelo CRS-PSC con diferentes cantidades de datos. Los algoritmos de ML tradicional parecen tener un límite superior en cuanto a *accuracy* mientras que el modelo LSTM parece capaz de mejorar aún más el rendimiento con el aumento de datos.

Sin embargo, en los estudios que trabajan con datos médicos, no siempre es fácil obtener información de una gran cantidad de pacientes, ya que estos estudios suelen trabajar con muestras limitadas y dependen de las instituciones encargadas de proveer la información de estos pacientes.

Dispersión de los datos (*missing values*)

Además de la falta de pacientes, otro problema muy común en ML con datos médicos es la dispersión de los datos. En muchas ocasiones, no se tiene una ficha completa del paciente en un momento dado, es decir, es normal encontrarse con valores desconocidos (*missing values*) en algunas variables al tratarse de información real de pacientes. Los humanos pueden imputar esos valores de manera consecuente a través de otra información. Sin embargo, eso no es posible para los algoritmos de ML, aunque sí hay algunos como las redes neuronales recurrentes (RNN) capaces de trabajar con *missing values*.

La estrategia más común a la hora de tratar con *missing values* es la imputación (por ejemplo, imputando la media), pero se corre el riesgo de sesgar los datos o requiere una alta capacidad computacional (por ejemplo, imputación basada en modelos *random forest*).

Coste computacional

Según el estudio *Chen et al. [12]*, todos los modelos menos el Opioid (más de 100.000 datos) fueron completados en un único equipo de escritorio con una CPU Intel Core i5-4750s relativamente barata y 8GB de memoria. El proyecto Opioid fue completado en un equipo con una CPU i5-4590S y 16GB de memoria. Como se ha visto en la Figura 2.1, los métodos de DL requieren mucho más tiempo que los demás. Aunque no se han registrado los tiempos de entrenamiento en el proyecto Opioid, el estudio dice que se necesitaron varias horas para entrenar el modelo con el uso de la CPU especificada. Infraestructuras computacionales con uso de GPUs acelerarían el desarrollo del modelo, pero son un gasto significativo, sobre todo para las pequeñas organizaciones sanitarias.

Interpretación del modelo

Los modelos predictivos relacionados con la medicina requieren un grado de interpretabilidad, ya que conocer los rasgos específicos que más influyen en una predicción puede ser importante para la toma de decisiones clínicas. Algunos algoritmos de ML como *random forest* tienen medidas especiales que ayudan al usuario a conocer cómo contribuyen las variables en la predicción dada, de manera similar a los pesos y *p-values* en regresión lineal. Como ya se ha dicho, los algoritmos de DL son todavía bastante difíciles de interpretar (*black box nature*).

Conclusiones

Tras el análisis de del estudio *Chen et al. [12]*, podemos concluir en que DL es una herramienta extremadamente poderosa para el aprendizaje complejo, sin embargo, no es una herramienta exhaustiva para todas las aplicaciones relacionadas con análisis médicos. Problemas muy comunes en estudios con datos médicos como una pequeña cantidad de datos, una alta dispersión o una baja calidad pueden hacer que no sean tan efectivos los métodos de DL.

Muchas veces es fácil confundir la capacidad de generalizar de los algoritmos de DL para crear un modelo capaz de analizar todo tipo de datos. Sin embargo, otros algoritmos de ML más tradicionales pueden ser mucho más eficientes computacionalmente, ofrecer modelos más fáciles de interpretar y, además, ser más precisos.

En conclusión, los investigadores de IA en el área clínica no deberían dejarse llevar por los grandes avances que ofrece DL a priori. Aunque es muy útil para determinadas tareas, como clasificar imágenes médicas, DL no es adecuado para todos los problemas de datos clínicos.

Por otro lado, la mayoría de estudios de IA en el ámbito médico se llevan a cabo con datos independientes de la institución que se encarga del proyecto. Sin embargo, recolectar, combinar y etiquetar datos entre diferentes instituciones sería totalmente beneficioso para la IA. Esta colección de datos debería ir más allá de las variables clínicas en las historias médicas electrónicas (EMR). El almacenamiento de

datos armonizados en repositorios accesibles para los investigadores haría avanzar rápidamente la IA en este campo.

2.4. Antecedentes de IA en fibrilación auricular

En la sección anterior se ha realizado un análisis de estudios que emplean técnicas de Inteligencia Artificial en el área médica. En esta sección se va a realizar un análisis más concreto sobre trabajos que utilizan IA para la predicción de riesgo de fibrilación auricular. De esta manera, se obtendrá una idea de los algoritmos y variables predictoras utilizadas en modelos predictivos de recurrencia de FA ya existentes, así como se conocerán los resultados de estos modelos, los cuales podrán compararse con los obtenidos al finalizar nuestro proyecto.

El estudio de revisión *Deng et al. [17]* proporciona una descripción general de *scores* (modelos predictivos) de riesgo y recurrencia de FA tras los resultados de la estrategia de control de ritmo (como puede ser la ablación por catéter, AC) o progresión de FA. En el Anexo II de la memoria se encuentra información más detallada de este y otros estudios recientes relacionados con herramientas de IA para la predicción de riesgo de aparición y/o recurrencia de FA.

Los scores analizados por el estudio *Deng et al. [17]* son los siguientes:

- **El score HATCH [16]**

El *score HATCH* fue el primer modelo predictivo diseñado para la progresión de FA paroxística o persistente. En el estudio original, se incluyeron 1219 pacientes y se observaron durante más de un año. Se utilizaron las siguientes variables predictoras: insuficiencia cardiaca (H), mayor edad (A), accidente isquémico transitorio previo (AIT) o ictus (T), enfermedad pulmonar obstructiva crónica (EPOC, C), e hipertensión (H). El *score HATCH* va de 0 a 7 puntos.

Según el modelo del estudio original, casi el 50 % de los pacientes con *score* > 5 progresaron a FA persistente, pero solo el 6 % de los pacientes con *score* 0 experimentaron esa progresión. El estudio posterior *Kochhäuser et al. [33]* sobre pacientes de FA que estaban esperando una AC determinó que el *score HATCH* tenía un pobre poder predictivo para la progresión de FA. Este otro estudio, *Potpara et al. [55]*, demostró que tenía un índice C de 0.6 en una cohorte de pacientes solitarios con FA durante un período de seguimiento de 12 años.

- **El score ALARMEc [75]**

El objetivo en un principio era un *score* para predecir el resultado de FA tras AC en un estudio que comparó el *score ALARMEc* con los *scores CHADS₂* y *CHA₂DS₂-VASc* para la estratificación del riesgo de accidente cerebrovascular.

ALARMEc incluyó cinco variables: tipo de FA, tamaño de la aurícula izquierda, insuficiencia renal, síndrome metabólico y miocardiopatía. El *score ALARMEc* se evaluó en solo 213 pacientes con FA paroxística o FA no paroxística que se sometieron a AC repetida. Después de un seguimiento de hasta 60 meses, solo la puntuación *ALARMEc* (AUC 0.657, P < 0,0001), pero no la puntuación *CHADS₂* (AUC 0.533, P = 0.413) o *CHA₂DS₂-VASc* (AUC 0.519, P = 0.641), predijo los resultados después de AC.

- **El score BASE-AF2 [7]**

El *score BASE-AF2* utilizó seis variables: índice de masa corporal (IMC) $28kg/m^2$, dilatación auricular > 40mm, tabaquismo actual, recurrencia temprana, duración de la FA > 6 años y tipo de FA no paroxística.

Se utilizó una muestra de 236 pacientes con FA paroxística que se sometieron a crioablación. El *score* va de 0 a 6 puntos, teniendo en cuenta un *score* ≥ 3 puntos como predictor independiente de recurrencia de FA.

- **El *score* APPLE [38]**

El *score* APPLE se obtuvo a partir de una cohorte de pacientes con FA de Alemania y se probó en una cohorte de validación de los EEUU. Se utilizaron cinco variables: edad > 65 años, FA persistente, eGFR alterada ($< 60\text{ml}/\text{min}/1,73\text{m}^2$), diámetro de la AI $\geq 43\text{mm}$, y EF $< 50\%$.

Los análisis de regresión logística mostraron que los *score* APPLE, CHADS₂ y CHA₂DS₂-VASC fueron predictores significativos de recurrencia de FA entre 3 y 12 meses. Según el análisis de la curva ROC, el *score* APPLE tuvo un mejor valor predictivo en comparación con CHADS₂ y CHA₂DS₂-VASC (índice C de 0.634 frente a 0.538 y 0.542, respectivamente, ambos P < 0,001).

- **El *score* CAAP-AF [71]**

El *score* CAAP-AF se realizó inicialmente en una cohorte de derivación de 1125 pacientes con FA y se probó en una cohorte de validación de 937 pacientes con FA que se sometieron a la primera AC en el mismo centro. Se tuvieron en cuenta dos tipos de FA: paroxística y no paroxística. Se utilizaron las siguientes variables: arteriopatía coronaria (CAD), diámetro de la aurícula izquierda, edad, FA persistente o de larga duración, número de fármacos antiarrítmicos fallidos y sexo.

En la cohorte de validación, el porcentaje de pacientes sin FA fue idéntico al observado en la cohorte de derivación (índice C de 0.650 frente a 0.691). Es de destacar que el *score* CAAP-AF se basó en la experiencia de un solo centro y la recurrencia de la FA se detectó solo mediante Holter de 24h o 7 días, pero no mediante la implantación de un registrador de bucle.

- **El *score* MB-LATER [48]**

El *score* MB-LATER se propuso para predecir la recurrencia muy tardía (> 12 meses) de FA después de la AC. Se utilizaron las siguientes variables: sexo masculino, bloqueo de rama, tamaño de la aurícula izquierda $\geq 47\text{mm}$, tipo de FA (paroxística, persistente o persistente de larga duración) y FA precoz recurrente.

El MB-LATER se derivó de una pequeña cohorte retrospectiva y se comparó con otras puntuaciones clínicas (APPLE, ALARMEc, BASE-AF2, CHADS₂, CHA₂DS₂-VASC o HATCH). En comparación con estas puntuaciones, el MB-LATER demostró un mejor valor predictivo (AUC de 0.782 frente a 0.716, 0.671, 0.648, 0.522, 0.519 o 0.583) y una mejor identificación de los pacientes con recurrencia de FA muy tardía mediante el análisis de la curva de decisión (DCA).

- **Los *scores* CHADS₂, CHA₂DS₂-VASC y R₂CHADS₂ [36]**

Son *scores* de riesgo para predecir el ictus y la tromboembolia. Dado que los componentes de los factores de riesgo de estos *scores* son factores de riesgo cardiovascular comunes, pueden estar relacionados con los resultados del control del ritmo o la progresión de la arritmia.

Estas son las variables que utilizaron:

- CHADS₂: insuficiencia cardíaca congestiva, hipertensión, edad ≥ 75 años, diabetes mellitus y accidente cerebrovascular/accidente isquémico transitorio.
- CHA₂DS₂-VASc: insuficiencia cardíaca congestiva, hipertensión, edad ≥ 75 años, diabetes mellitus, accidente cerebrovascular/accidente isquémico transitorio, enfermedad vascular, 65-74 años y sexo femenino.
- R₂CHADS₂: disfunción renal, insuficiencia cardíaca congestiva, hipertensión, edad ≥ 75 años, diabetes mellitus y accidente cerebrovascular/accidente isquémico transitorio.

Según el estudio *Letsas et al. [41]*, los *scores* CHADS₂ y CHA₂DS₂-VASc se asociaron con la recurrencia de la FA en pacientes con FA paroxística sometidos a un solo procedimiento de AC. CHADS₂ (AUC 0.644) y CHA₂DS₂-VASc (AUC 0.627) tuvieron el valor predictivo más alto para la recurrencia de FA.

Otro estudio, *Jacobs et al. [29]*, que incluyó pacientes con FA paroxística y FA persistente, informó que los *scores* de CHADS₂ (HR 1.19, P <0.001) y CHA₂DS₂-VASc (HR 1.15, P <0.0001) fueron buenos para estratificar a los pacientes durante 5 años después de la AC, siendo el score CHA₂DS₂-VASc (HR 1.13, P = 0.001) superior al CHADS₂ para predecir la recurrencia de FA.

Conclusiones

A continuación se listan algunas de las conclusiones sacadas en el anterior estudio de comparación de *scores* de recurrencia de FA. Entre otras, obtenemos variables predictoras importantes y resultados de modelos predictivos que pueden servirnos de referencia en la realización de nuestra tarea.

- Entre las variables predictoras más utilizadas y/o más determinantes en los *scores* descritos se encuentran las siguientes: diámetro de la aurícula izquierda, subtipo de FA, recurrencia temprana, edad, insuficiencia cardiaca y realización de crioablación para la AC. Se prestará atención a la relevancia de estas variables en nuestra tarea.
- Los pacientes de estos estudios han tenido un seguimiento al menos de 12 meses. Para detectar la recurrencia de FA se utilizaron síntomas de arritmia con ECG de 12 derivaciones y monitorización de Holter continuo.
- Valoración de los resultados obtenidos:
 - Los *scores* CHADS₂, CHA₂DS₂-VASc y R₂CHADS₂, que no son predictores de riesgo relacionadas con el ritmo cardíaco, se probaron en diferentes cohortes con resultados contradictorios.
 - El *score* HATCH no tuvo un valor significativo en la predicción de la recurrencia después de AC.
 - En los estudios que describen ALARMEc y CAAP-AF, también se evaluó el valor predictivo de los *score* para los resultados repetidos de la AC. Las pruebas de validación de ALARMEc fueron contradictorias y CAAP-AF no fue validado por ninguna otra cohorte.
 - El *score* APPLE se ha validado en al menos tres cohortes (AUC de 0.634, 0.617 y 0.716, todas P < 0,05), incluida la cohorte de derivación del *score* MB-LATER. Los factores de riesgo de APPLE son índices clínicos de fácil adquisición, lo que hace que sea un buen predictor de la recurrencia de FA después de la AC.
 - Para la recurrencia temprana basada en factores clínicos después del procedimiento de AC, los *scores* BASE-AF2 y MB-LATER podrían usarse.
 - Se demostró que el *score* MB-LATER tiene un mejor valor predictivo para recurrencia muy tardía que otros *scores* en un estudio de cohorte prospectivo pequeño.

3. Planificación y análisis del proyecto

Una vez hecho un análisis del marco teórico y el estado del arte en el que se encuadra el proyecto, en esta sección se presentará la planificación general para desarrollar el mismo. Se dará paso a la arquitectura del proyecto, las herramientas utilizadas, el alcance del proyecto, la planificación del tiempo (diagrama de Gantt), la evaluación económica y el análisis de riesgos.

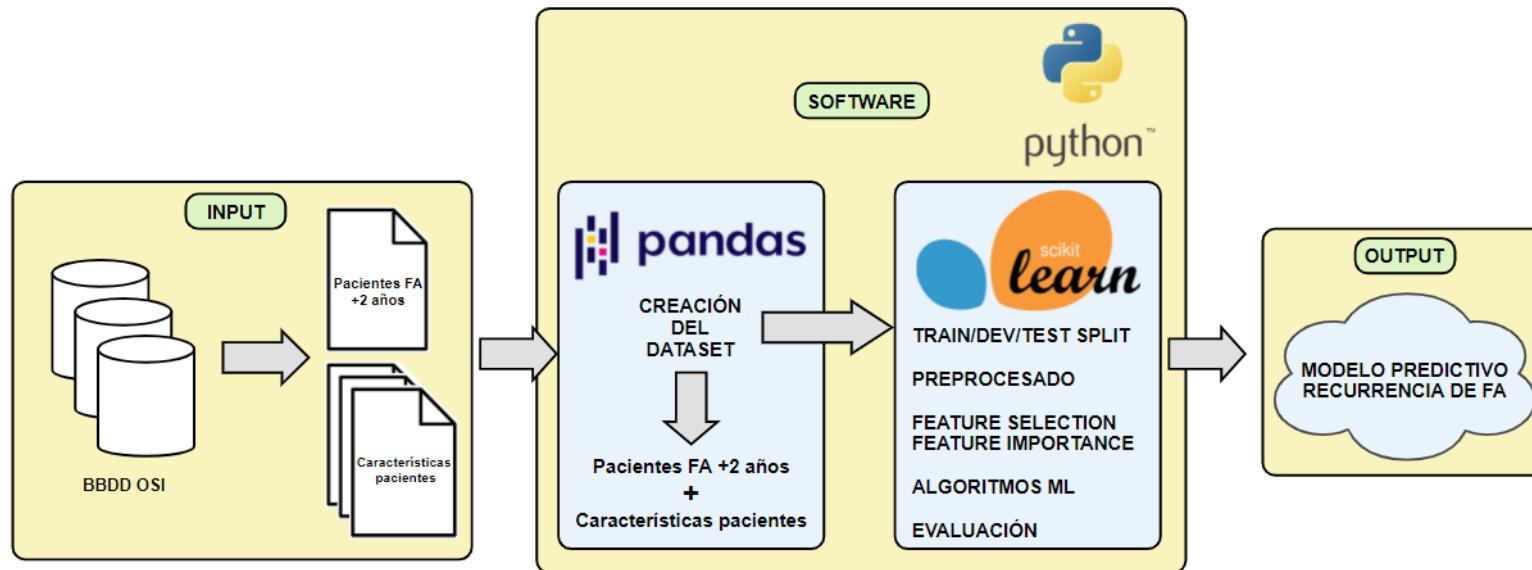
3.1. Arquitectura

La arquitectura del proyecto muestra la estructura, los procesos y las herramientas fundamentales para la creación del producto a desarrollar. La Figura 3.1 muestra la arquitectura del software desarrollado, en la cual aparecen el *input*, la estructura y herramientas utilizadas para el desarrollo de la aplicación y el *output*. Recordamos que el objetivo es crear un modelo predictivo de recurrencia de fibrilación auricular a partir de datos de pacientes con un primer evento de esta arritmia.

La entrada (*input*) está formada por varios documentos en formato *Libro de Excel (.xlsx)*. El documento principal contiene el código de identificación de cada uno de los pacientes con un primer episodio de FA y la variable de salida (*clase*) que indica si el paciente, tras aproximadamente 2 años, permanece en la arritmia o no. El resto de documentos contienen datos acerca de los pacientes (diagnósticos, pruebas de laboratorio, tratamientos, datos demográficos...) que deben extraerse y formar los vectores de rasgos (*features*) para cada paciente con un primer evento de FA, y de esta manera, crear el conjunto de datos para ML (*dataset*).

En el segundo bloque se muestra la estructura y las herramientas necesarias para la creación del software. Todo el software ha sido desarrollado en el lenguaje de programación **Python**. La arquitectura del software se divide en dos bloques. Por un lado, para la creación del conjunto de datos (*dataset*) se ha utilizado la librería de Python especializada en el manejo y análisis de estructuras de datos **Pandas**. Mediante esta librería ha sido posible extraer y combinar los datos de los pacientes recibidos, creando un *dataset* estructurado para Machine Learning donde cada ejemplo es el vector de rasgos del paciente correspondiente con su variable de salida (*clase*). Por otro lado, para todas las tareas relacionadas con el aprendizaje automático se ha utilizado la librería de software libre para Python **Scikit-learn**. Mediante esta herramienta se han llevado a cabo todas las tareas para la creación del modelo predictivo: división del conjunto de datos, preprocesado de datos, selección de variables, entrenamiento de modelos, evaluación...

Por último, la salida (*output*) es el modelo predictivo óptimo de recurrencia de FA, entrenado con los datos de entrada y mediante algoritmos de Scikit-learn.



3.1. Figura: Arquitectura del software

3.2. Herramientas utilizadas

A continuación se detallan las herramientas utilizadas en el desarrollo del proyecto. Algunas de las herramientas ya han sido utilizadas en proyectos previos, mientras que otras han sido utilizadas por primera vez y ha sido necesario invertir tiempo en buscar información y aprender a utilizarlas.

- **Python:** Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, y es multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y programación funcional. Ha sido utilizado durante todas las tareas de programación del proyecto, principalmente porque ofrece los siguientes paquetes de gran utilidad en el desarrollo del proyecto:
 - **Pandas:** Librería de Python especializada en el manejo y análisis de estructuras de datos. El conjunto de datos creado se ha guardado en un *DataFrame* (estructura de dos dimensiones, tabla). Además, se trabaja de manera muy eficiente a la hora de acceder, modificar e insertar datos, y permite leer y escribir fácilmente ficheros en formato CSV, Excel... Esta herramienta no se había utilizado en ningún proyecto previo, por lo que ha sido necesaria la búsqueda de información y el siguiente tutorial: <https://www.kaggle.com/learn/pandas>.
 - **Scikit-learn:** Librería de Python para aprendizaje automático de software libre. Incluye funciones para el preprocesado de datos, división del conjunto de datos, selección de variables, algoritmos de clasificación y esquemas de evaluación. También se ha usado junto con las librerías **Matplotlib** y **Seaborn** para la visualización de gráficos y diagramas. Esta herramienta no se había utilizado en ningún proyecto previo por lo que ha sido necesaria la búsqueda de información: https://scikit-learn.org/stable/user_guide.html
 - **Keras:** Librería de Python para el entrenamiento de algoritmos de Deep Learning como redes neuronales, convolucionales o recurrentes. Es muy útil como complemento a Scikit-learn cuando se necesita trabajar con algoritmos de Deep Learning. Se ha empleado en la tarea secundaria para la creación del modelo predictivo de recurrencia de FA añadiendo las características de los electrocardiogramas. Esta herramienta no se había utilizado en ningún proyecto previo por lo que ha sido necesaria la búsqueda de información: <https://keras.io/guides/>
- **PyCharm:** Es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos, parte del suite de herramientas de programación de JetBrains. Es uno de los entornos de desarrollo más completos para Python, por lo que se ha utilizado para la creación del software. PyCharm tiene compatibilidad integrada para librerías científicas y admite Pandas, Numpy y Matplotlib, entre otras.
- **Cacoo:** Es una herramienta de dibujo que permite crear una gran variedad de diagramas. Ha sido utilizada para crear los siguientes diagramas: Arquitectura del software, EDT y Diseño del software.
- **GanttProject:** Es una herramienta visual y fácil de usar para la creación de cronogramas. Ha sido utilizada para crear el diagrama de Gantt en el que se define la duración y la fecha de todas las tareas realizadas en el proyecto.
- **LaTeX:** Es un procesador de texto de lenguaje de marcado orientado a la creación de documentos escritos formales. Se basa en la utilización de *macros* de TeX para facilitar su uso, y es muy utilizado para la composición de artículos académicos, tesis y libros técnicos.
- **Overleaf:** Es un editor de texto online basado en LaTeX. Ha sido utilizado para la redacción de la memoria del proyecto.
- **Microsoft Office 365:** Es un paquete de softwares para la creación y gestión de documentos de ofimática (documentos de texto, hojas de cálculo...). Ha sido utilizado para escritura de documentos relacionados con la memoria del proyecto y la gestión de los datos de entrada (*input*) en formato de hoja de cálculo.
- **Blackboard Collaborate / Webex / Jitsi Meet / Zoom:** Son herramientas de videoconferencias en tiempo real que, en muchos casos, permiten agregar archivos, compartir aplicaciones y utilizar una pizarra virtual para interactuar. Han sido utilizadas para realizar las reuniones con el grupo de trabajo y con el cliente/data-manager.

- **Git:** Es el sistema de control de versiones más popular hoy en día, distribuido bajo licencias de código abierto. Git sigue una arquitectura distribuida, lo que significa que existe una copia del repositorio y su historial completo de cambios en el sistema de cada desarrollador. También es descentralizado, lo que posibilita que los cambios puedan realizarse de manera local para posteriormente ser sincronizados con el resto de copias. Ha sido utilizado para la gestión del código del software desarrollado.
- **GitHub:** Es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador y que incorpora el sistema Git de control de versiones. En esta plataforma los desarrolladores suben el código de sus aplicaciones y herramientas, y los usuarios no solo pueden descargarlas, sino también entrar a su perfil para leer sobre ellas o colaborar con su desarrollo. Ha sido utilizado para alojar el código del software desarrollado y gestionado mediante el sistema de control de versiones Git.

3.3. Alcance del proyecto

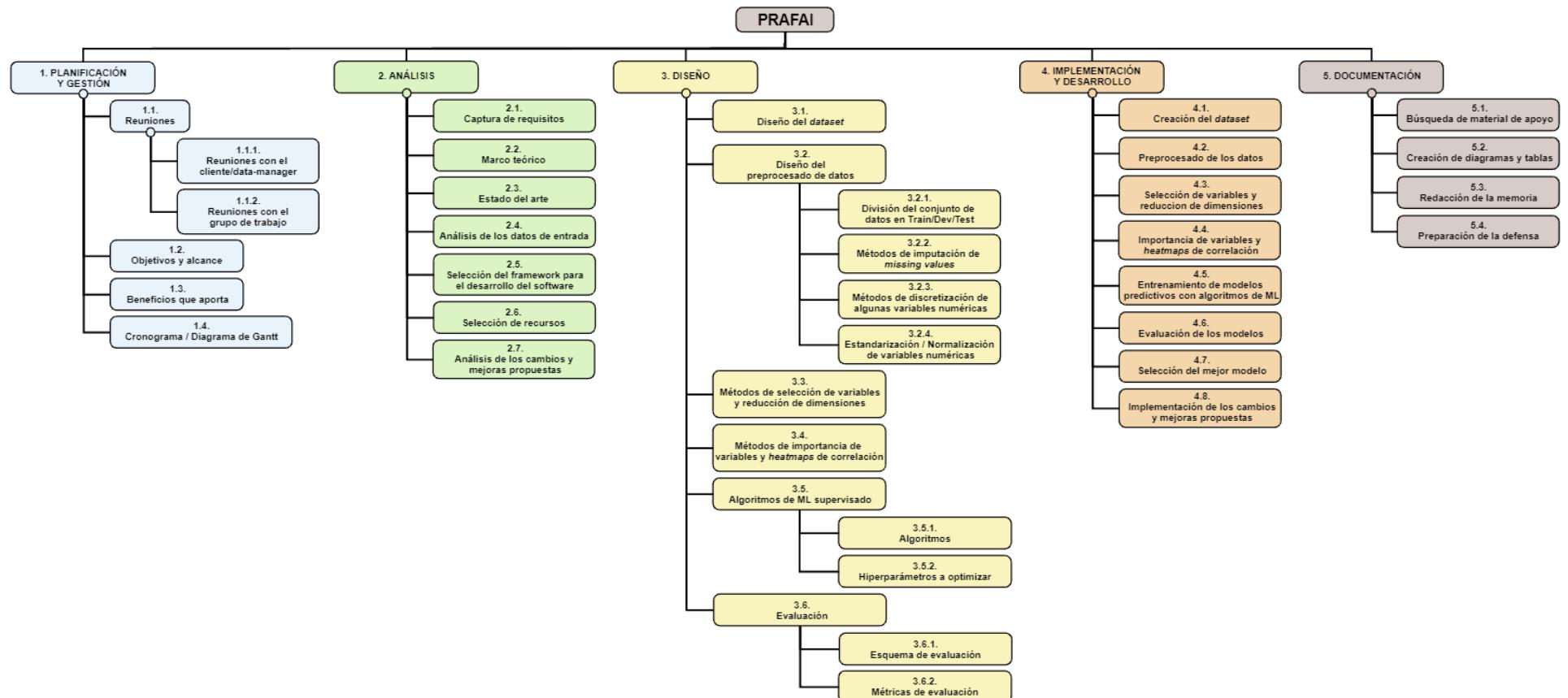
En esta sección se define el alcance del proyecto. Para ello, se ha realizado un diagrama EDT (Estructura de Descomposición de Trabajo) que consta de cinco bloques principales (ver Figura 3.2):

1. **Planificación y gestión:** Es la primera fase del proyecto, pero también se realizan tareas de planificación y gestión durante todo el proyecto. Forman parte de la gestión tareas como las reuniones realizadas, la lista de objetivos a definir y la planificación del tiempo.
2. **Análisis:** En este bloque se deben recoger y establecer los requisitos para el desarrollo del software por parte del cliente y decidir las herramientas y recursos adecuados para llevar a cabo el trabajo. También es la fase en la que se hace un análisis del marco teórico y estado del arte.
3. **Diseño:** Teniendo en cuenta las dos fases anteriores, se define la estructura del software para dar solución al problema. En nuestro caso, consta del diseño de las tareas de creación del conjunto de datos y el entrenamiento del modelo predictivo óptimo de recurrencia de FA.
4. **Implementación y desarrollo:** A este bloque pertenecen todas las tareas de programación para la creación del software según el diseño realizado en la fase anterior.
5. **Documentación:** Con esta fase concluye el proyecto y pertenecen todas las tareas que requieren una redacción de información sobre el proyecto. Las tareas de este bloque se van realizando durante el transcurso del proyecto.



3.2. Figura: Bloques principales del diagrama EDT

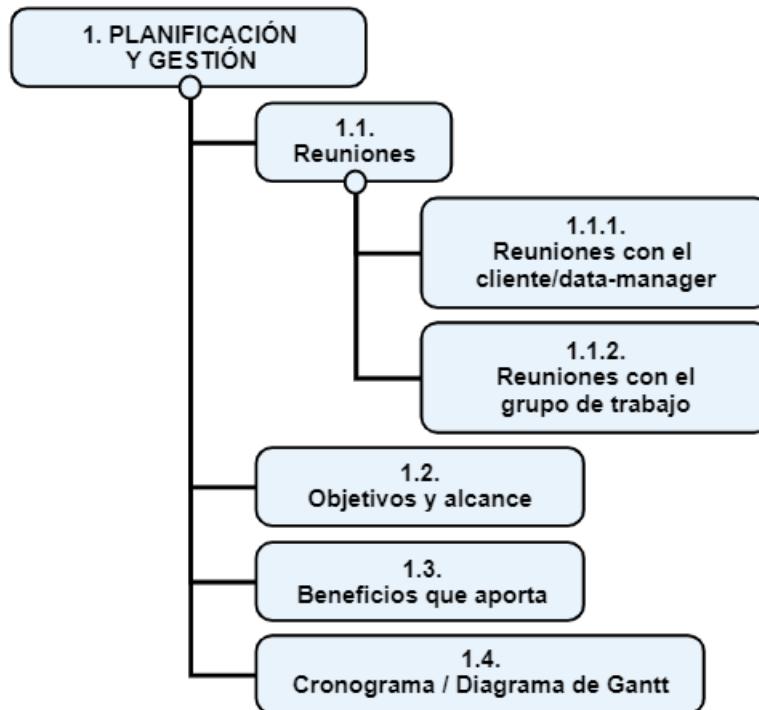
A continuación se muestra el diagrama EDT al completo, una vez desglosados los bloques a todos los niveles (ver Figura 3.3). Cada bloque y tarea en el EDT precede al siguiente, excepto algunas tareas que se realizan en paralelo a las demás durante todo el proyecto, como por ejemplo, las reuniones, el análisis de mejoras y cambios, y las tareas de documentación.



3.3. Figura: Diagrama EDT

3.3.1. Planificación y gestión

En este apartado se muestran las tareas de planificación y gestión realizadas durante el proyecto (ver Figura 3.4).



3.4. Figura: Diagrama EDT del bloque de planificación y gestión

En las siguientes tablas se muestran los detalles de cada tarea para este apartado.

1.1.1. Reuniones con el cliente/data-manager	
Paquete de trabajo:	Planificación y gestión
Esfuerzo estimado:	10 horas
Periodo estimado:	304 días (tarea <i>hammock</i>)
Descripción:	Reuniones con el cliente/data-manager del proyecto para dar a conocer el proyecto, su contexto, sus objetivos, captura de requisitos y los datos de entrada, así como para aclarar cualquier duda relacionada con el proyecto, pero fuera del ámbito de la informática.
Salidas/Entregables:	Actas con anotaciones de lo hablado durante las reuniones y lo acordado para futuras tareas a desarrollar.
Recursos necesarios:	Blackboard Collaborate / Webex / Jitsi Meet / Zoom y documentos de texto de Microsoft Office 365.

3.1. Cuadro: Reuniones con el cliente/data-manager

1.1.2. Reuniones con el grupo de trabajo
Paquete de trabajo: Planificación y gestión
Esfuerzo estimado: 18 horas
Periodo estimado: 304 días (tarea <i>hammock</i>)
Descripción: Reuniones con los tutores del proyecto para hacer un seguimiento del mismo con el fin de tener un control periódico. En estas reuniones se tratan temas como la resolución de dudas o problemas, modificaciones en la planificación o análisis del trabajo realizado en las etapas del proyecto.
Salidas/Entregables: Actas con anotaciones de lo hablado durante las reuniones y lo acordado para futuras tareas a desarrollar.
Recursos necesarios: Blackboard Collaborate / Webex / Jitsi Meet / Zoom y documentos de texto de Microsoft Office 365.

3.2. Cuadro: Reuniones con el grupo de trabajo

1.2. Objetivos y alcance
Paquete de trabajo: Planificación y gestión
Esfuerzo estimado: 5 horas
Periodo estimado: 3 días
Descripción: Fijar los objetivos del proyecto, así como indentificar, definir y estimar la duración de las tareas a realizar durante el proyecto.
Salidas/Entregables: Documentos con toda la información recopilada para crear los apartados <i>Objetivos</i> y <i>Alcance</i> de la memoria.
Recursos necesarios: Documentos de texto de Microsoft Office 365.

3.3. Cuadro: Objetivos y alcance

1.3. Beneficios que aporta
Paquete de trabajo: Planificación y gestión
Esfuerzo estimado: 5 horas
Periodo estimado: 3 días
Descripción: Identificar y definir los beneficios futuros que puede llegar a aportar la realización de este proyecto, ya sea en el ámbito de la medicina como el ámbito de la informática.
Salidas/Entregables: Documentos con toda la información recopilada para crear el apartado <i>Beneficios que aporta el trabajo</i> de la memoria.
Recursos necesarios: Documentos de texto de Microsoft Office 365.

3.4. Cuadro: Beneficios que aporta

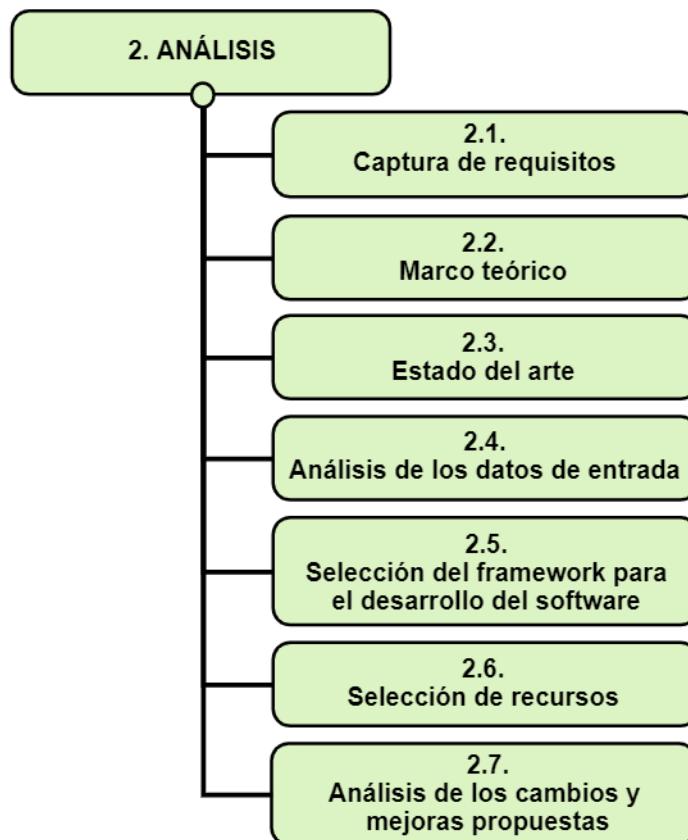
1.4. Cronograma / Diagrama de Gantt
Paquete de trabajo: Planificación y gestión
Esfuerzo estimado: 10 horas
Periodo estimado: 3 días
Descripción: Creación de un diagrama de Gantt para la planificación del tiempo, en el que se especifica la duración y la fecha de realización de cada tarea del proyecto.
Salidas/Entregables: Diagrama de Gantt.
Recursos necesarios: GanttProject.

3.5. Cuadro: Cronograma / Diagrama de Gantt

3.3.2. Análisis

En este apartado se muestran las tareas relacionadas con el análisis del marco teórico en el que se encuadra el proyecto, así como la captura de requisitos y el estado del arte. También se toman decisiones

sobre las herramientas y recursos que van a utilizarse (ver Figura 3.4).



3.5. Figura: Diagrama EDT del bloque de análisis

2.1 Captura de requisitos
Paquete de trabajo: Análisis
Esfuerzo estimado: 5 horas
Periodo estimado: 3 días
Descripción: Tras la primera reunión con el cliente/data-manager, se definen todas las especificaciones para el desarrollo del software: datos de entrada, tipo de aprendizaje, técnicas de preprocesado, técnicas de selección de variables, algoritmos de clasificación supervisada, esquemas de evaluación...
Salidas/Entregables: Documento de requisitos.
Recursos necesarios: Documento de texto de Microsoft Office 365.

3.6. Cuadro: Captura de requisitos

2.2. Marco teórico
Paquete de trabajo: Análisis
Esfuerzo estimado: 10 horas
Periodo estimado: 5 días
Descripción: Definir una base teórica sólida sobre la cual se va a llevar a cabo el desarrollo de proyecto. Se explican los conceptos, términos y el enfoque que se va a dar para solucionar el problema expuesto. En este proyecto, es necesaria una explicación acerca de la Inteligencia Artificial (IA), Machine Learning (ML) y Deep Learning (DP), así como una guía de realización de un estudio de ML supervisado dando a conocer qué son los rasgos o características y el diseño de un modelo predictivo y su evaluación, entre otros.
Salidas/Entregables: Documentos con toda la información recopilada para crear el apartado <i>Marco teórico</i> de la memoria.
Recursos necesarios: Navegador y documentos de texto de Microsoft Office 365.

3.7. Cuadro: Marco teórico

2.3. Estado del arte
Paquete de trabajo: Análisis
Esfuerzo estimado: 20 horas
Periodo estimado: 14 días
Descripción: Recopilar y presentar información sobre investigaciones y publicaciones que se han realizado relacionadas al proyecto. En este caso, se han buscado los proyectos más recientes y relevantes en los que se han estudiado modelos de riesgo de arritmias como FA. En concreto, el objetivo es conocer los resultados obtenidos y las técnicas utilizadas en proyectos del mismo ámbito: rasgos escogidos, métodos de preprocesado aplicados, algoritmos de ML utilizados, métricas de evaluación con las que se ha evaluado...
Salidas/Entregables: Documentos con toda la información recopilada para crear el apartado <i>Estado del arte</i> de la memoria.
Recursos necesarios: Navegador y documentos de texto de Microsoft Office 365.

3.8. Cuadro: Estado del arte

2.4. Análisis de los datos de entrada
Paquete de trabajo: Análisis
Esfuerzo estimado: 12 horas
Periodo estimado: 10 días
Descripción: Hacer un análisis de los datos de entrada recibidos: localizar los ejemplos que van a formar el conjunto de datos y sus variables de salida, localizar los datos para crear los vectores de rasgos, obtener el tamaño de la muestra, analizar las variables predictoras...
Salidas/Entregables: Anotaciones con la información recopilada para la creación del conjunto de datos.
Recursos necesarios: Navegador, documentos Excel con datos de entrada y documentos de texto con información de los datos de entrada.

3.9. Cuadro: Análisis de los datos de entrada

2.5. Selección del <i>framework</i> para el desarrollo del software
Paquete de trabajo: Análisis
Esfuerzo estimado: 2 horas
Periodo estimado: 1 día
Descripción: Comparar y elegir las aplicaciones más adecuadas para el desarrollo del software. En este caso, se debe elegir el entorno de programación del software.
Salidas/Entregables: <i>Framework</i> para el desarrollo del software.
Recursos necesarios: Navegador.

3.10. Cuadro: Selección del *framework* para el desarrollo del software

2.6. Selección de recursos
Paquete de trabajo: Análisis
Esfuerzo estimado: 3 horas
Periodo estimado: 1 día
Descripción: Comparar y elegir las herramientas y recursos más adecuados para el desarrollo del proyecto y su documentación: lenguaje de programación y librerías, aplicaciones para la creación de diagramas y cronogramas, aplicaciones para la redacción de la memoria...
Salidas/Entregables: Conjunto de herramientas y recursos seleccionados.
Recursos necesarios: Navegador.

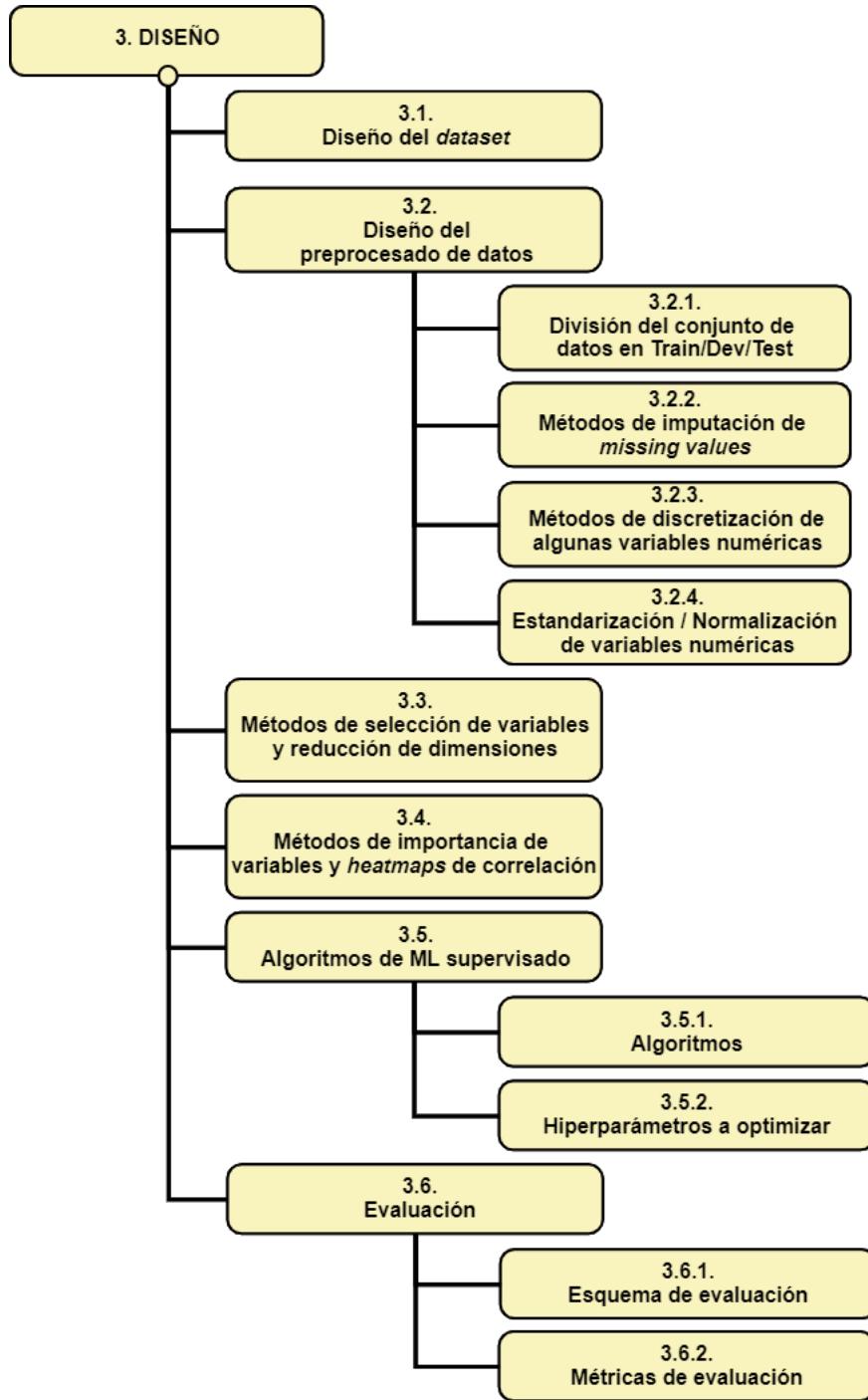
3.11. Cuadro: Selección de recursos

2.7. Análisis de los cambios y mejoras propuestas
Paquete de trabajo: Análisis
Esfuerzo estimado: 15 horas
Periodo estimado: 12 días
Descripción: Definir los cambios o mejoras propuestas por parte del desarrollador, tutores del proyecto o cliente, durante o una vez finalizado el desarrollo del software.
Salidas/Entregables: Anotaciones para realizar los cambios o mejoras propuestas.
Recursos necesarios: Documentos de texto de Microsoft Office 365.

3.12. Cuadro: Análisis de los cambios y mejoras propuestas

3.3.3. Diseño

En este apartado, una vez realizado el análisis del bloque anterior, se ha diseñado una solución válida y eficaz para alcanzar los objetivos del proyecto. El diseño se ha dividido en las tareas que se muestran en la Figura 3.6.



3.6. Figura: Diagrama EDT del bloque de diseño

3.1. Diseño del dataset
Paquete de trabajo: Diseño
Esfuerzo estimado: 10 horas
Periodo estimado: 5 días
Descripción: Diseñar la composición del conjunto de datos que se utilizará para el entrenamiento y evaluación del modelo predictivo desarrollado. En concreto, hay que diseñar los procedimientos para crear cada ejemplo uniendo todos los valores de los rasgos/variables del paciente con FA y el valor de la variable de salida (clase).
Salidas/Entregables: Diseño para la creación del conjunto de datos (<i>dataset</i>).
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.13. Cuadro: Diseño del *dataset*

3.2.1. División del conjunto de datos en Train/Dev/Test
Paquete de trabajo: Diseño
Esfuerzo estimado: 2 horas
Periodo estimado: 1 día
Descripción: Decidir los porcentajes para la división del conjunto de datos en tres subconjuntos: Train (ejemplos para el entrenamiento del modelo), Dev (ejemplos para la evaluación del modelo entrenado) y Test (ejemplos nunca vistos durante el entrenamiento para testear el modelo final).
Salidas/Entregables: Elección de los porcentajes para la división del conjunto de datos en entrenamiento/desarrollo/test (Train/Dev/Test).
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.14. Cuadro: División del conjunto de datos en Train/Dev/Test

3.2.2. Métodos de imputación de <i>missing values</i>
Paquete de trabajo: Diseño
Esfuerzo estimado: 10 horas
Periodo estimado: 5 días
Descripción: Seleccionar las variables que contienen algún valor nulo (<i>missing value</i>) en el conjunto de datos y diseñar los métodos de imputación de estos valores. En variables numéricas, la imputación se puede hacer con la media aritmética o la mediana de los valores existentes, o creando modelos de regresión para predecir el posible valor.
Salidas/Entregables: Diseño de los métodos de imputación de <i>missing values</i> para las variables con algún valor nulo.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.15. Cuadro: Métodos de imputación de *missing values*

3.2.3. Métodos de discretización de algunas variables numéricas
Paquete de trabajo: Diseño
Esfuerzo estimado: 2 horas
Periodo estimado: 1 días
Descripción: Seleccionar las variables numéricas que conviene discretizar (convertir en variables categóricas) y diseñar los métodos de discretización más adecuados. Un ejemplo es la discretización del IMC (índice de masa corporal) según los puntos de corte con los que se clasifica la obesidad.
Salidas/Entregables: Diseño de los métodos de discretización para las variables numéricas que conviene convertir en categóricas.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.16. Cuadro: Métodos de discretización de algunas variables numéricas

3.2.4. Estandarización/Normalización de variables numéricas
Paquete de trabajo: Diseño
Esfuerzo estimado: 10 horas
Periodo estimado: 5 días
Descripción: Seleccionar y diseñar la implementación de los métodos para estandarizar o normalizar los valores de las variables numéricas. Es importante este proceso ya que la escala de cada variable numérica puede variar mucho dependiendo de la variable, y la estandarización o normalización se encarga de ajustar estos valores en la misma escala. De esta manera, el algoritmo de ML prodrá establecer correlaciones entre las variables más fácilmente.
Salidas/Entregables: Selección de los métodos de estandarización o normalización para las variables numéricas y diseño de su implementación.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.17. Cuadro: Estandarización/Normalización de variables numéricas

3.3. Métodos de selección de variables y reducción de dimensiones
Paquete de trabajo: Diseño
Esfuerzo estimado: 10 horas
Periodo estimado: 9 días
Descripción: Escoger y diseñar la implementación de los métodos para seleccionar las variables que más aportan al modelo, con el objetivo de reducir la complejidad y mejorar los resultados. La selección de variables (<i>feature selection</i>) se puede hacer mediante filtrados (<i>filters</i>) basados en la ganancia de información (<i>Gini</i>) que proporciona cada variable o en las medidas de correlación (χ^2 , Pearson...) entre las variables. También se pueden aplicar otras técnicas (<i>wrappers</i>) que dependen del modelo de clasificación ya que utilizan su desempeño como medida para determinar la relevancia de los rasgos. Por otro lado, en vez de seleccionar un conjunto de variables, se pueden reducir las dimensiones del vector de rasgos mediante transformaciones lineales con métodos como PCA.
Salidas/Entregables: Elección de los métodos para seleccionar las variables más importantes o reducir la dimensión del vector de rasgos, y diseño de su implementación.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.18. Cuadro: Métodos de selección de variables y reducción de dimensiones

3.4. Métodos de importancia de variables y <i>heatmaps</i> de correlación
Paquete de trabajo: Diseño
Esfuerzo estimado: 6 horas
Periodo estimado: 9 días
Descripción: Seleccionar y diseñar la implementación de los métodos que muestren la importancia de cada variable a la hora de hacer las predicciones en un modelo entrenado. De esta manera, se podrá observar el valor de qué variable tiene más influencia en las predicciones. También son de gran ayuda los (<i>heatmaps</i>) para ver la correlación entre pares de variables.
Salidas/Entregables: Selección de los métodos para mostrar la importancia de las variables en las predicciones y las correlaciones entre pares de variables, y diseño de su implementación.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.19. Cuadro: Métodos de importancia de variables y *heatmaps* de correlación

3.5.1. Algoritmos
Paquete de trabajo: Diseño
Esfuerzo estimado: 12 horas
Periodo estimado: 9 días
Descripción: Seleccionar qué algoritmos de clasificación supervisada son los más adecuados para nuestra tarea (clasificación binaria) y nuestro conjunto de datos: <i>logistic regression, random forest, SVM, MLP...</i> Asimismo, diseñar la implementación de estos algoritmos en nuestro entorno de desarrollo.
Salidas/Entregables: Algoritmos de clasificación supervisada seleccionados y diseño de su implementación.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.20. Cuadro: Algoritmos

3.5.2. Hiperparámetros a optimizar
Paquete de trabajo: Diseño
Esfuerzo estimado: 3 horas
Periodo estimado: 9 días
Descripción: Algunos de los algoritmos de ML seleccionados en la tarea anterior tienen hiperparámetros ajustables, por lo que se deben seleccionar aquellos que nos convenga ajustar para optimizar los resultados.
Salidas/Entregables: Hiperparámetros a ajustar seleccionados dependiendo del algoritmo de ML.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.21. Cuadro: Hiperparámetros a optimizar

3.6.1. Esquema de evaluación
Paquete de trabajo: Diseño
Esfuerzo estimado: 2 horas
Periodo estimado: 2 días
Descripción: Seleccionar y diseñar la implementación del esquema de evaluación más adecuado para obtener los resultados sobre el modelo predictivo entrenado. Los métodos de evaluación más habituales son Cross Validation y Hold-out.
Salidas/Entregables: Diseño del esquema de evaluación seleccionado.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

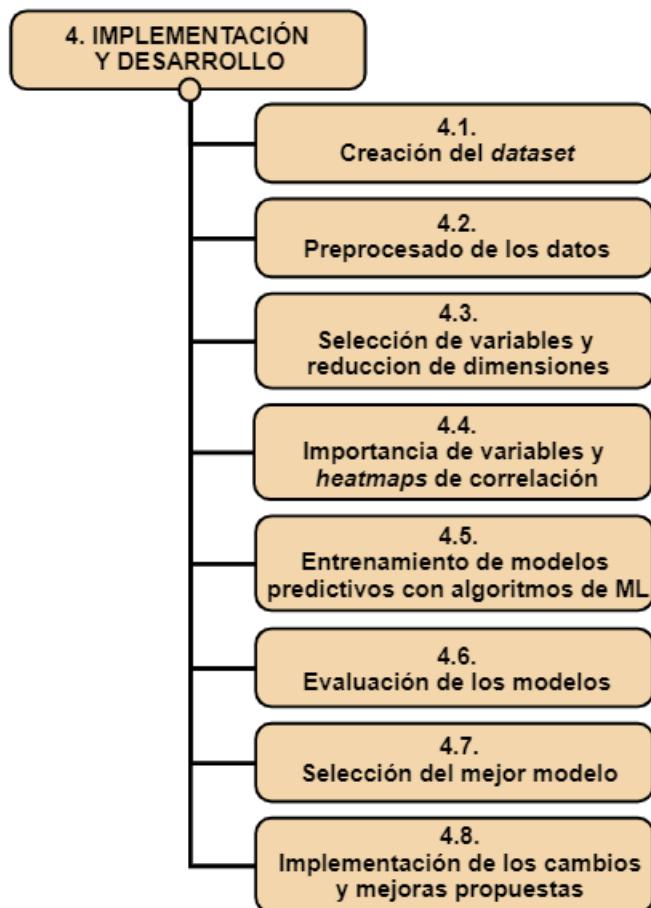
3.22. Cuadro: Esquema de evaluación

3.6.2. Métricas de evaluación
Paquete de trabajo: Diseño
Esfuerzo estimado: 3 horas
Periodo estimado: 2 días
Descripción: Seleccionar las figuras de mérito a tener en cuenta para evaluar el modelo predictivo entrenado. Las métricas más habituales son <i>accuracy</i> , <i>recall</i> , <i>precision</i> , <i>F-score</i> ... Otras como AUC (área bajo la curva) pueden ser de gran utilidad al no variar dependiendo de la distribución de la clase.
Salidas/Entregables: Figuras de mérito seleccionadas para evaluar el modelo.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.23. Cuadro: Métricas de evaluación

3.3.4. Implementación y desarrollo

A este apartado pertenecen las tareas para llevar a cabo la implementación del diseño del bloque anterior. Cada una de las tareas está relacionada con una tarea del bloque de diseño (ver Figura 3.7).



3.7. Figura: Diagrama EDT del bloque de implementación y desarrollo

4.1. Creación del dataset
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 25 horas
Periodo estimado: 16 días
Descripción: Crear el conjunto de datos uniendo en cada ejemplo los valores de las características del paciente y el valor de la clase, como se ha diseñado en la tarea correspondiente (ver Cuadro 3.13).
Salidas/Entregables: Conjunto de datos <i>dataset</i> completo.
Recursos necesarios: PyCharm, Python(Pandas) y datos de entrada en documentos Excel.

3.24. Cuadro: Creación del dataset

4.2. Preprocesado de los datos
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 30 horas
Periodo estimado: 21 días
Descripción: Implementar las técnicas de preprocesado seleccionadas y diseñadas: división del conjunto de datos en entrenamiento/desarrollo/test (Train/Dev/Test), imputación de <i>missing values</i> , discretización de variables numéricas y estandarización/normalización de las variables numéricas. Se seguirán los diseños creados en las tareas de los Cuadros 3.14, 3.15, 3.16 y 3.17. La implementación se aplicará al conjunto de datos obtenido tras la tarea anterior.
Salidas/Entregables: Conjuntos de datos divididos en Train/Dev/Test y con los diferentes métodos de imputación de <i>missing values</i> y estandarización/nomalización aplicados.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn) y <i>dataset</i> de entrada.

3.25. Cuadro: Preprocesado de los datos

4.3. Selección de variables y reducción de dimensiones
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 10 horas
Periodo estimado: 7 días
Descripción: Implementar los métodos de selección de variables y reducción de dimensiones seleccionados y diseñados tras realizar la tarea del Cuadro 3.18. La implementación se aplicará a los conjuntos de datos obtenidos tras la tarea anterior.
Salidas/Entregables: Conjuntos de datos con las variables más importantes seleccionadas o con dimensiones reducidas, y diagramas con información acerca de estos procesos.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn) y <i>datasets</i> preprocesados.

3.26. Cuadro: Selección de variables y reducción de dimensiones

4.4. Importancia de variables y <i>heatmaps</i> de correlación
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 10 horas
Periodo estimado: 7 días
Descripción: Crear los <i>heatmaps</i> de correlación e implementar los métodos de importancia de variables seleccionados y diseñados tras realizar la tarea del Cuadro 3.19. La implementación se aplicará a los conjuntos de datos obtenidos tras el preprocesado de datos (tarea 3.2.).
Salidas/Entregables: Diagramas e información acerca de la importancia de variables y matrices de correlación entre las variables.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn) y <i>datasets</i> preprocesados.

3.27. Cuadro: Importancia de variables y *heatmaps* de correlación

4.5. Entrenamiento de modelos predictivos con algoritmos de ML
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 50 horas
Periodo estimado: 43 días
Descripción: Implementar los algoritmos de ML supervisado seleccionados tras realizar la tarea del Cuadro 3.20 (y optimizando sus hiperparámetros seleccionados tras realizar la tarea del Cuadro 3.21, dependiendo del caso). El entrenamiento de los modelos predictivos se hará sobre los conjuntos de datos obtenidos tras el preprocesado de datos (tarea 3.2.) y/o selección de variables/reducción de dimensiones (tarea 3.3.).
Salidas/Entregables: Modelos predictivos.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn) y <i>datasets</i> preprocesados y/o tras selección de variables/reducción de dimensiones.

3.28. Cuadro: Entrenamiento de modelos predictivos con algoritmos de ML

4.6. Evaluación de los modelos
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 10 horas
Periodo estimado: 43 días
Descripción: Implementar el esquema de evaluación seleccionado y diseñado tras realizar la tarea del Cuadro 3.22. La evaluación se hará mediante los resultados de las métricas seleccionadas tras realizar la tarea del Cuadro 3.23. Las evaluaciones se harán sobre los modelos predictivos entrenados en la tarea anterior.
Salidas/Entregables: Resultados de las evaluaciones de los modelos predictivos entrenados.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn), modelos predictivos entrenados y <i>datasets</i> de entrenamiento, desarrollo y test (Train/Dev/Test).

3.29. Cuadro: Evaluación de los modelos

4.7. Selección del mejor modelo
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 5 horas
Periodo estimado: 7 días
Descripción: Seleccionar el modelo predictivo con mejores resultados en las evaluaciones realizadas. Para ello, hay que decidir el criterio de selección, en este caso, a qué métricas se les va a dar más importancia.
Salidas/Entregables: Modelo predictivo con mejores resultados tras los criterios de evaluación establecidos.
Recursos necesarios: Diagramas y resultados de las evaluaciones de los modelos.

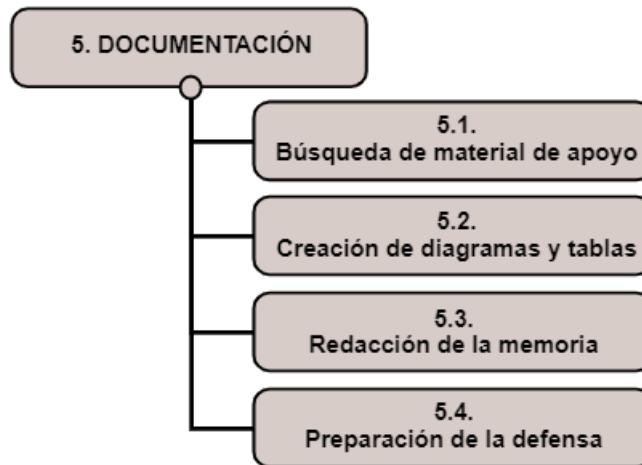
3.30. Cuadro: Selección del mejor modelo

4.8. Implementación de los cambios y mejoras propuestas
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 30 horas
Periodo estimado: 14 días
Descripción: Implementar los cambios a realizar en el software desarrollado tras el análisis realizado en la tarea del Cuadro 3.12.
Salidas/Entregables: Software con los cambios y mejoras aplicadas.
Recursos necesarios: Depende del cambio o mejora.

3.31. Cuadro: Implementación de los cambios y mejoras propuestas

3.3.5. Documentación

En este último apartado se detallan las tareas relacionadas con la documentación del trabajo realizado (ver Figura 3.8).



3.8. Figura: Diagrama EDT del bloque de documentación

5.1. Búsqueda de material de apoyo	
Paquete de trabajo:	Documentación
Esfuerzo estimado:	15 horas
Periodo estimado:	290 días (tarea <i>hammock</i>)
Descripción:	Buscar artículos, libros o documentos que tengan relación con cualquier tema relacionado con el proyecto.
Salidas/Entregables:	Bibliografía.
Recursos necesarios:	Navegador.

3.32. Cuadro: Búsqueda de material de apoyo

5.2. Creación de diagramas y tablas	
Paquete de trabajo:	Documentación
Esfuerzo estimado:	17 horas
Periodo estimado:	89 días
Descripción:	Crear tablas y diagramas en las tareas de los bloques de análisis y diseño que se usarán en la memoria del proyecto: diagrama de la arquitectura del software, diagrama EDT, tablas de las tareas del EDT, diagrama del diseño del software, diagrama de Gantt...
Salidas/Entregables:	Tablas y diagramas.
Recursos necesarios:	Cacoo, Overleaf, LaTeX y GanttProject.

3.33. Cuadro: Creación de diagramas y tablas

5.3. Redacción de la memoria	
Paquete de trabajo:	Documentación
Esfuerzo estimado:	150 horas
Periodo estimado:	89 días
Descripción:	Redactar la memoria con la información acerca de todo el desarrollo del proyecto.
Salidas/Entregables:	Memoria del proyecto.
Recursos necesarios:	Overleaf y LaTeX.

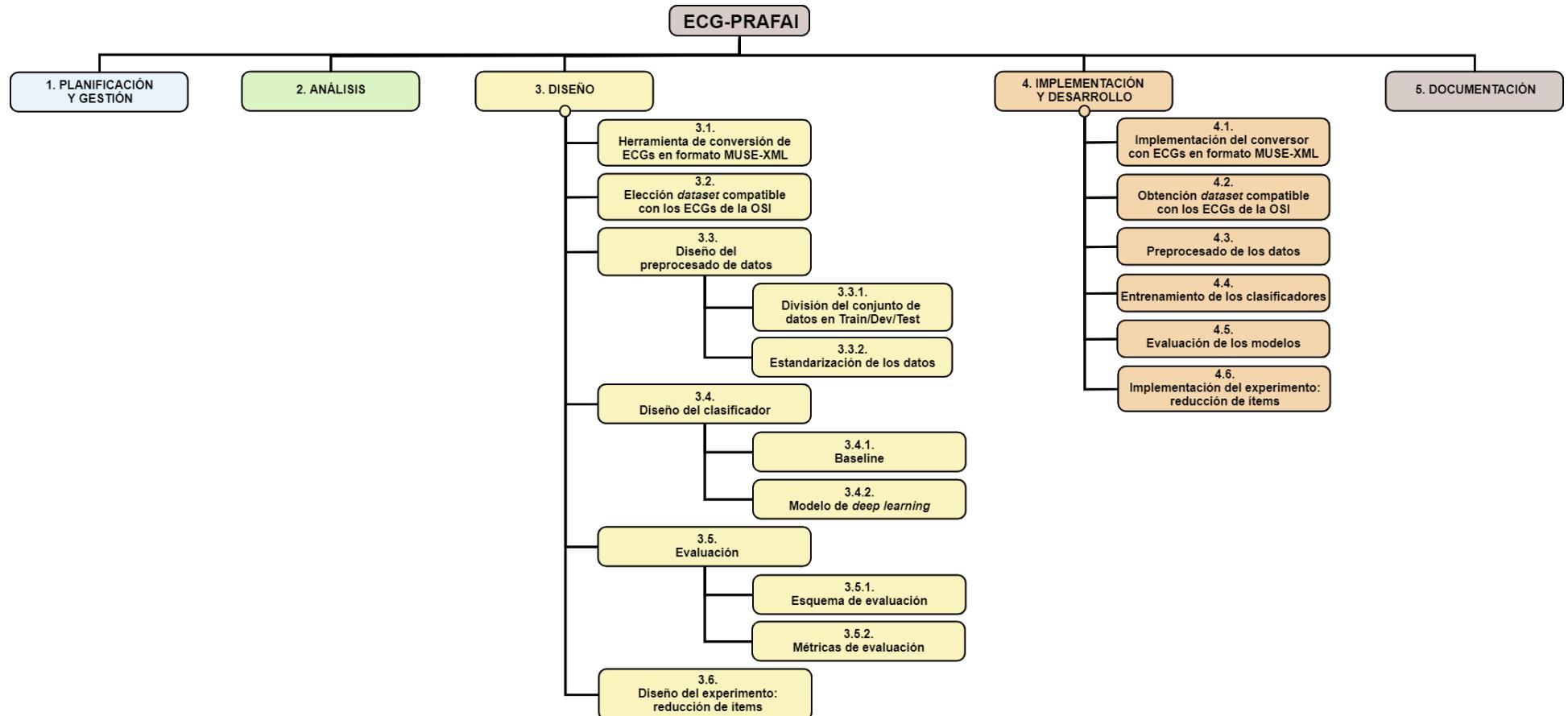
3.34. Cuadro: Redacción de la memoria

5.4. Preparación de la defensa
Paquete de trabajo: Documentación
Esfuerzo estimado: 30 horas
Periodo estimado: 14 días
Descripción: Elaborar y preparar la presentación de la defensa del proyecto.
Salidas/Entregables: Presentación en formato PowerPoint o PDF.
Recursos necesarios: Ordenador, proyecto y documentación.

3.35. Cuadro: Preparación de la defensa

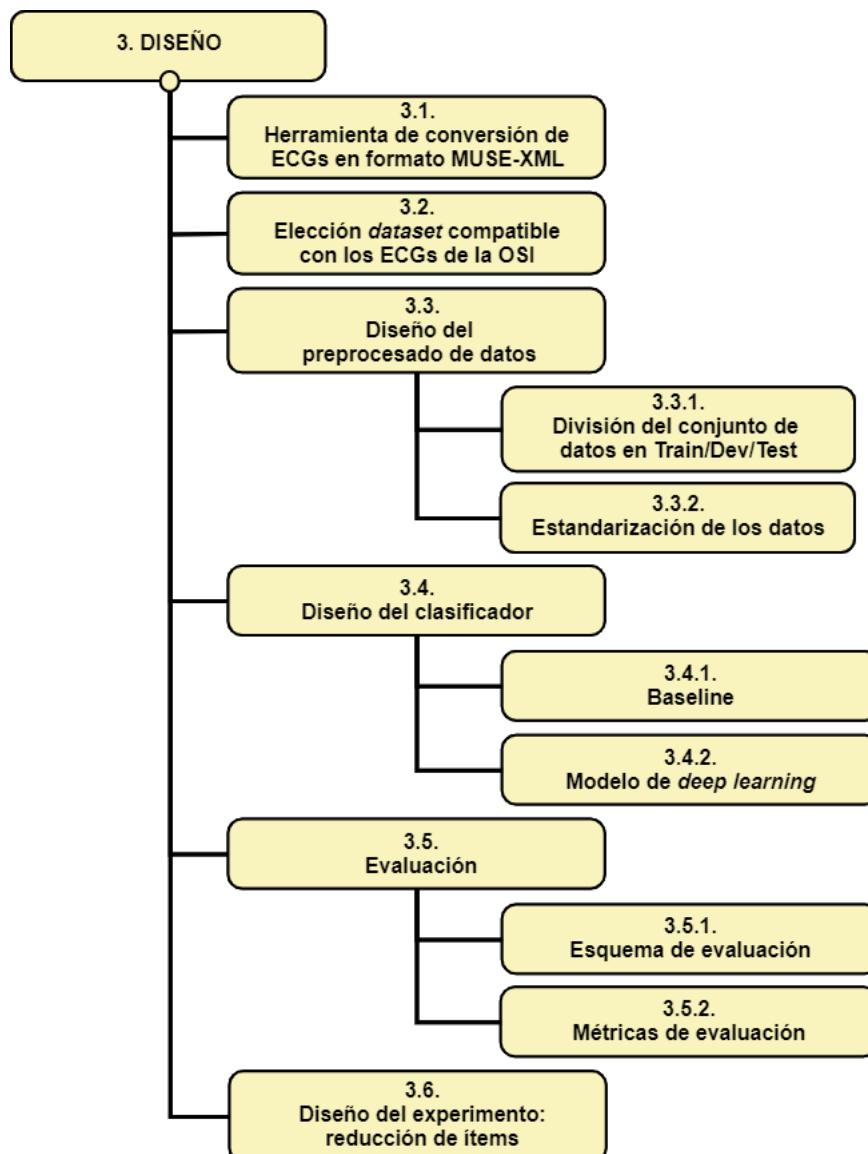
3.3.6. Alcance de la tarea secundaria: ECG-PRAFAI

En esta sección se define el alcance de la tarea secundaria: crear un modelo predictivo de recurrencia de fibrilación auricular utilizando la información extraída de los propios electrocardiogramas. Para ello, se ha realizado un nuevo diagrama EDT (Estructura de Descomposición de Trabajo), el cual está formado por los mismo cinco bloques principales. Los bloques *Planificación y gestión*, *Análisis* y *Documentación* tienen las mismas tareas que el EDT de la tarea principal (ver Figura 3.3). Sin embargo, las tareas cambian en los bloques *Diseño* e *Implementación y desarrollo*, ya que se trata de una tarea diferente. A continuación se muestra el diagrama EDT de la tarea secundaria ECG-PRAFAI (ver Figura 3.9).



3.9. Figura: Diagrama EDT de la tarea secundaria: ECG-PRAFAI

3.3.6.1. Diseño



3.10. Figura: Diagrama EDT del bloque de diseño (ECG-PRAFAI)

3.1. Herramienta de conversión de ECGs en formato MUSE-XML
Paquete de trabajo: Diseño
Esfuerzo estimado: 6 horas
Periodo estimado: 5 días
Descripción: Búsqueda de una herramienta de conversión de ECGs de 12 derivaciones de formato MUSE-XML a formato CSV.
Salidas/Entregables: Software para la conversión de ECGs en formato MUSE-XML.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.36. Cuadro: Herramienta de conversión de ECGs en formato MUSE-XML

3.2. Elección dataset compatible con los ECGs de la OSI
Paquete de trabajo: Diseño
Esfuerzo estimado: 12 horas
Periodo estimado: 7 días
Descripción: Búsqueda de un conjunto de datos formado por ECGs de 12 derivaciones extraídos de sistemas MUSE. Los ECGs deberían pertenecer a pacientes tras un primer evento de fibrilación auricular, de esta manera podría crearse un modelo de recurrencia de FA extrapolable a los ECGs de la OSI.
Salidas/Entregables: Conjunto de datos de ECGs de 12 derivaciones extraídos de sistemas MUSE.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.37. Cuadro: Elección del *dataset* compatible con los ECGs de la OSI

3.3.1. División del conjunto de datos en Train/Dev/Test
Paquete de trabajo: Diseño
Esfuerzo estimado: 1 hora
Periodo estimado: 1 día
Descripción: Decidir los porcentajes para la división del conjunto de datos en tres subconjuntos: Train (ejemplos para el entrenamiento del modelo), Dev (ejemplos para la evaluación del modelo entrenado) y Test (ejemplos nunca vistos durante el entrenamiento para testear el modelo final).
Salidas/Entregables: Elección de los porcentajes para la división del conjunto de datos en entrenamiento/desarrollo/test (Train/Dev/Test).
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.38. Cuadro: División del conjunto de datos en Train/Dev/Test

3.3.2. Estandarización de los datos
Paquete de trabajo: Diseño
Esfuerzo estimado: 1 hora
Periodo estimado: 1 día
Descripción: Seleccionar y diseñar la implementación de un método para estandarizar los valores de entrada al algoritmo de aprendizaje automático. De esta manera, se escalan los datos y el algoritmo podrá establecer correlaciones entre las variables más fácilmente.
Salidas/Entregables: Elección del método de estandarización de los datos y diseño de su implementación.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.39. Cuadro: Estandarización de los datos

3.4.1. Baseline
Paquete de trabajo: Diseño
Esfuerzo estimado: 2 horas
Periodo estimado: 1 día
Descripción: Diseño de un clasificador <i>baseline</i> para la creación de un primer modelo predictivo.
Salidas/Entregables: Diseño del modelo <i>baseline</i> .
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.40. Cuadro: Baseline

3.4.2. Modelo de Deep Learning
Paquete de trabajo: Diseño
Esfuerzo estimado: 5 horas
Periodo estimado: 3 días
Descripción: Diseño de un clasificador basado en algoritmos de Deep Learning como las redes neuronales para la creación de un modelo predictivo más complejo.
Salidas/Entregables: Diseño del modelo de Deep Learning.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.41. Cuadro: Modelo de *deep learning*

3.5.1. Esquema de evaluación
Paquete de trabajo: Diseño
Esfuerzo estimado: 1 hora
Periodo estimado: 1 día
Descripción: Seleccionar y diseñar la implementación del esquema de evaluación más adecuado para obtener los resultados sobre el modelo predictivo entrenado. Los métodos de evaluación más habituales son Cross Validation y Hold-out.
Salidas/Entregables: Diseño del esquema de evaluación seleccionado.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.42. Cuadro: Esquema de evaluación

3.5.2. Métricas de evaluación
Paquete de trabajo: Diseño
Esfuerzo estimado: 1 hora
Periodo estimado: 1 día
Descripción: Seleccionar las figuras de mérito a tener en cuenta para evaluar el modelo predictivo entrenado. Las métricas más habituales son <i>accuracy</i> , <i>reacall</i> , <i>precision</i> , <i>F-score</i> ... Otras como AUC (área bajo la curva) pueden ser de gran utilidad al no variar dependiendo de la distribución de la clase.
Salidas/Entregables: Figuras de mérito seleccionadas para evaluar el modelo.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.43. Cuadro: Métricas de evaluación

3.6. Diseño del experimento: reducción de ítems
Paquete de trabajo: Diseño
Esfuerzo estimado: 4 horas
Periodo estimado: 3 días
Descripción: Diseño para el experimento que va a consistir en ir entrenando modelos disminuyendo el número de ítems de entrenamiento, hasta observar un empeoramiento significativo en los resultados. De esta manera, se puede obtener una aproximación de los ECGs necesarios para crear un modelo predictivo de recurrencia de FA robusto en base a ECGs de 12 derivaciones.
Salidas/Entregables: Diseño del método seleccionado para ir reduciendo los ítems del conjunto de datos e ir entrenando modelos de forma iterativa.
Recursos necesarios: Navegador, Cacoo y documentos de texto de Microsoft Office 365.

3.44. Cuadro: Diseño del experimento: reducción de ítems

3.3.6.2. Implementación y desarrollo



3.11. Figura: Diagrama EDT del bloque de implementación y desarrollo (ECG-PRAFAI)

4.1. Implementación del conversor con ECGs en formato MUSE-XML	
Paquete de trabajo:	Implementación y desarrollo
Esfuerzo estimado:	2 horas
Periodo estimado:	2 días
Descripción:	Utilizar la herramienta conversora de ECGs para transformarlos de formato MUSE-XML a formato CSV, como se ha diseñado en la tarea correspondiente (ver Cuadro 3.36).
Salidas/Entregables:	ECGs en formato CSV.
Recursos necesarios:	PyCharm, Python(Pandas) y datos de entrada en formato MUSE-XML.

3.45. Cuadro: Implementación del conversor con ECGs en formato MUSE-XML

4.2. Obtención dataset compatible con los ECGs de la OSI	
Paquete de trabajo:	Implementación y desarrollo
Esfuerzo estimado:	1 hora
Periodo estimado:	1 día
Descripción:	Descargar un conjunto de datos formado por ECGs de 12 derivaciones extraídos de sistemas MUSE para la creación del modelo predictivo, como se ha diseñado en la tarea correspondiente (ver Cuadro 3.37).
Salidas/Entregables:	Conjunto de datos de ECGs de 12 derivaciones.
Recursos necesarios:	Navegador y PyCharm.

3.46. Cuadro: Obtención dataset compatible con los ECGs de la OSI

4.3. Preprocesado de los datos
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 1 hora
Periodo estimado: 1 día
Descripción: Implementar las técnicas de preprocesado seleccionadas y diseñadas: división del conjunto de datos en entrenamiento/desarrollo/test (Train/Dev/Test) y estandarización de los datos. Se seguirá el diseño creado en la tareas de los Cuadros 3.38 y 3.39. La implementación se aplicará al conjunto de datos obtenido tras la tarea anterior.
Salidas/Entregables: Conjuntos de datos divididos y estandarizados en Train/-Dev/Test.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn) y <i>dataset</i> de entrada.

3.47. Cuadro: Preprocesado de los datos

4.4. Entrenamiento de los clasificadores
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 8 horas
Periodo estimado: 6 días
Descripción: Implementar los algoritmos de clasificación seleccionados, tanto el <i>baseline</i> como el modelo de <i>deep learning</i> , como se ha diseñado en las tareas correspondientes (ver Cuadros 3.40 y 3.41).
Salidas/Entregables: Modelos predictivos.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn) y <i>dataset</i> .

3.48. Cuadro: Entrenamiento de los clasificadores

4.5. Evaluación de los modelos
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 2 horas
Periodo estimado: 1 días
Descripción: Implementar el esquema de evaluación seleccionado y diseñado tras realizar la tarea del Cuadro 3.42. La evaluación se hará mediante los resultados de las métricas seleccionadas tras realizar la tarea del Cuadro 3.43. Las evaluaciones se harán sobre los modelos predictivos entrenados en la tarea anterior.
Salidas/Entregables: Resultados de las evaluaciones de los modelos predictivos entrenados.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn), modelos predictivos entrenados y <i>datasets</i> de entrenamiento, desarrollo y test (Train/Dev/Test).

3.49. Cuadro: Evaluación de los modelos

4.6. Implementación del experimento: reducción de ítems
Paquete de trabajo: Implementación y desarrollo
Esfuerzo estimado: 6 horas
Periodo estimado: 3 días
Descripción: Implementar el experimento que consiste en la reducción iterativa de ítems del conjunto de datos y el entrenamiento de diferentes modelos hasta observar un decremento significativo en los resultados de la evaluación, como se ha diseñado en la tarea correspondiente (ver Cuadros 3.44).
Salidas/Entregables: Resultados de las evaluaciones tras reducir el número de datos de entrenamiento de forma iterativa.
Recursos necesarios: PyCharm, Python(Pandas y Scikit-learn), modelos predictivos entrenados y <i>datasets</i> de entrenamiento, desarrollo y test (Train/Dev/Test).

3.50. Cuadro: Implementación del experimento: reducción de ítems

3.4. Cronograma / Diagrama de Gantt

En esta sección se da a conocer la planificación temporal del proyecto. Una vez especificadas las tareas a realizar en el diagrama EDT, se ha planeado su duración y cuándo se deben de realizar durante el transcurso de desarrollo del proyecto.

En las siguientes tablas se muestra el esfuerzo (en horas) y la duración/periodo (en días) estimado para el desarrollo del proyecto, desglosado por fases y tareas. El esfuerzo es la cantidad de horas totales invertidas en la realización de una tarea, mientras que la duración o el periodo representa la cantidad de días que han transcurrido desde el comienzo hasta la finalización de la tarea. Como se puede observar, la duración total planificada ha sido de 304 días, alrededor de 10 meses. Por otro lado, el esfuerzo total ha sido de 620 horas, es decir, en total se ha planificado trabajar durante 620 horas en el proyecto.

Fase/Tarea	Esfuerzo (horas)	Periodo (días)
1. Planificación y gestión	48	304 *
1.1. Reuniones	28	304 *
1.1.1. Reuniones con el cliente/ data-manager	10	304 *
1.1.2. Reuniones con el grupo de trabajo	18	304 *
1.2. Objetivos y alcance	5	3
1.3. Beneficios que aporta el proyecto	5	3
1.4. Cronograma / Diagrama de Gantt	10	3
2. Análisis	67	213
2.1. Captura de requisitos	5	3
2.2. Marco teórico	10	5
2.3. Estado del arte	20	14
2.4. Análisis de los datos de entrada	12	10
2.5. Selección del framework para el desarrollo del software	2	1
2.6. Selección de recursos	3	1
2.7. Análisis de los cambios y mejoras propuestas	15	12
3. Diseño	70	100
3.1. Diseño del dataset	10	5
3.2. Diseño del preprocesado de datos	24	29
3.2.1. División del conjunto de datos en Train/Dev/Test	2	1
3.2.2. Métodos de imputación de missing values	10	5
3.2.3. Métodos de discretización de algunas variables numéricas	2	1
3.2.4. Estandarización/Normalización de variables numéricas	10	5
3.3. Métodos de selección de variables y reducción de dimensiones	10	9
3.4. Métodos de importancia de variables y <i>heatmaps</i> de correlación	6	9
3.5. Algoritmos de ML supervisado	15	9
3.5.1. Algoritmos	12	9
3.5.2. Hiperparámetros a optimizar	3	9
3.6. Evaluación	5	4
3.6.1. Esquema de evaluación	2	2
3.6.2. Métricas de evaluación	3	2

3.51. Cuadro: Planificación del tiempo (I)

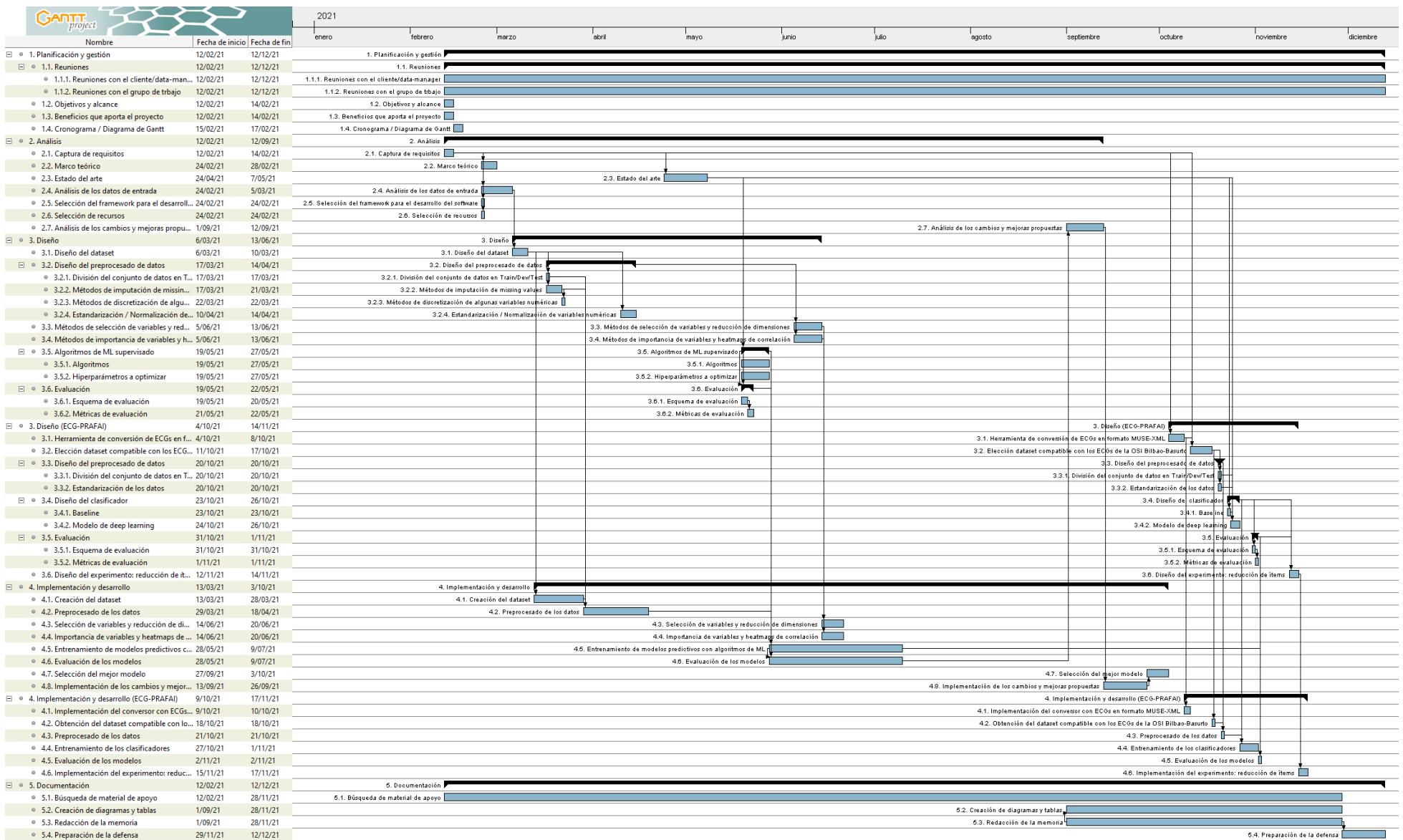
Fase/Tarea	Esfuerzo (horas)	Duración (días)
3. Diseño (ECG-PRAFAI)	33	42
3.1. Herramienta de conversión de ECGs en formato MUSE-XML	6	5
3.2. Elección <i>dataset</i> compatible con los ECGs de la OSI	12	7
3.3. Diseño del preprocesado de datos	2	1
3.3.1. División del conjunto de datos en Train/Dev/Test	1	1
3.3.2. Estandarización de los datos	1	1
3.4. Diseño del clasificador	7	4
3.4.1. Baseline	2	1
3.4.2. Modelo de Deep Learning	5	3
3.5. Evaluación	2	2
3.5.1. Esquema de evaluación	1	1
3.5.2. Métricas de evaluación	1	1
3.6. Diseño del experimento: reducción de ítems	4	3
4. Implementación y desarrollo	170	205
4.1. Creación del dataset	25	16
4.2. Preprocesado de los datos	30	21
4.3. Selección de variables y reducción de dimensiones	10	7
4.4. Importancia de variables y heatmaps de correlación	10	7
4.5. Entrenamiento de modelos predictivos con algoritmos de ML	50	43
4.6. Evaluación de los modelos	10	43
4.7. Selección del mejor modelo	5	7
4.8. Implementación de los cambios y mejoras propuestas	30	14
4. Implementación y desarrollo (ECG-PRAFAI)	20	40
4.1. Implementación del conversor con ECGs en formato MUSE-XML	2	2
4.2. Obtención <i>dataset</i> compatible con los ECGs de la OSI	1	1
4.3. Preprocesado de los datos	1	1
4.4. Entrenamiento de los clasificadores	8	6
4.5. Evaluación de los modelos	2	1
4.6. Implementación del experimento: reducción de ítems	6	3
5. Documentación	212	304 *
5.1. Búsqueda de material de apoyo	15	290 *
5.2. Creación de diagramas y tablas	17	89
5.3. Redacción de la memoria	150	89
5.4. Preparación de la defensa	30	14
TOTAL	620	304

3.52. Cuadro: Planificación del tiempo (II)

*: Hay algunas tareas, como las reuniones o tareas de documentación, que se van a realizar a lo largo de todo el desarrollo del proyecto. Estas tareas están marcadas con un * en la tabla anterior y se denominan tareas *hammock*.

Por otro lado, utilizando la herramienta *GanttProject*, se ha creado el diagrama de Gantt o cronograma (ver Figura 3.12). Se han incluido todas las tareas de la tabla anterior, estableciendo en cada una de ellas las fechas de inicio y fin. De esta manera, se ha especificado en el diagrama cuándo se deben realizar y qué duración deben tener las tareas.

La fecha de comienzo del proyecto ha sido el **12 de febrero de 2021** y la fecha estimada de finalización es el **12 de diciembre de 2021**. Cabe destacar que durante el transcurso del proyecto ha habido un periodo de vacaciones del 9 de julio al 31 de agosto (ambos incluidos), esto supone 53 días en los que no se ha trabajado. A pesar de que en el diagrama de Gantt aparecen las tareas *hammock* en realización, durante ese periodo de tiempo no se ha realizado ninguna tarea. Por lo tanto, en vez de 304 días, **la duración real del proyecto sería de 251 días**, sin tener en cuenta el periodo de vacaciones.



3.12. Figura: Diagrama de Gantt

3.5. Evaluación económica

En este apartado se va a realizar un presupuesto de los costes de desarrollo de un proyecto de estas magnitudes. Como se trata de un trabajo fin de grado, no se espera obtener ningún beneficio económico, pero cada vez que se desarrolla un proyecto es importante estimar sus consecuencias económicas y el coste del desarrollo. De esta manera, en el caso de ser comercializado, se podrían conocer los beneficios futuros.

3.5.1. Costes por mano de obra

En cuanto a la mano de obra de los recursos humanos, hay que tener en cuenta el coste por hora de los miembros del equipo de trabajo. Según la información salarial que ofrece el *Boletón Oficial del Estado (BOE)* para el año 2021, el salario de un ingeniero informático que acaba de terminar su grado puede rondar los 18.000 y 22.000 euros brutos al año. Tomando 20.450€ como un valor medio entre esas dos cifras, ese sería el salario bruto al año del desarrollador del proyecto.

Para calcular el salario neto habría que descontar retenciones fiscales (retención por IRPF del 11,92 % y cotización del 6,35 % a la Seguridad Social), por lo que se estima que 16.713,8€ sería el salario neto anual de un ingeniero informático que acaba de terminar su grado, un sueldo neto mensual de 1.392,8€. Una vez calculado el mensual, el salario por hora de trabajo se puede obtener con la siguiente ecuación:

$$\text{Sueldo/Hora} = \frac{\text{Sueldo mensual}}{\text{Semanas/Mes} \cdot \text{Días trabajo/Semana} \cdot \text{Horas trabajo/Día}}$$

Teniendo en cuenta que un mes consta aproximadamente de 4 semanas y la jornada laboral se extiende de lunes a viernes con una duración de 8 horas diarias, se obtiene el siguiente sueldo por hora de trabajo:

$$\text{Sueldo/Hora} = \frac{1392,8}{4 \cdot 5 \cdot 8} = \frac{1392,8}{160} = 8,71\text{€/h}$$

La empresa tiene que añadir a los costes por hora de los recursos humanos los pagos a la Seguridad Social. Concretamente, estos son los porcentajes en algunos ámbitos de la Seguridad Social que se deben añadir al salario de los trabajadores: contingencias comunes 23,6 %, desempleo 5,5 %, formación profesional 0,6 %, FOGASA 0,2 %, y accidentes de trabajo y enfermedades profesionales 1,5 %. En total al coste por hora de cada trabajador se le debe sumar el 31,4 %, por lo que el coste por hora del desarrollador del proyecto en la empresa sería de **11,44€/h**.

Finalmente, una vez obtenidos el coste por hora de los trabajadores y el esfuerzo total del proyecto, se calcula el coste total por mano de obra de los recursos humanos. En este proyecto, sabiendo que el salario del desarrollador es de 11,44€/h y el esfuerzo total es de 620h, el coste por mano de obra asciende a **7.092,80€**.

3.5.2. Costes en hardware

En cuanto al hardware necesario en el desarrollo del proyecto, se ha utilizado un ordenador portátil valorado en 600€, con una esperanza de vida de 4 años y un 40 % de uso durante el proyecto (la duración del proyecto ha sido de 304 días, 251 días sin tener en cuenta el periodo de vacaciones).

La amortización anual del ordenador es de $600\text{€}/4\text{años} = 150\text{€}/\text{año}$. Por lo que el coste del ordenador en el proyecto es de $150\text{€}/\text{año} * (251\text{ días}/365\text{ días al año}) * 40 \% = 41,26\text{€}$.

3.5.3. Costes en software

Por otro lado, respecto al software necesario en el desarrollo del proyecto, se ha utilizado el *IDE PyCharm Community Edition* con licencia *Apache 2*, es decir, gratuito y de código abierto. El resto de software utilizado en el proyecto tampoco ha supuesto ningún coste adicional, o bien es software gratuito (*GanttProject*, *Cacoo...*), o bien viene incluido en el precio del ordenador portátil (*Microsoft Office 365*).

Por lo tanto, no hay que contabilizar ningún coste en cuanto al software utilizado durante el desarrollo del proyecto.

3.5.4. Costes indirectos

Para terminar con los gastos, hay que calcular los costes indirectos. Los recursos que generan estos gastos no influyen directamente en el desarrollo de nuestro proyecto, pero son necesarios para utilizar los recursos hardware y software, por ejemplo, la electricidad y la conexión a internet.

Un método sencillo para la estimación de los costes indirectos es calcular el 2 % de la suma del resto de gastos (mano de obra, hardware y software). Por lo tanto, sabiendo que los costes por mano de obra han sido de 7.092,80€, los costes de hardware han sido de 41,26€, y no ha habido costes de software, los gastos indirectos han sido de $(7092,80 + 41,26) * 2\% = 142,68\text{€}$.

3.5.5. Costes totales

En este último apartado de la evaluación económica, se ha calculado el coste total que supone la realización del proyecto (ver Cuadro 3.53).

Tipo de coste	Coste generado
Mano de obra	7.092,80€
Costes en <i>hardware</i>	41,26€
Costes en <i>software</i>	0€
Costes indirectos	142,68€
TOTAL	7.277,08€

3.53. Cuadro: Costes totales del proyecto

Como se muestra en el Cuadro 3.53, la suma del conjunto de gastos del proyecto asciende a **7.277,08€**.

3.6. Análisis de riesgos

En este apartado se va a realizar el análisis y gestión de riesgos del proyecto. La gestión de riesgos identifica los imprevistos que pueden surgir durante el desarrollo del proyecto, y planifica estrategias para reducir el impacto que pueden llegar a tener en el proyecto. Para ello, por cada riesgo identificado, en las siguientes tablas se especifica su descripción, plan de prevención, plan de contingencia, probabilidad de que suceda e impacto en el proyecto.

La probabilidad e impacto de los riesgos se ha dividido en 5 categorías: muy bajo, bajo, medio, alto y muy alto. En el caso del impacto, se ha decidido que la medida de impacto dependerá de la pérdida de tiempo de trabajo, de manera que: *muy bajo* y *bajo* para problemas que se pueden solucionar en cuestión de una hora o menos, *medio* para problemas que requieren más de una hora de atención en una misma jornada, y *alto* y *muy alto* si la solución requiere más de una jornada laboral.

Enfermedad o lesión del desarrollador	
Descripción	Imposibilidad temporal para trabajar del desarrollador del proyecto, ya sea por una lesión o enfermedad por causas derivadas o ajenas al trabajo.
Plan de prevención	Para prevenir enfermedades o lesiones laborales, mantener una postura corporal correcta frente al ordenador y hacer descansos cortos periódicamente. En cuanto a prevenciones ajenas al trabajo mantener hábitos de vida saludables.
Plan de contingencia	Acudir al médico para diagnosticar el alcance de la enfermedad o lesión. En el caso de ser una enfermedad leve, continuar con el trabajo en la medida de lo posible y con precaución, y en el caso de mayor gravedad, retrasar las tareas en el periodo de tiempo de indisponibilidad hasta su recuperación.
Probabilidad	Baja
Impacto	De medio a muy alto (dependiendo del diagnóstico)

3.54. Cuadro: Riesgo 1

Problemas con el equipo de trabajo y recursos	
Descripción	Problemas surgidos en el hardware y software necesario para el desarrollo del proyecto, así como pérdidas de información: ordenador portátil estropeado, problemas para el uso de las herramientas software, pérdida de datos, programas desarrollados o documentación...
Plan de prevención	Para prevenir problemas con el hardware, mantener el equipo en condiciones adecuadas y realizar revisiones periódicamente. Para prevenir pérdidas de información, hacer copias de seguridad de forma periódica y almacenarlas en algún disco duro externo o servidor web.
Plan de contingencia	En cuanto a problemas con el hardware, como un ordenador estropeado, se reemplazará por otro funcional y se adecuará para su uso en el proyecto. En el caso de pérdidas de información, se restaurará la última copia de seguridad guardada.
Probabilidad	Problemas en hardware: Muy baja Problemas en software: Baja
Impacto	Problemas de hardware: Muy alto Problemas en software: Medio

3.55. Cuadro: Riesgo 2

Falta de experiencia y conocimientos en trabajos de este ámbito	
Descripción	Aunque durante el grado se han realizado tareas de Inteligencia Artificial y Machine Learning, es la primera vez que se trabaja con herramientas de ML en Python(librería Scikit-learn), al igual que con otras herramientas de estructuras de datos de Python (por ejemplo, librería Pandas).
Plan de prevención	Dedicar el tiempo necesario al aprendizaje de las nuevas herramientas e informarse en fuentes fiables.
Plan de contingencia	Aplicar los conocimientos obtenidos en el aprendizaje y búsqueda de información para solventar los problemas surgidos por falta de conocimiento o experiencia.
Probabilidad	Alta
Impacto	Medio

3.56. Cuadro: Riesgo 3

Retrasos en tareas del proyecto	
Descripción	Puede generarse una acumulación en tareas retrasadas en el tiempo y no lograr ajustarse a la planificación y objetivos establecidos.
Plan de prevención	Hacer un seguimiento preciso de cada tarea para que sea terminada en los plazos establecidos.
Plan de contingencia	Ampliar las fechas límite de finalización de las tareas retrasadas de tal manera que no se dejen para el final del proyecto o incompletas.
Probabilidad	Alta
Impacto	De medio a muy alto (dependiendo del caso)

3.57. Cuadro: Riesgo 4

Pérdidas de conexión a internet	
Descripción	En cualquier momento puede darse una pérdida de comunicación con herramientas online por la pérdida de conexión a internet, así como la imposibilidad de gestionar código o documentación guardada en la nube.
Plan de prevención	Conectarse a líneas de internet estables y con suficiente velocidad, y usar preferentemente una conexión vía Ethernet en vez de tecnología Wi-Fi.
Plan de contingencia	Cambiar de punto de acceso a internet o reiniciar los recursos proveedores de internet como routers o puntos de acceso móviles.
Probabilidad	Alta
Impacto	Muy bajo

3.58. Cuadro: Riesgo 5

Retraso en la recepción de los datos de entrada	
Descripción	Los datos de entrada definitivos con los que se realizará el modelo predictivo se van a recibir unos meses después del inicio del proyecto, por las complicaciones surgidas al data-manager en la extracción de los pacientes con un primer evento de FA y etiquetado de recurrencia de FA.
Plan de prevención	Realizar una planificación para comenzar el desarrollo del proyecto con las primeras muestras de los datos, a pesar de que sean pocos y provisionales.
Plan de contingencia	Llevar a cabo el desarrollo inicial planificado con el conjunto de datos provisional.
Probabilidad	Muy alta
Impacto	Bajo

3.59. Cuadro: Riesgo 6

Falta de datos para el modelo predictivo con ECGs (ECG-PRAFAI)	
Descripción	Los ECGs de pacientes de la OSI del hospital con un primer evento de FA deben extraerse de sistemas MUSE. Pueden surgir complicaciones a la hora de obtener esos datos por parte del data-manager, ya sea por falta de permisos o imposibilidad de extracción de estos ECGs en formato legible (MUSE-XML) para el posterior aprendizaje automático.
Plan de prevención	Realizar una planificación para buscar conjuntos de datos públicos de ECGs que sean equivalentes a los que deberíamos obtener de la OSI del hospital de la red de salud vasca con el que estamos trabajando.
Plan de contingencia	Llevar a cabo la tarea con el conjunto de datos equivalente encontrado.
Probabilidad	Muy alta
Impacto	Alto

3.60. Cuadro: Riesgo 7

No encontrar un dataset de ECGs equivalente al de nuestra tarea (ECG-PRAFAI)	
Descripción	Puede pasar que no se encuentre un conjunto de datos disponible con ECGs de pacientes tras un primer evento de FA. En este caso, no podría crearse un modelo predictivo de recurrencia de FA.
Plan de prevención	Buscar un conjunto de datos de ECGs para una tarea de clasificación de arritmias entre las que se encuentre la fibrilación auricular.
Plan de contingencia	Hacer un cambio de objetivo y crear un modelo de clasificación de FA (en vez de predicción de recurrencia), y realizar experimentos que puedan ser extrapolables a la tarea de recurrencia de FA.
Probabilidad	Alta
Impacto	Muy alto

3.61. Cuadro: Riesgo 8

4. Descripción, diseño y desarrollo de la solución

En esta sección se va a llevar a cabo el diseño y desarrollo del proyecto tras la planificación y el análisis realizado en el apartado anterior. Concretamente, se va a comenzar con una descripción del corpus y un estudio de las variables predictoras que se van a emplear, seguido de una explicación detallada del propio diseño y desarrollo de la solución propuesta.

4.1. Descripción del corpus

Para llevar a cabo los objetivos, tras la aprobación del Comité de Ética del hospital de la OSI de la red de salud vasca con el que se está trabajando, se ha configurado una base de datos de pacientes a los que se les ha diagnosticado fibrilación auricular en ese mismo hospital desde el 1 de enero de 2015 hasta el 31 de julio de 2018. Los datos de estos pacientes han sido recogidos hasta finales de 2020 (plazo aproximado de 2 años). Los pacientes han sido localizados por el Departamento de Estadística buscando por el código CIE-9 de fibrilación auricular (código 427.31).

Todo el contenido de la base de datos ha sido anonimizado mediante un código de identificación aleatorio (diferente al código de identificación del paciente -CIC-). De esta forma, se garantiza el artículo 3 de la Ley 14/2007 de Investigación Biomédica, que se refiere al proceso por el cual ya no es posible establecer, por medios razonables, el vínculo entre un dato y el sujeto real. Inicialmente, para la recogida de todas las variables necesarias, se ha obtenido una lista del Departamento de Estadística que incluye el CIC de los pacientes para buscar e incluir los datos ecocardiográficos y si la FA permanece o no 2 años después del primer evento. Esta lista solo estuvo en posesión del gestor de datos (data-manager) e investigadores del hospital de la OSI de la red de salud vasca con la que estamos trabajando hasta que se recopilaron los datos antes mencionados y se anonimizó la base de datos final, la cual se nos ha ofrecido para el desarrollo de este proyecto. Por lo tanto, se ajusta a los criterios del modelo de recopilación de datos seudonimizados.

4.1.1. Criterio de inclusión

Todos los pacientes con una primera evidencia de código CIE-9 de fibrilación auricular en la OSI del hospital del 1 de enero de 2015 al 31 de julio de 2018. Se han tenido en cuenta nuevas terapias anticoagulantes en este período para incluir casos posiblemente mal codificados. Por lo tanto, también se ha recopilado la información de los pacientes que han sido tratados con anticoagulantes y no aparecían con un diagnóstico de FA, ya que la anticoagulación es el principal tratamiento de la fibrilación auricular.

4.1.2. Criterio de exclusión

Ausencia de contacto con el Sistema de Salud o no derivación a FA tras el diagnóstico.

4.1.3. Tamaño de la muestra

Los datos preliminares muestran un conjunto de 16.268 pacientes que han sido diagnosticados con una nueva aparición de fibrilación auricular desde enero del 2015 hasta septiembre de 2020 en la región OSI del hospital. Conseguidos esos diagnósticos, en el estudio han sido incluidos los 11.822 diagnósticos hasta julio de 2018 (ya que la recurrencia a los 2 años será la variable de salida). En el preprocesamiento de los datos en bruto, alrededor de un 50-60 % de los datos se ha excluido, principalmente por electrocardiogramas

previos de fibrilación auricular o ausencia de registros electrocardiográficos o de FA documentados tras el primer episodio. Por lo que, en un principio, se estimaba una muestra cercana a 7.000 pacientes.

A la hora de comenzar con el proyecto, se recibió un conjunto de alrededor de 2700 pacientes con un primer evento de FA. Sin embargo, solamente 270 pacientes estaban clasificados con recurrencia de FA o no en un plazo de 2 años. El data-manager sería el encargado de completar la muestra.

Finalmente, de los 2700 pacientes recibidos solamente han sido etiquetados con recurrencia de FA o no **523 pacientes**. A priori es una cantidad de ejemplos muy baja para una tarea de aprendizaje automático, pero es el conjunto de datos proporcionado y no se han podido obtener más ejemplos de entrenamiento. Las razones por las que no se han podido etiquetar el resto de ejemplos son las siguientes:

- En muchos de los pacientes no se han obtenido suficientes datos en el plazo de 2 años desde el primer diagnóstico de FA para decidir si permanece en la arritmia.
- En otros pacientes existen diferentes razones: el paciente ha fallecido en el plazo de 2 años, la información o el seguimiento del paciente son privados, solo hay diagnósticos previos...

4.2. Estudio de variables/rasgos

A continuación, se listan los rasgos o variables predictoras seleccionadas por el data-manager para la caracterización de los ejemplos de la muestra. Por cada variable se da una explicación de por qué podría relacionarse con la predicción de recurrencia de FA.

4.2.1. Demográficas

- **Edad** (en el momento del diagnóstico de FA):

La edad es una variable que se relaciona de forma directa con la incidencia de fibrilación auricular. Es por esto que se espera que sea una variable predictora importante para la permanencia o recurrencia de la arritmia, y como tal está incluida en la mayoría de *scores* relacionados con la persistencia en fibrilación auricular como el Hatch *score*, tal y como muestra el estudio *Barrett et al. [3]*. La explicación fisiopatológica reside en que los cambios electrofisiológicos que suceden en pacientes de edad más avanzada (fibrosis miocárdica, dilatación auricular, rigidez de aurícula izquierda o alteraciones eléctricas relacionadas con la edad) favorecen la progresión de la arritmia, como se recoge en la publicación *Blum et al. [5]*.

- **Género:**

La fibrilación auricular es más frecuente en varones, aunque el sexo femenino se relaciona con mayor permanencia en la misma, tal y como se menciona en el estudio *Roh et al. [58]*. Lo incluyen 3 de los *scores* vigentes (CHADS₂-VASc, CAAP-AF, MB-LATER).

- **Situación de trabajo:** pensionista o no

Esta variable no se ha asociado de forma directa con el desarrollo ni permanencia en FA hasta la fecha.

- **Estado de dependencia:** ingresado en residencia o no

Esta variable no se ha asociado de forma directa con el desarrollo ni permanencia en FA hasta la fecha.

- Características generales: **peso**, **talla**, **IMC** (media de todos los valores obtenidos en el tiempo por cada paciente)

El índice de masa corporal se calcula a través del peso y la talla ($\text{peso}(\text{kg})/\text{talla}(\text{m}^2)$). Si conviene categorizar la variable puede hacerse según los siguientes puntos de corte con los que se clasifica la obesidad (mencionado en los artículos *Proietti and Boriany [56]* y *Wong et al. [72]*):

Descripción	IMC
Desnutrición severa	≤ 16
Desnutrición moderada	(16, 18.5)
Bajo peso	[18.5, 22]
Peso normal	(22, 25)
Sobrepeso	[25, 30]
Obesidad tipo I	[30, 35)
Obesidad tipo II	[35, 40)
Obesidad tipo III	[40, 70]

4.1. Cuadro: Categorización del IMC

En un *score* de medida de recurrencia de FA tras crioablación el IMC se incluye como uno de los factores predictores para la misma (BASE-AF2). En cambio, hay disparidad entre los metaanálisis que evalúan la influencia del IMC sobre la recurrencia de la FA, aunque los más recientes abogan por su asociación. Recientes estudios hablan de la ‘paradoja de la obesidad’ que hace referencia a que, a pesar de ser un factor favorecedor del desarrollo de la FA, no lo es de algunos eventos adversos relacionados con la misma (mortalidad cardiovascular y riesgo tromboembólico). Así está recogido en las publicaciones *Deng et al. [18]* y *Zhuang et al. [80]*.

4.2.2. Historia médica personal

Los datos recogidos en estas variables son previos al primer diagnóstico de FA del paciente. Todas son variables binarias, las cuales indican si el paciente ha padecido esta patología (o consume la sustancia especificada).

- **Consumo de alcohol:**

Aunque algunos estudios de pequeñas cohortes no han encontrado relación entre el consumo de alcohol y la progresión de la FA tras algún intervencionismo, la mayoría de literatura [59] apoya que es uno de los principales factores de progresión.

- **Fumador:**

El cese del hábito tabáquico constaba entre las variables de la cohorte prospectiva del ARREST-AF [53], en el que se comparaba el control de factores de riesgo y la recurrencia de fibrilación auricular en pacientes sometidos a intervencionismo.

- **Otra drogodependencia:**

No valorado de forma exclusiva en estudios previos, se considera una variable a tener en cuenta en la recurrencia de FA.

- **Depresión, ansiedad:**

La presencia de estos factores en los pacientes con FA está descrita en la literatura [34] e interesa su relación con la progresión o recurrencia de la misma.

■ Demencia:

La direccionalidad de mayor demencia en pacientes con FA es conocida en la literatura [62], pero no están descritos estudios direccionados en relación inversa. Además, una parte de la poblaciónañosa (con mayor riesgo de desarrollo de FA) presenta este tipo de patología neurodegenerativa por el envejecimiento poblacional, por lo que conocer en qué medida afecta a la progresión de la FA es de especial relevancia.

■ Insuficiencia renal:

La enfermedad renal crónica (ERC) es uno de los factores de recurrencia clásicos de la fibrilación auricular, incluido en el registro europeo de recurrencia a largo plazo tras ablación de FA [68].

■ Osteoporosis:

La osteopontina y osteoprotegerina se han postulado como posibles marcadores de riesgo de recurrencia post-procedimiento. El papel de la osteoporosis no está descrito como tal en la literatura [8].

■ Menopausia:

En la publicación *Lin et al.* [43] se ha valorado la ablación en pacientes pre- y post-menopáusicas sin variaciones significativas.

■ EPOC, SAOS:

Algunos *scores* incluyen estas patologías respiratorias como el HATCH *score*, que es uno de los modelos que trata de predecir recurrencia desde primer evento como busca este estudio. Así se recoge en el estudio *Naruse et al.* [49]. En el caso del EPOC, hay artículos (por ejemplo, *Gu et al.* [21]) que hablan de la mejoría de calidad de vida con la ablación a pesar del aumento de las recurrencias con la enfermedad.

■ Diabetes tipo 1, diabetes tipo 2, hipertiroidismo:

Recientes artículos (por ejemplo, *Higa et al.* [24]), hacen referencia a la importancia del control de la patología endocrinológica de base en el control de ritmo, por lo que la valoración del tratamiento durante el periodo tendrá especial relevancia.

4.2.3. Historia cardiovascular específica

Los datos recogidos en estas variables son previos al primer diagnóstico de FA del paciente. Todas son variables binarias, las cuales indican si el paciente ha padecido esta patología.

■ Dislipidemia, hipercolesterolemia, síndrome metabólico:

Algunos factores de riesgo cardiovascular clásicos como el LDLc no son los clásicamente relacionados con el desarrollo ni progresión de la FA, pero hay artículos (por ejemplo, *Li et al.* [42]) que lo relacionan. El síndrome metabólico está más instaurado en los *scores* predictivos.

■ Hipertensión arterial, cardiopatía isquémica:

La hipertensión arterial y cardiopatía isquémica constan entre los principales factores de riesgo para el desarrollo de fibrilación auricular mediante alteraciones estructurales. Ambos están incluidos en alguno de los *scores* (cardiopatía isquémica en CAA-P-AF y CHA₂DS₂-VASc, e hipertensión arterial en HATCH). Así está recogido en el estudio *Lau et al.* [40].

■ **Insuficiencia cardiaca:**

Varios scores incluyen la insuficiencia cardiaca como factor de riesgo de recurrencia de FA (HATCH, APPLE, CHA₂DS₂-VASc).

■ **Fibrilación-palpitación, flutter:**

Son diagnósticos y códigos distintos a FA de forma estricta y se valorarán de forma independiente a ésta.

■ **Ictus:**

La estrecha relación entre el ictus cardioembólico y la fibrilación auricular hace muy interesante la percepción de ictus previo en un paciente que debutá con FA como factor de riesgo de desarrollo de fibrilación auricular. Así se ha expuesto en varios artículos desde 2005 como *de Vos et al.* [15] y *Dalen and Alpert* [14].

■ **Miocardiopatía:**

Independientemente de la disfunción que provoquen condicionan una alteración morfológica en probable relación con el posible desarrollo de FA. ALARMEc lo incluye en su *score* predictivo.

■ **Otras arritmias, bradicardia, bloqueo auriculoventricular, bloqueos de rama, contracciones prematuras, ectópica, extrasistólica, posoperatoria, psicógena, ritmo bigeminal, sinus coronaria:**

De estas variables, el bloqueo auriculoventricular es el único presente en otros *scores*, y es de utilidad valorar el sustrato arrítmico previo y el tipo del mismo para la posible predicción de recurrencia arrítmica. Así lo recoge la publicación *Park et al.* [52].

■ **Enfermedades de la válvula mitral reumáticas, otras valvulopatías, valvulopatía mitral congénita:**

La mitral reumática y otras severas se incluyen en los *scores*, tal y como se muestra en el artículo *Wu et al.* [73].

■ **Arteriopatía periférica:**

Incluida en el CHA₂DS₂-VASc, es una variable a tener en cuenta.

4.2.4. Hospitalizaciones

En estas variables se recoge información acerca de las hospitalizaciones de los pacientes con un primer evento de FA.

- Número de ingresos
- Cuántos días antes del evento de FA fue el último ingreso
- Cuántos días ingresó
- Ingreso previo al evento de FA por una patología cardiaca

4.2.5. Pruebas de laboratorio

En estas variables se recogen los resultados obtenidos en las pruebas de laboratorio realizadas a los pacientes con un primer diagnóstico de FA. Son variables numéricas, y se ha calculado para cada paciente la media aritmética de cada una de las pruebas realizadas, anteriores y posteriores al evento de FA.

Las pruebas que se han tenido en cuenta son las siguientes: urea, creatinina, albúmina, glucosa, ácido úrico, HbA1c, natremia, potasio, calcio, colesterol total, HDL, no HDL, LDL, BNP, NT-proBNP, troponina (TNI), troponina (TNT), dímero D, fibrinógeno, aldosterona, PCR, VSG, leucocitos, PCT, TSH, T3 y T4L.

Los siguientes parámetros analíticos se relacionan con diagnósticos explicados los apartados de historia médica personal e historia cardiovascular del paciente: urea, creatinina, glucosa, HbA1c, potasio, colesterol total, HDL, no HDL, LDL, BNP, NT-proBNP, troponina (TNI), troponina (TNT), TSH, T3, T4L.

El ácido úrico es un parámetro analítico estudiado con potencial de predicción de recurrencia de FA, y hay trabajos (por ejemplo, *Canpolat et al. [7]* y *Tsiachris et al. [69]*) que hacen referencia a los parámetros inflamatorios (PCR, VSG, leucocitos, PCT) como posibles predictores.

4.2.6. Tratamientos (códigos ATC)

Los datos de estas variables se han recogido desde un año antes y posteriores al evento de FA del paciente. Todas son variables binarias, las cuales indican si el paciente ha sido medicado con el fármaco específico.

- Fármacos con código ATC **C01, C02, C03, C04, C07, C08, C09, C10:**

Son fármacos del aparato cardiovascular (menos vasoprotectores) que aportan información sobre diagnósticos previos que no hayan sido codificados y pueden ser buenos predictores directos o indirectos de la recurrencia.

- Fármacos **C01B:**

Un análisis segmentado de los pacientes que han estado tomando fármacos C01B (antiarrítmicos clase I o III) es imprescindible. Estos fármacos son, junto a los procedimientos ablativos, los tratamientos manejados con control de ritmo.

- Por la información directa e indirecta que pueden aportar, también se consideran de interés los siguientes grupos farmacológicos:
 - Antiagregantes, anticoagulantes (**B01A**)
 - Ácido acetilsalicílico (**N02BA01**)
 - Inhibidores de la bomba de protones (**A02BC**)
 - Diuréticos (**C03**)
 - Estrógenos (**G03A**)
 - Antidiabéticos (**A10 fármacos usados en diabetes**)
 - Antidepresivos (**N06A**)
 - Antipsicóticos (**N05A**) y benzodiacepinas (**N05B**)
- **Polimedición:** indica si el paciente ha sido tratado con 5 o más fármacos de los anteriores.

4.2.7. Procedimientos

- **Ecocardiograma, ecocardiograma con contraste, electrocardiograma**

Los datos recogidos en estas variables son previos al evento de FA del paciente. Son variables binarias, las cuales indican si al paciente se le ha realizado el procedimiento específico.

- **FEVI, diámetro de aurícula izquierda, área de aurícula izquierda**

Los datos de estas variables se han recogido en el momento del diagnóstico de FA del paciente. Son variables numéricas.

La realización de las pruebas puede tener relevancia porque ha podido haber sospecha previa de cardiopatía estructural. La FEVI y diámetro de aurícula izquierda son predictores muy importantes ampliamente descritos.

4.2.8. Intervenciones

Los datos recogidos en estas variables son posteriores al evento de FA del paciente. Son variables binarias, las cuales indican si al paciente se le ha realizado la intervención específica.

- **Ablación por radiofrecuencia (por catéter)**
- **Cardioversión (eléctrica o farmacológica) / CONTRASHOCK**
- **Colocación de marcapasos o DAI** (previo o tras el evento)

4.3. Diseño y desarrollo de la solución

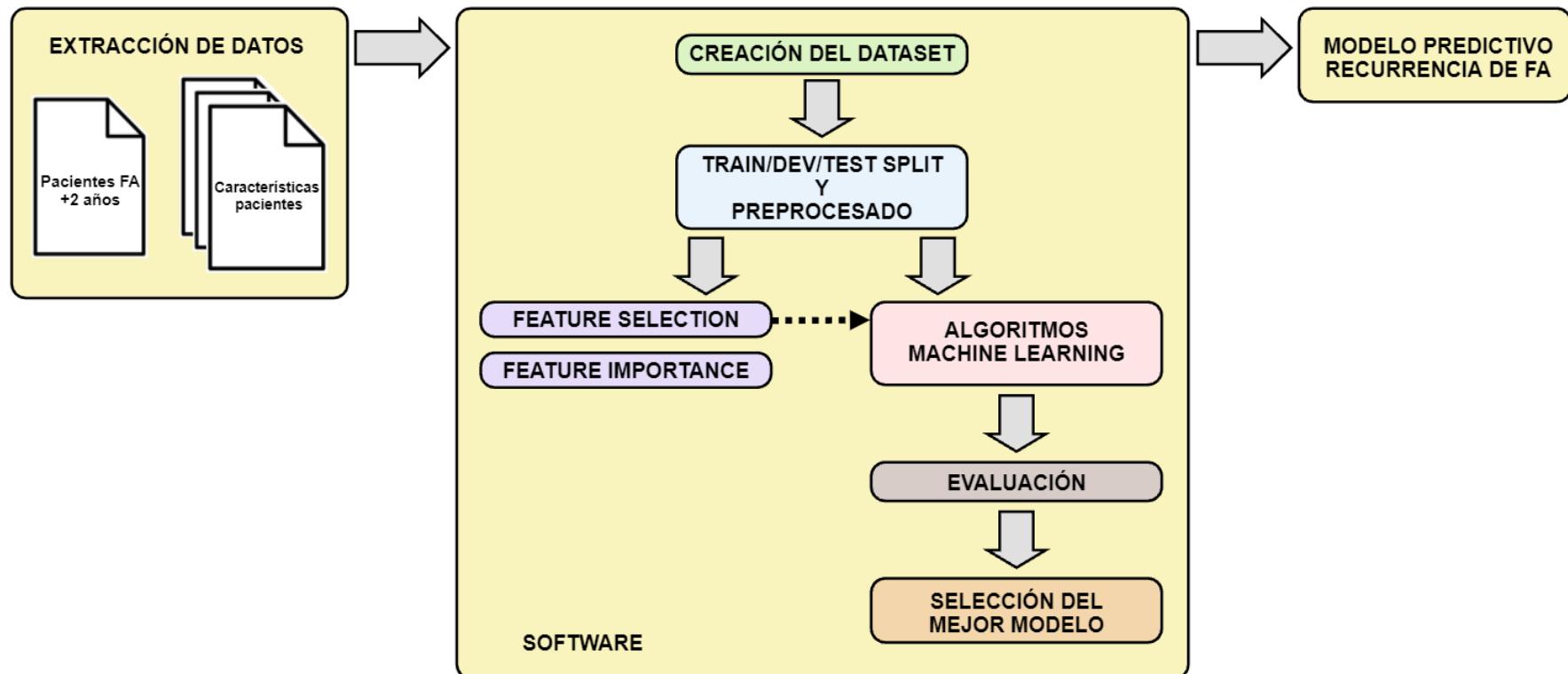
En esta sección se va a llevar a cabo una explicación detallada de la solución diseñada y desarrollada. Antes de entrar en la explicación del diseño de la solución, como ya se ha visto en la sección *Arquitectura*, hay que conocer el *input* y el *output* del software a desarrollar.

- **Input:** fichero que contiene los códigos de identificación de los pacientes con un primer episodio de FA y la variable de salida (*clase*) que indica si el paciente permanece en la arritmia o no tras 2 años, y múltiples ficheros con datos de pacientes de los que se deben extraer los rasgos (*features*) para la creación del *dataset*.

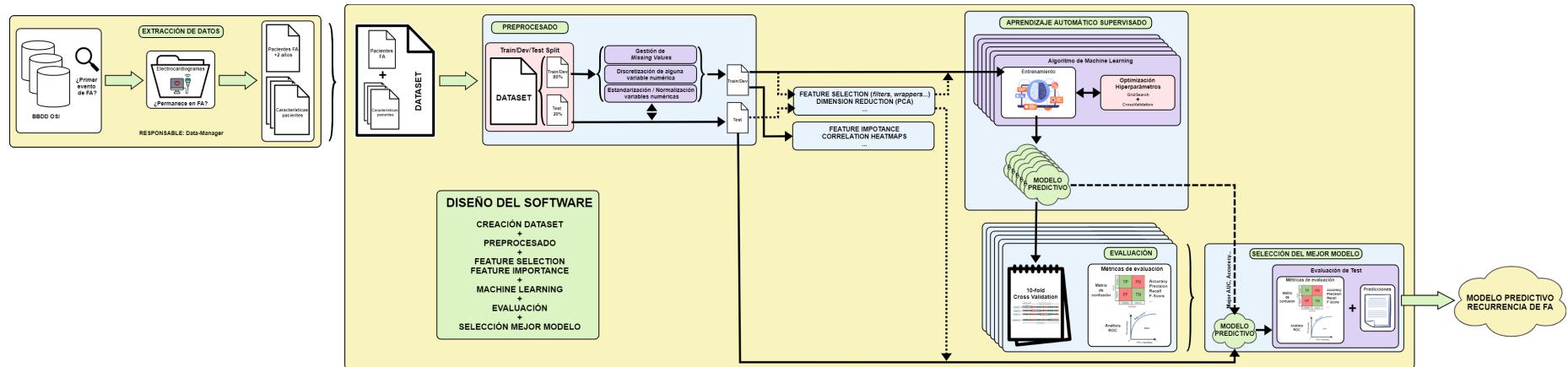
- **Output:** modelo predictivo óptimo de recurrencia de FA.

El desarrollo de la solución consta de varias fases en las que se han llevado a cabo las siguientes tareas: creación del *dataset*, división del conjunto de datos en entrenamiento/desarrollo/test (Train/Dev/Test) y preprocesado de datos, selección e importancia de variables, entrenamiento de modelos con diferentes algoritmos de ML, evaluación de los modelos y selección del mejor modelo. El diseño del software resumido en las anteriores tareas está reflejado en la Figura 4.1.

En la Figura 4.2 se muestra el diseño de la solución detalladamente. En los próximos apartados se va a dar una explicación en profundidad de cada una de las fases del diseño y desarrollo.



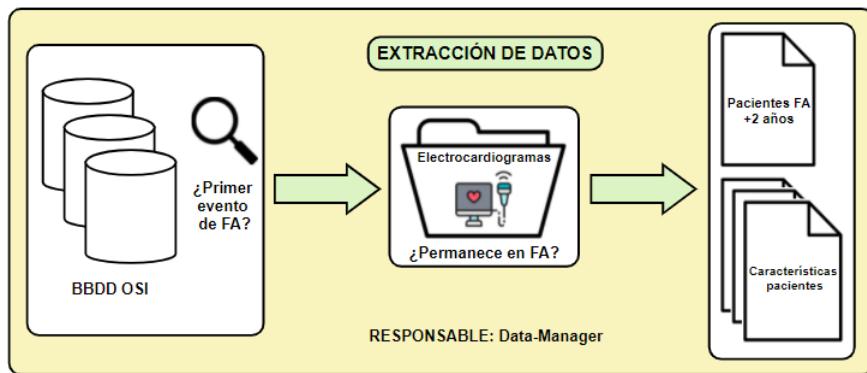
4.1. Figura: Diseño de la solución (resumido en bloques de tareas)



4.2. Figura: Diseño de la solución (detallado)

4.3.1. Extracción de datos

En este apartado se da una explicación detallada de la fase de extracción de los datos de entrada (ver Figura 4.3).



4.3. Figura: Extracción de datos

La extracción de los datos que van a ser la entrada a nuestro software es trabajo del data-manager del proyecto. Sin embargo, es necesario dar a conocer el proceso de selección y extracción de estos datos para comprender el contexto del proyecto.

El gestor de los datos (data-manager) es uno de los investigadores de la OSI del hospital participante en el proyecto y, como ya se ha explicado en la sección *Descripción del corpus*, ha sido el encargado de recopilar y anonimizar la información de los pacientes con un primer evento de FA desde el 1 de enero de 2015 hasta el 31 de julio de 2018.

Para la extracción de estos pacientes se ha hecho una búsqueda en la base de datos de la OSI del hospital mediante el código CIE-9 perteneciente a la fibrilación auricular (427.31). Sin embargo, muchos de los pacientes que sufren FA no han sido registrados en el sistema con ese diagnóstico. Por lo tanto, se han tenido en cuenta los pacientes con nuevas terapias anticoagulantes en ese periodo de tiempo y que no aparecían con un diagnóstico de FA, para incluir casos posiblemente mal codificados, ya que la anticoagulación es el principal tratamiento de la fibrilación auricular.

Por otro lado, también es conveniente dar una explicación de los diferentes tipos de FA que se han considerado en el estudio.

- **Fibrilación auricular permanente:** De larga duración y permanece en el paciente el resto de su vida. La arritmia no se detiene con medicamentos ni mediante la cardioversión controlada. Sin embargo, existen casos de FA permanente que en ecocardiogramas posteriores se ha detectado una mejora en el ritmo sinusal.
- **Fibrilación auricular paroxística:** De breve duración, aparece y desaparece de forma espontánea.
- **Fibrilación auricular persistente:** Tras un periodo de tiempo desde el primer episodio de FA se detecta que la arritmia no ha desaparecido por sí sola. Es necesaria la toma de medicamentos para controlar el ritmo cardíaco, o bien recurrir a técnicas como la cardioversión para que el corazón vuelva a su ritmo sinusal normal.

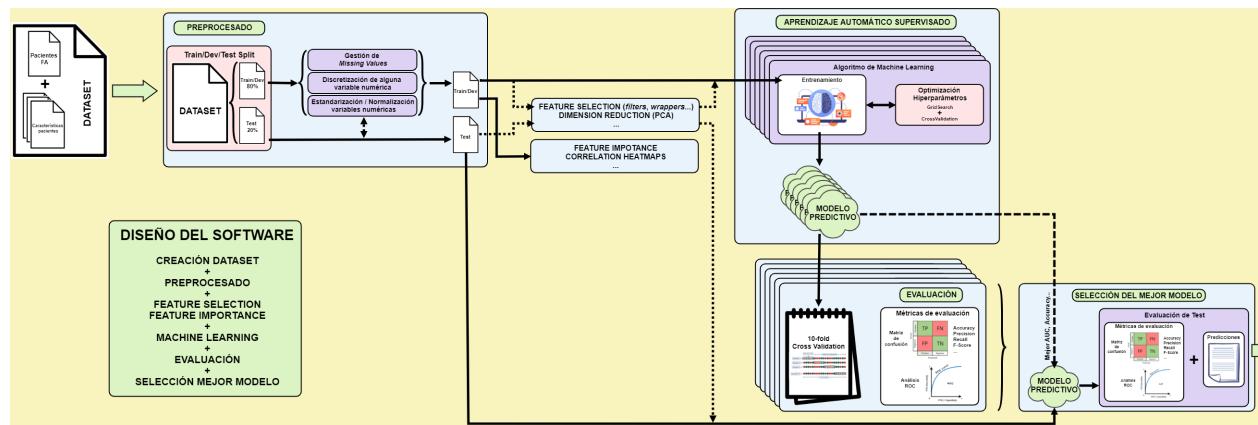
Finalmente, para determinar si el paciente permanece en la arritmia o no después del primer evento de FA, y de esta manera, determinar la variable de salida de recurrencia de FA, se ha seguido el método explicado a continuación. Por cada paciente se han estudiado todos los electrocardiogramas realizados entre 6 meses y 3 años desde el primer diagnóstico de FA. La complicación que ha supuesto este estudio ha sido que no a todos los pacientes se les ha realizado el mismo número de electrocardiogramas ni ha transcurrido el mismo tiempo después del diagnóstico. Por ejemplo, a un paciente se le ha realizado un solo electrocardiograma pasado un año, y a otro paciente se le han realizado cuatro en un periodo de dos años. Por lo tanto, examinando los resultados de los electrocardiogramas disponibles y otro conjunto de

variables de influencia como el tratamiento con fármacos C01B, se ha tomado la decisión de determinar si hay recurrencia de FA o no (vuelta al ritmo sinusal normal). Esta es una de las razones por las se ha reducido tanto el tamaño de la muestra, tal y como se ha explicado en la sección *Tamaño de la muestra*.

La extracción de datos termina con la creación del documento que contiene los aproximadamente 500 pacientes con un primer evento de FA y la variable que indica si hay recurrencia o no de FA después de un plazo aproximado de 2 años. También se extraen de la base de datos de la OSI del hospital un conjunto de documentos que contienen los valores de los rasgos (*features*) a utilizar en el modelo de ML (las variables descritas en el apartado *Estudio de variables/rasgos*).

4.3.2. Software diseñado

En este apartado se da una explicación detallada del software diseñado (ver Figura 4.4).



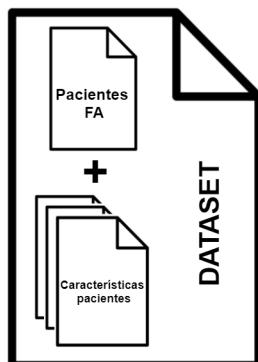
4.4. Figura: Software diseñado

4.3.2.1. Creación del *dataset*

La fase de creación del *dataset* consiste en crear una base de datos en formato tabulado que recoja para cada paciente los valores de las 100 variables predictoras establecidas en el momento del diagnóstico de FA. La información de los pacientes que se debe añadir a las variables de la base de datos se extrae de múltiples ficheros ofrecidos por el data-manager.

De esta manera, se crea un conjunto de datos adaptado para el posterior aprendizaje automático a realizar sobre él. Para cada paciente se unifican los datos de los diferentes ficheros, creando un vector de rasgos junto con la clase (recurrencia o no de FA).

Como se muestra en la Figura 4.5, el *dataset* está formado por los vectores de rasgos de cada instancia creada al combinar la información de los pacientes con recurrencia de FA (o no) a los 2 años y sus características.



4.5. Figura: Diseño de la creación del *dataset*

Una vez creado el *dataset*, el vector de rasgos esta formado por las siguientes variables:

- id
- fecha
- tipo_fa
- urea
- creatinina
- albumina
- glucosa
- hba1c
- potasio
- calcio
- hdl
- no_hdl
- ldl
- colesterol
- ntprobnp
- bnp
- troponina_tni
- troponina_tnt
- dimero_d
- fibrinogeno
- aldosterona
- leucocitos
- pct
- tsh
- t4l
- sodio
- vsg
- t3l
- ecocardiograma
- ecocardiograma_contraste
- fevi
- diametro_ai
- area_ai
- cardioversion
- ablacion
- marcapasos_dai
- numero_ingresos
- etilogia_cardiologica
- numero_dias_desde_ingreso_hasta_evento
- numero_dias_ingresado
- depresion
- alcohol
- drogodependencia
- ansiedad
- demencia
- insuficiencia_renal
- menopausia
- osteoporosis
- diabetes_tipo1
- diabetes_tipo2
- dislipidemia
- hipercolesterolemia
- fibrilacion_palpitacion
- flutter
- insuficiencia_cardiaca
- fumador
- sahos
- hipertiroidismo
- sindrome_metabolico
- hipertension_arterial
- cardiopatia_isquemica
- ictus
- miocardiopatia
- otras_arritmias_psicogena_ritmo_bigeminal
- bloqueos_rama
- bloqueo_auriculoventricular
- bradicardia
- contracciones_prematuras_ectopia_extrasistolica
- posoperatoria
- sinusal_coronaria
- valvula_mitral_reumaticas
- otras_valvulopatias
- valvulopatia_mitral_congenita
- arteriopatia_periferica
- epoc
- genero
- edad
- pensionista
- residenciado
- talla
- imc
- peso
- b01a
- n02ba01
- a02bc
- c03
- g03a
- a10
- n06a
- n05a
- n05b
- c01
- c01b
- c02
- c04
- c07
- c08
- c09
- c10
- polimedicion
- sigue_fa

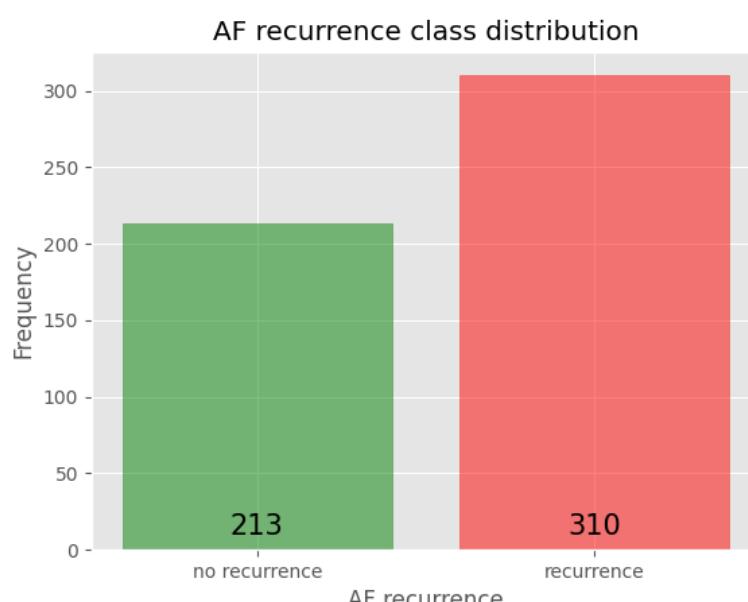
La gran mayoría de variables anteriores son las características explicadas en el apartado *Estudio de variables/rasgos*. Las siguientes variables no se habían descrito todavía:

- *id*: ID del paciente anonimizado (no se va a utilizar en el entrenamiento del modelo).
- *fecha*: Fecha del diagnóstico del primer evento de FA del paciente.
- *tipo_fa*: Tipo de fibrilación auricular diagnosticada.
- *sigue_fa*: Variable de salida (clase) que indica la recurrencia o no de FA.

Respecto al tipo de las variables, existen variables binarias, numéricas y categóricas en el conjunto de datos.

- Entre las **variables binarias** se encuentran todas aquellas relacionadas con la historia médica personal (alcohol, fumador...) e historia cardiovascular específica (dislipidemia, hipercolesterolemia...), así como el género (0, femenino; 1, masculino), pensionista, residenciado, ingreso previo al evento de FA por una patología cardiaca (*etología_cardiaca*), tratamientos (se especifica si han sido medicados o no), procedimientos realizados como ecocardiograma, ecocardiograma con contraste o electrocardiograma, intervenciones realizadas, y recurrencia de FA.
- Entre las **variables numéricas** se encuentran el ID del paciente, fecha del evento de FA, edad, peso, talla, información acerca de hospitalizaciones como el número de ingresos o días ingresado, los resultados de las pruebas de laboratorio, y datos como la FEVI, diámetro de aurícula izquierda o área de aurícula izquierda.
- Entre las **variables categóricas** se encuentran el tipo de FA (0, FA no especificada; 1, FA paroxística; 2, FA persistente; 3, FA permanente) y el IMC (según la clasificación de la obesidad, ver Cuadro 4.1).

Otros aspectos a analizar una vez creado el *dataset* es la distribución de las variables predictoras. Por ejemplo, es de gran importancia saber la distribución de la clase (ver Figura 4.6).

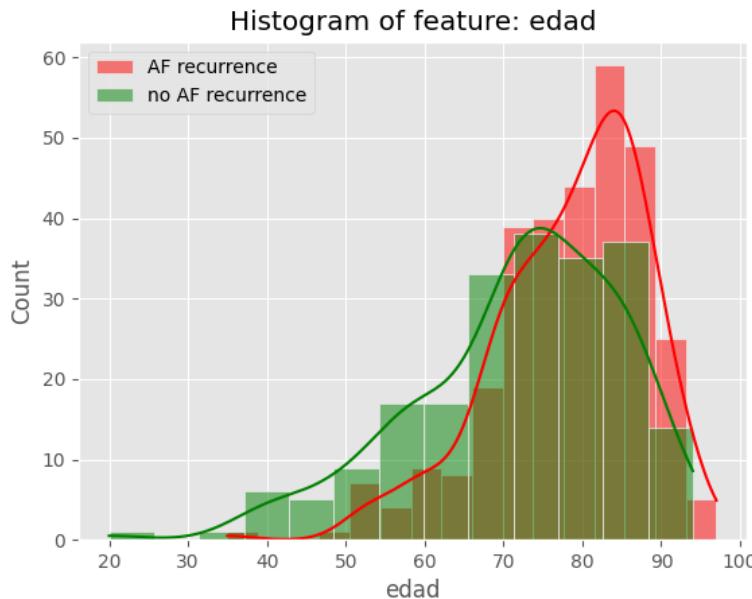


4.6. Figura: Distribución de la clase

Como muestra la Figura 4.6, podemos considerar que el conjunto de datos está suficientemente balanceado. Encontramos 213 ítems con clase *no recurrencia* (40.73 %), frente a 310 ítems con clase *recurrencia* (59.27 %). No consideramos que técnicas como sobremuestreo (*oversampling*) o submuestreo (*undersampling*) sean necesarias al tener un conjunto de datos pequeño y no haber un gran desbalance

en la clase, y preferimos trabajar con el conjunto de datos de los pacientes reales obtenidos, aunque en futuras pruebas podría utilizarse alguna de estos métodos de balanceo para ver cómo afectaría a los resultados (evitando un posible *overfitting*).

También es interesante realizar un análisis de la distribución de algunas de las variables consideradas a priori más influyentes en la recurrencia de FA. En la Figura 4.7 podemos ver la distribución de la variable *edad*.



4.7. Figura: Distribución de la variable *edad*

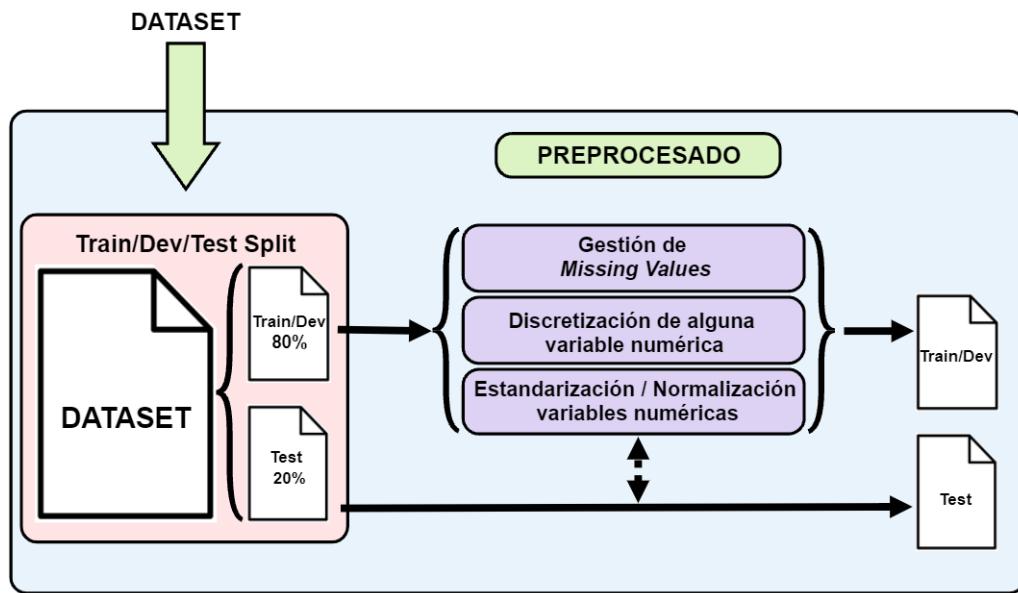
Como muestra el histograma, podemos observar que el rango de 70-90 años es la franja de edad más habitual para el desarrollo de un primer evento de FA. De la misma manera, vemos como en edades menores a 70 años la recurrencia es menos habitual que pasados los 70 años.

4.3.2.2. División del conjunto de datos y preprocessado

Una vez creado el conjunto de datos (*dataset*) para el posterior entrenamiento del modelo predictivo, es necesaria una división de los datos, obteniendo muestras diferentes para el entrenamiento, desarrollo/validación y evaluación del modelo final.

Por otro lado, es preferible que el *dataset* pase por una serie de modificaciones en los datos (preprocessado) con el objetivo de adaptarlo al algoritmo de ML y obtener mejores resultados.

En la Figura 4.8 se muestran las diferentes fases del preprocessado de los datos.



4.8. Figura: Diseño del preprocessamiento y división Train/Dev/Test

4.3.2.2.1 División del *dataset*

Tras la creación del *dataset* al completo, se va a proceder a dividir el *dataset* en tres subconjuntos: conjunto de entrenamiento, evaluación y prueba.

En este estudio, vamos a utilizar el esquema de evaluación *k-fold Cross Validation*, por lo que el conjunto de entrenamiento (Train) y desarrollo/evaluación (Dev) va a ser el mismo (Train/Dev), y el propio método *Cross Validation* es el encargado de ir dividiendo los ítems en estos dos conjuntos mientras se va realizando la evaluación (explicación del esquema *Cross Validation* en el apartado 4.3.2.6.1).

Por lo tanto, el *dataset* al completo de 523 instancias va a dividirse en dos subconjuntos de la siguiente forma:

- El conjunto Train/Dev tendrá el 80 % de las instancias totales (418 ítems), y con él se realizará el entrenamiento y evaluación de los modelos predictivos.
- El conjunto Test tendrá el 20 % restante (105 ítems), y servirá para evaluar el funcionamiento del modelo óptimo sobre instancias nunca antes vistas en el entrenamiento.

Se ha dividido el conjunto de datos en conjuntos Train/Dev y Test de manera que se conservan las mismas proporciones de ejemplos de cada clase que se observan en el conjunto de datos original, es decir, se mantiene la distribución y proporciones de las clases en cada subconjunto (*stratified split*).

4.3.2.2.2 Gestión de *missing values*

Se ha observado que hay un número alto de *missing values* en el *dataset*. Los *missing values* son valores nulos o datos faltantes que suelen aparecer muy frecuentemente en estudios médicos.

El problema es que hay que trabajar con una gran cantidad de *missing values* en múltiples variables como las pruebas de laboratorio (no todos los pacientes tienen todas las pruebas realizadas), peso, talla, IMC... Los algoritmos de clasificación más habituales no están adaptados para el tratamiento de *missing values*, y por ello, se debe seguir una estrategia para trabajar con ellos.

El primer paso ha sido hacer un análisis de los *missing values*. Se debe tener en cuenta cuántos *missing values* hay por cada variable predictora y observar si estos están distribuidos de manera uniforme o heterogénea. Una distribución uniforme significaría que hay un número similar de *missing values* por cada variable que contiene *missing values*. Una distribución heterogénea supondría diferentes cantidades

de *missing values* por cada variable, es decir, algunas variables contienen muchos *missing values*, y otras, pocos. Este análisis podría ser de ayuda a la hora de elegir la estrategia para tratarlos.

Como se muestra en las Figuras 4.9 y 4.10, hay un gran número de *missing values* en muchas de las variables. Son las variables numéricas aquellas que contienen al menos un *missing value*: pruebas de laboratorio, FEVI, diámetro y área de la aurícula izquierda, días de hospitalización, talla, peso e IMC.

La distribución es bastante heterogénea, algunas variables como *urea*, *creatinina* o *glucosa* tienen un porcentaje de *missing values* relativamente bajo (en torno al 15%), pero hay muchas variables que superan el 50% de *missing values*. En las variables *hba1c* y *aldosterona* no hay ningún dato (100% de *missing values*), por lo que van a ser directamente descartadas. El resto de variables van a mantenerse de momento (aunque haya bastantes que superen el 80% de *missing values*) y se harán futuras pruebas eliminándolas con el objetivo de ver si hay una mejoría en los resultados del modelo predictivo entrenado.

A continuación se explican las diferentes estrategias planteadas para tratar los *missing values*:

- La primera estrategia que se ha planteado es la **eliminación de ejemplos** que tienen *missing values*. Es la estrategia más común y sencilla en *datasets* con grandes cantidades de datos, y una de las más usadas cuando se trabaja en NLP (Procesamiento del Lenguaje Natural). Sin embargo, en este proyecto no sería una estrategia aplicable ya que supone dos grandes problemas. Por un lado, el estudio dejaría de ser real porque se eliminarían pacientes, y por otro lado, se dispone de pocos datos en el conjunto de entrenamiento, por lo que eliminar más ejemplos alteraría seriamente el aprendizaje. Por lo tanto, se ha descartado esta estrategia.
- La segunda estrategia sería la imputación de valores a los *missing values* mediante técnicas estadísticas. En el caso de tener *missing values* en variables categóricas o binarias, se aplicaría la **moda** de esa variable a cada *missing value*. En el caso de variables numéricas, se aplicaría la **media aritmética** o la **mediana** de esta variable. Estas estrategias parecen en principio sencillas y bastante coherentes, pero tratándose de características como el peso o la altura de un paciente, no parece muy adecuado asignar la media o la mediana. Aun así, son estrategias que vamos a utilizar, en este caso al ser variables numéricas las que contienen *missing values*, se imputará la media o la mediana.
- En tercer lugar, habría que plantear si se van a utilizar **clásificadores insensibles a los missing values**. Un algoritmo insensible a los *missing values* es capaz de crear un modelo robusto sin tener que tratar los valores desconocidos de antemano. Habría que realizar pruebas para determinar con qué clasificadores se obtienen mejores resultados: con clasificadores insensibles a los *missing values* y sin el tratamiento de ellos, o con clasificadores sensibles a *missing values* y un conjunto entrenamiento sin *missing values*. Ejemplos de algoritmos muy comunes insensibles a *missing values* son *k-Nearest Neighbors*, *Naïve Bayes* o *Random Forest*. Por otro lado, vamos a crear modelos con algunos algoritmos sensibles a *missing values* como *XGBoost* y *LightGBM*.
- Como cuarta estrategia se ha planteado la **predicción de missing values**. Esta estrategia consistiría en crear clasificadores intermedios que sean capaces de predecir los valores de cada una de las variables que contiene *missing values*. El inconveniente de esta estrategia es que se debería entrenar un modelo por cada una de las variables que contiene *missing values* (que en este estudio son bastantes). Además, a la hora de crear el conjunto de entrenamiento para la predicción de una variable, deberían descartarse el resto de variables que también presentan *missing values*, por lo que el aprendizaje podría ser poco robusto.

Para la implementación de esta estrategia de imputación de *missing values* se ha utilizado un regresor lineal, ya que todas las variables que contienen *missing values* son continuas y es el algoritmo de regresión más sencillo. Por lo tanto, cuando se quieran imputar los *missing values* de una variable (*urea*, por ejemplo) mediante predicciones de un modelo de regresión lineal, se deberán seguir los siguientes pasos:

1. Crear un *dataset* solamente con las variables sin *missing values* y la que se quiere predecir (*urea*).

2. Eliminar del *dataset* y guardar las instancias que tienen *missing values* en esa variable (*urea*), para posteriormente predecir su valor.
3. Entrenar un modelo de regresión lineal con las instancias que se han mantenido, aquellas que no tienen *missing value* en la variable a predecir (*urea*).
4. Con el regresor entrenado, predecir los valores de la variable (*urea*) en las instancias que tenían *missing values* guardadas en el paso 2.

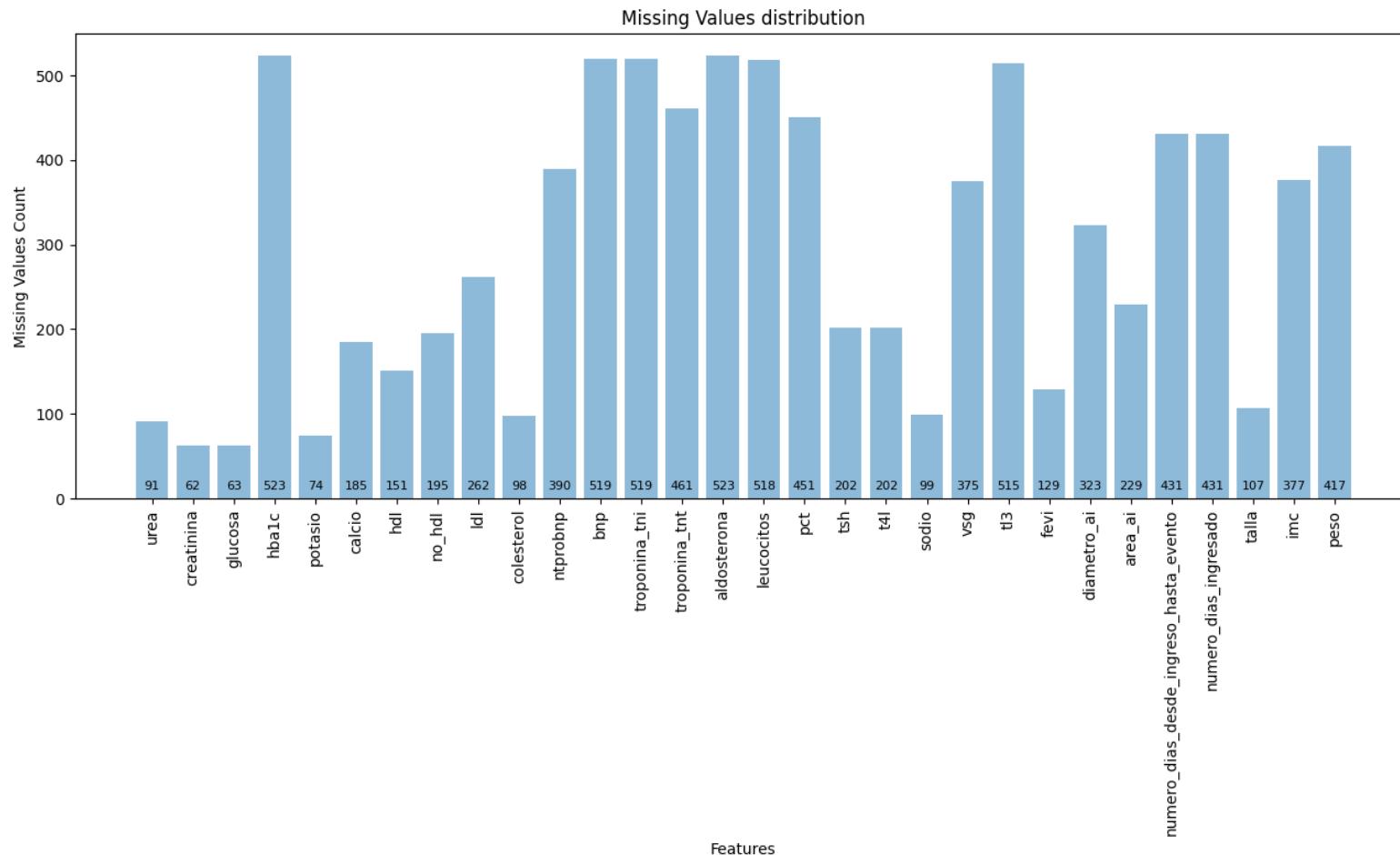
Algunos de los valores predichos puede que no sean correctos. Se han predicho algunos valores negativos en variables de pruebas de laboratorio, lo que sería un resultado incorrecto en una prueba real, por lo que se ha decidido modificar estos valores negativos a 0.

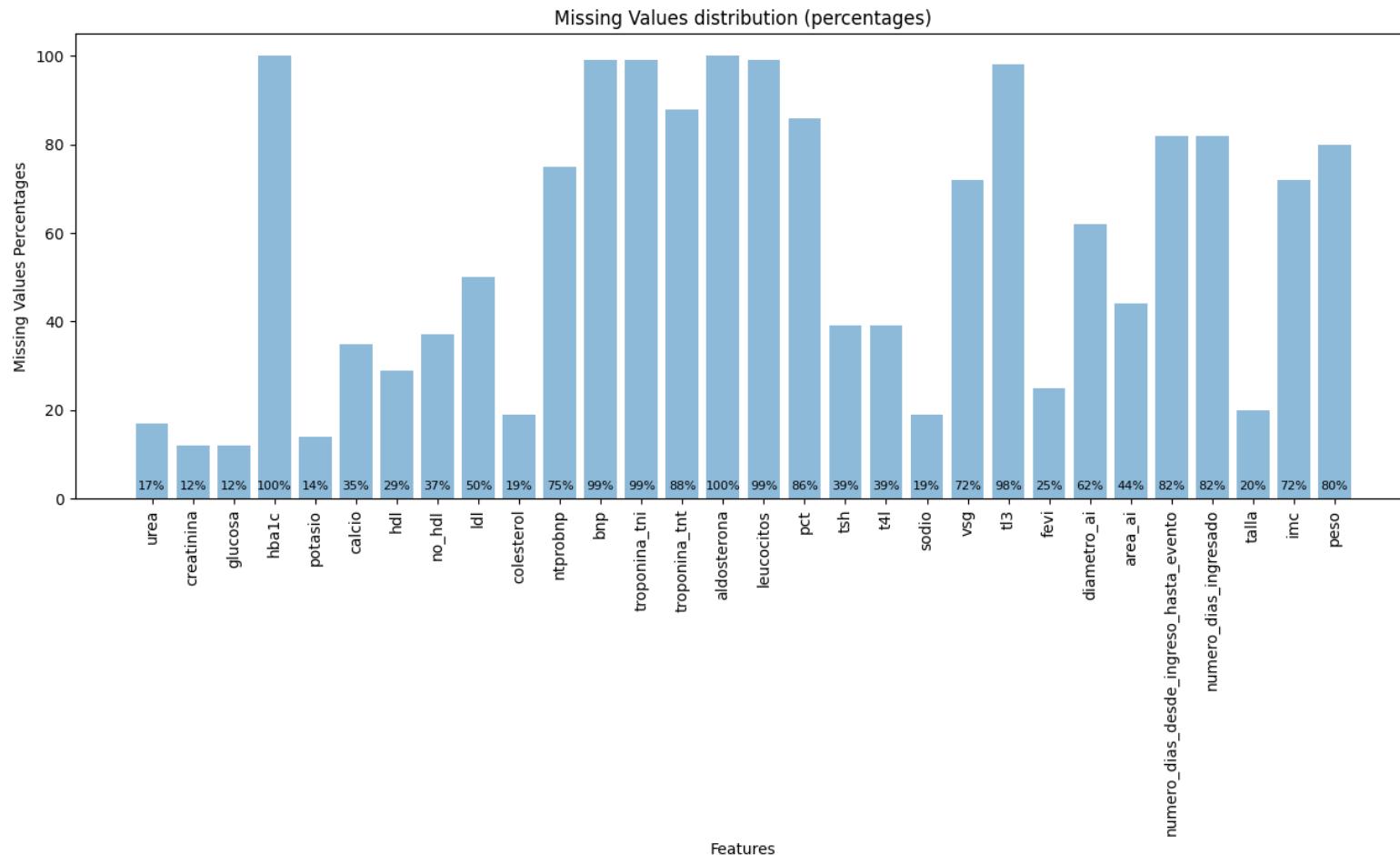
El proceso anterior debe repetirse para todas aquellas variables que tengan *missing values*.

- La quinta estrategia que se ha planteado ha sido la **imputación de un valor diferente** al resto en cada variable con *missing values*. De esta manera, a la hora de entrenar un modelo predictivo, el algoritmo tendría en cuenta todos *los missing values* de la misma manera mediante el valor diferente imputado. Para la implementación de esta estrategia, como las variables con *missing values* son numéricas y con valores no negativos, se ha decidido imputar el valor -1.
- Una sexta estrategia planteada ha sido ***data-augmentation (dropout)***. Esta estrategia se basa en la generación de ejemplos artificiales con valores como la moda, media o mediana. No se va aplicar en este proyecto.

Por lo tanto, tras finalizar la gestión de los *missing values*, hemos obtenido 5 conjuntos de datos diferentes:

- *Dataset* con ***missing values***.
- *Dataset* con imputación de la **media aritmética**.
- *Dataset* con imputación de la **mediana**.
- *Dataset* con imputación de **valores predichos mediante regresores lineales**.
- *Dataset* con imputación de **valores diferentes (-1)**.

4.9. Figura: Distribución de *missing values* por variables

4.10. Figura: Distribución de *missing values* por variables (porcentajes)

4.3.2.2.3 Discretización de alguna variable numérica

Discretizar alguna de las variables continuas puede ser otra técnica de preprocesado de datos, la cual podría servir para reducir la complejidad de los modelos entrenados y mejorar los resultados. La discretización consiste en convertir una variable numérica en categórica.

Según las indicaciones del data-manager el IMC (Índice de Masa Corporal) es una variable que conviene discretizar, ya que en la mayoría de estudios se trata como una variable categórica. La discretización del IMC puede hacerse según los puntos de corte con los que se clasifica la obesidad (ver Cuadro 4.1). De esta manera, se obtienen 8 categorías para el IMC:

- 0: Desnutrición severa, IMC [10,16]
- 1: Desnutrición moderada, IMC (16, 18.5)
- 2: Bajo peso, IMC [18.5, 22]
- 3: Peso normal, IMC (22, 25)
- 4: Sobrepeso, IMC [25, 30]
- 5: Obesidad tipo I, IMC [30, 35]
- 6: Obesidad tipo II, IMC [35, 40]
- 7: Obesidad tipo III, IMC [40, 70]

En futuras pruebas, una vez entrenado el modelo óptimo y analizados los resultados de su evaluación, podría ser interesante discretizar otras variables como la edad o el área de la aurícula izquierda. En el caso de la variable *area_ai*, según indicaciones del data-manager, aunque es recomendable mantener los datos cuantitativos, sería interesante categorizarla de la siguiente forma: ≤ 20 , $(20,30]$, $(30,40]$ y > 40 .

4.3.2.2.4 Estandarización / Normalización de variables numéricas

Una vez aplicados los métodos de gestión de *missing values* anteriores, se aplicarán métodos de estandarización o normalización en los datos. En general, los algoritmos de aprendizaje se benefician de la estandarización del conjunto de datos. Si algunos valores atípicos (*outliers*) están presentes en el conjunto, los transformadores o escaladores robustos son apropiados.

La estandarización o normalización debe aplicarse solamente sobre el conjunto de entrenamiento, ya que introducir también los datos del Test condicionaría el escalado de estas variables con los datos de este subconjunto. Una vez aplicada la estandarización/normalización, deberá aplicarse también en las instancias del Test, pero utilizando los parámetros (media, varianza) de la estandarización/normalización del entrenamiento.

En este proyecto se han probado cinco técnicas diferentes de escalado y normalización de datos para las variables numéricas. Todas ellas han sido implementadas mediante el paquete de preprocesado de datos de la librería Scikit-learn de Python.

- **StandardScaler**: se centran los datos numéricos restando la media aritmética de cada variable, y después, dividiéndola por su desviación estándar.

Standardization:

$$z = \frac{x - \mu}{\sigma}$$

with mean:

$$\mu = \frac{1}{N} \sum_{i=1}^N (x_i)$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

4.11. Figura: Fórmulas para la estandarización mediante *StandardScaler*

- **MinMaxScaler**: se escalan los valores de las variables numéricas en el rango [0, 1], restándole el valor mínimo de la variable, y después, dividiéndolo por la diferencia entre el valor máximo y el mínimo.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

4.12. Figura: Fórmula para la estandarización mediante *MinMaxScaler*

- **MaxAbsScaler**: se escalan los valores de las variables dividiendo el valor absoluto máximo y, de esta manera, escalando las variables numéricas en el rango [-1, 1].

$$x_{scaled} = \frac{x}{\max(|x|)}$$

4.13. Figura: Fórmula para la estandarización mediante *MaxAbsScaler*

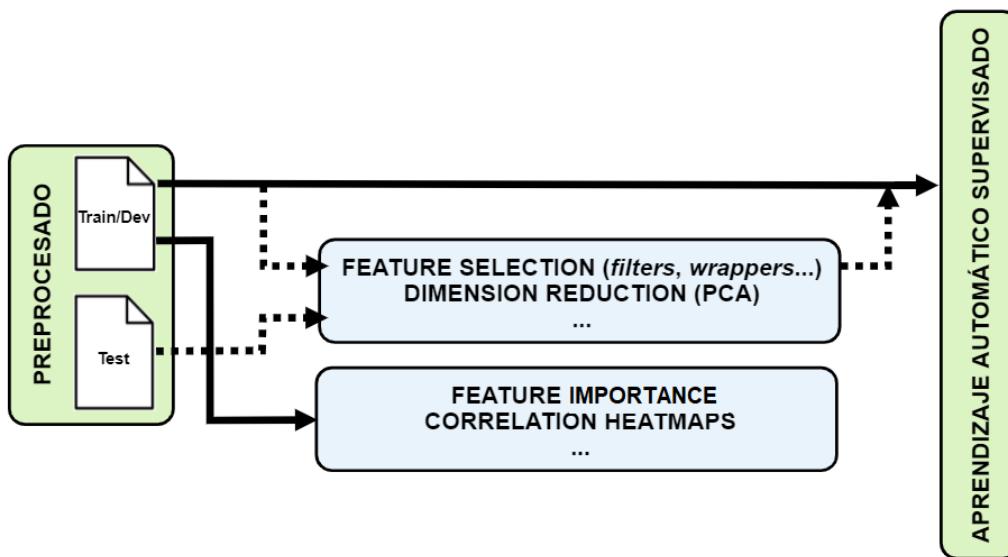
- **QuantileTransformer**: proporciona una transformación no paramétrica para escalar los valores numéricos a una distribución uniforme con valores en el rango [0,1].
- **Normalizer**: la normalización es el proceso de escalar las variables numéricas para tener una norma unitaria (normalización L2, norma Euclídea de 1).

So if $x = (x_1, \dots, x_p)$ is a row of the data matrix, the normalizer changes it to $(x_1/\|x\|_2, \dots, x_p/\|x\|_2)$, where $\|x\|_2 := \sqrt{x_1^2 + \dots + x_p^2}$.

4.14. Figura: Fórmula para la estandarización mediante *Normalizer*

4.3.2.3. Selección de variables / Reducción de dimensiones

Tras el preprocesado de los datos, podría ser interesante modificar el espacio de caracterización de las instancias, ya sea realizando una selección de variables relevantes o reduciendo la dimensión del vector de rasgos. Estas técnicas se aplican sobre el conjunto de entrenamiento, y posteriormente, se lleva a cabo la adaptación del vector de rasgos correspondiente al conjunto Test, ya que los dos conjuntos deben estar caracterizados por el mismo vector de rasgos. En la Figura 4.15 se muestran estas dos posibles técnicas: *feature selection* y *dimension reduction*.



4.15. Figura: Diseño de la selección de variables / reducción de dimensiones e importancia/correlación de las variables

4.3.2.3.1 Feature selection (selección de variables)

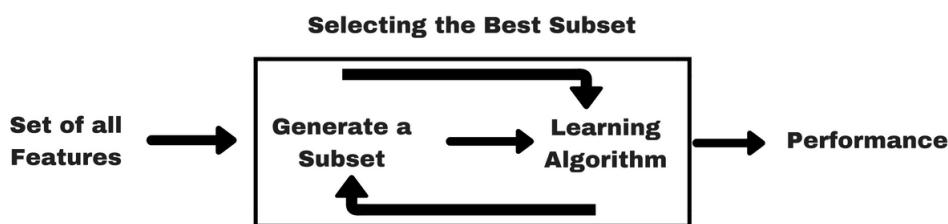
Eliminar variables irrelevantes puede reducir la complejidad en el cálculo de los algoritmos de ML y mejorar la interpretabilidad del modelo. Para la selección de variables se han estudiado tres técnicas diferentes:

- **Filter methods (filtrado):** La selección de variables es independiente de cualquier algoritmo de aprendizaje automático. En cambio, las características se seleccionan sobre la base de sus puntuaciones (*scores*) en varias pruebas estadísticas para su correlación con la variable de salida; es decir, se seleccionarán las variables con mayor *score* tras la prueba estadística. Por ejemplo, puede utilizarse la prueba chi-cuadrado (χ^2), la cual se aplica a pares de características para evaluar la probabilidad de correlación o asociación entre ellas utilizando su distribución de frecuencia.



4.16. Figura: *Filter methods* para selección de variables

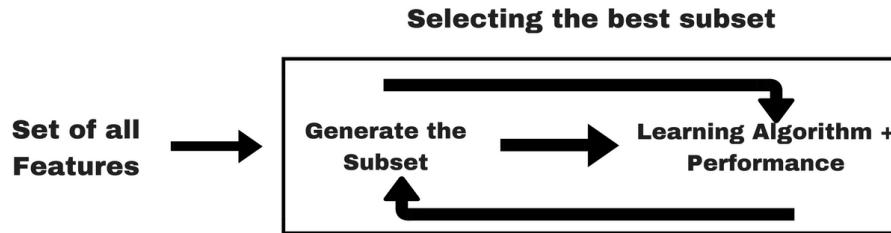
- **Wrapper methods (envoltura):** Se selecciona un subconjunto de características y se entrena un modelo con ellas. Basándose en las inferencias que se extraen del modelo anterior, se decide agregar o eliminar características de su subconjunto. Se trata esencialmente de un problema de búsqueda. Si hay muchas variables o el conjunto de datos es muy grande, estos métodos suelen ser muy costosos desde el punto de vista computacional. *Recursive Feature Elimination (RFE)* y *Sequential Feature Selection (SFS)* son ejemplos de estos métodos.



4.17. Figura: *Wrapper methods* para selección de variables

■ **Embedded methods (integrados)**: Combinan las cualidades de los métodos de filtrado y envoltura. Se implementan mediante algoritmos que tienen sus propios métodos de selección de variables integrados. Algunos de los ejemplos más populares de estos métodos son la regresión Lasso y Ridge, que tienen funciones de penalización incorporadas para reducir el sobreajuste (*overfitting*).

- La regresión Lasso realiza la regularización L1 que agrega una penalización equivalente al valor absoluto de la magnitud de los coeficientes.
- La regresión de Ridge realiza la regularización L2 que agrega una penalización equivalente al cuadrado de la magnitud de los coeficientes.



4.18. Figura: *Embedded methods* para selección de variables

Tras haber analizado las tres técnicas anteriores de selección de variables, se ha optado por implementar 5 métodos y, tras el entrenamiento posterior de los modelos predictivos, utilizar la selección de variables que mejore los resultados ofrecidos.

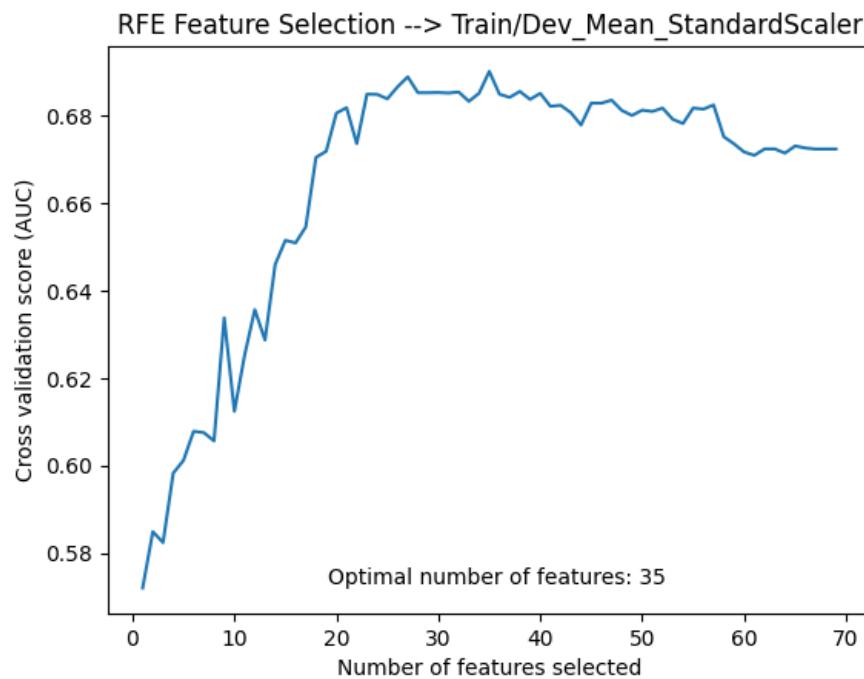
Para comenzar la selección de variables se ha utilizado un filtro que elimina todas aquellas variables que tengan un umbral de varianza de 0. Es decir, se van a eliminar aquellas variables que tienen el mismo valor en todos los ejemplos, ya que no van a aportar información en el aprendizaje al no existir diferentes valores. En Scikit-learn se ha utilizado el objeto *VarianceThreshold* con *threshold=0*. Tras este filtro se han eliminado 28 variables del conjunto de entrenamiento que tenían una varianza de 0: *aldosterona, ictus, menopausia, fibrilacion_palpitacion, otras_valvulopatias, insuficiencia_cardiaca, otras_arritmias_psicogena ritmo bigeminal, sindrome_metabolico, valvulopatia_mitral_congenita, etiologia_cardiologica, hba1c, diabetes_tipo2, hypertension_arterial, bloqueos_rama, miocardiopatia, bradicardia, arteriopatia_periferica, insuficiencia_renal, flutter, drogodependencia, fumador, diabetes.tipo1, sinus.alcoronaria, posoperatoria, alcohol, contracciones_prematuras ectopia_extrasistolica, dislipidemia, hipercolesterolemia*.

A continuación, se han aplicado los siguientes 5 métodos de selección de variables:

■ **Recursive Feature Elimination (RFE):**

Es un algoritmo de selección de variables de tipo *wrapper* que tiene como objetivo encontrar el subconjunto de variables con mejor rendimiento. Usa un algoritmo de *machine learning* como núcleo para la selección de variables. En este caso hemos utilizado *SVC* con *kernel linear* tras varias pruebas. RFE busca un subconjunto de variables empezando por todas las variables del conjunto de datos inicial y reduciendo su dimensión. Para ello, entrena un modelo en cada iteración usando el conjunto de variables correspondiente, clasifica las variables según su importancia y descarta las menos importantes. En Python y con la librería Scikit-learn, usando *RFEcv* se puede conseguir el número óptimo de variables mediante *Cross Validation* (se optimiza la métrica AUC en las evaluaciones).

Como ejemplo, vamos a aplicar RFE sobre el conjunto de datos de entrenamiento. Concretamente, sobre el conjunto con imputación de *missing values* mediante la media aritmética y estandarización mediante *StandardScaler* (ver Figura 4.19).



4.19. Figura: Métrica AUC respecto al número de variables seleccionadas en el dataset Train/-Dev_Mean_StandardScaler

El gráfico de la Figura 4.19 muestra el número óptimo de variables para maximizar la métrica AUC al entrenar un modelo *SVC* con *kernel linear*. Con 35 variables se obtiene un AUC por encima del 0.70. Las variables seleccionadas son las siguientes: *fecha, potasio, calcio, colesterol, ntprobnp, troponina_tni, leucocitos, vsg, fevi, diametro_ai, area_ai, numero_dias_desde_ingreso_hasta_evento, edad, electrocardiograma, cardioversion, ablacion, ansiedad, demencia, sahos, hipertiroidismo, cardiopatia_isquemica, epoch, genero, pensionista, residenciado, a02bc, n05a, c01, c01b, c02, c04, c07, c09, c10, polimedication.*

- ***Lasso_SelectFromModel:***

Se ha usado un estimador disperso con una regularización L1 (Lasso), donde se reducen los coeficientes de las variables menos importantes a 0 y, por lo tanto, se van a eliminar. Se ha usado como algoritmo la regresión logística (*LogisticRegression*). Para seleccionar las variables con un coeficiente diferente a 0 se ha utilizado *SelectFromModel* de Scikit-Learn.

Como ejemplo, vamos a aplicar este método de selección de variables sobre el conjunto de datos de entrenamiento con imputación de *missing values* mediante la media aritmética y estandarización mediante *StandardScaler*. En la Figura 4.20 se muestran los coeficientes de importancia asignados a las variables y se seleccionarán solamente aquellas que tengan un coeficiente mayor a 0 (un total de 46 variables): *fecha, urea, creatinina, glucosa, potasio, calcio, hdl, no_hdl, ldl, colesterol, ntprobnp, bnp, troponina_tni, troponina_tnt, leucocitos, pct, sodio, vsg, t3l, fevi, diametro_ai, area_ai, numero_dias_desde_ingreso_hasta_evento, numero_dias_ingresado, edad, peso, tipo_fa, ablacion, sahos, cardiopatia_isquemica, genero, pensionista, residenciado, imc, b01a, a02bc, c03, a10, n05a, c01, c01b, c04, c07, c09, c10, polimedication.*

- ***Lasso_SFS:***

Primero se ha aplicado una regularización L1 (Lasso) para reducir los coeficientes de las variables menos importantes a 0, de la misma manera que en el método anterior. Después, para seleccionar las variables más importantes, se ha usado Forward-SFS (*Forward Sequential Feature Selection*), el cual consiste en un procedimiento *greedy* que en cada iteración busca la mejor variable y la añade al conjunto de variables seleccionadas. En cada iteración se busca la variable que maximiza el *score* de *Cross Validation* al entrenar un estimador con cada variable (regresión logística con regularización Lasso). Se eligen la mitad de las variables iniciales.

Como ejemplo, al aplicar este método de selección de variables sobre el conjunto de datos de entrenamiento con imputación de *missing values* mediante la media aritmética y estandarización mediante *StandardScaler*, se han seleccionado un total de 34 variables: *urea*, *hdl*, *colesterol*, *bnp*, *leucocitos*, *pct*, *tsh*, *t4l*, *sodio*, *t3l*, *diametro_ai*, *area_ai*, *numero_dias_desde_ingreso_hasta_evento*, *numero_dias_ingresado*, *tipo_fa*, *ecocardiograma*, *ecocardiograma_contraste*, *cardioversion*, *marcapasos_dai*, *depresion*, *ansiedad*, *demencia*, *sahos*, *hipertiroidismo*, *bloqueo_auriculoventricular*, *valvula_mitral_reumaticas*, *pensionista*, *imc*, *n02ba01*, *g03a*, *a10*, *c01b*, *c02*, *c04*.

■ *Trees_SelectFromModel*:

Se ha usado un estimador basado en árboles de decisión para calcular la importancia de las variables en base a su impureza (coeficiente de Gini) y, de esta manera, descartar las variables irrelevantes. Como algoritmo basado en árboles de decisión se ha usado *ExtraTreesClassifier* de Scikit-learn con 50 árboles, y para la selección de variables *SelectFromModel*, donde el umbral para la selección es la media de las importancias de las variables (*Gini impurity*).

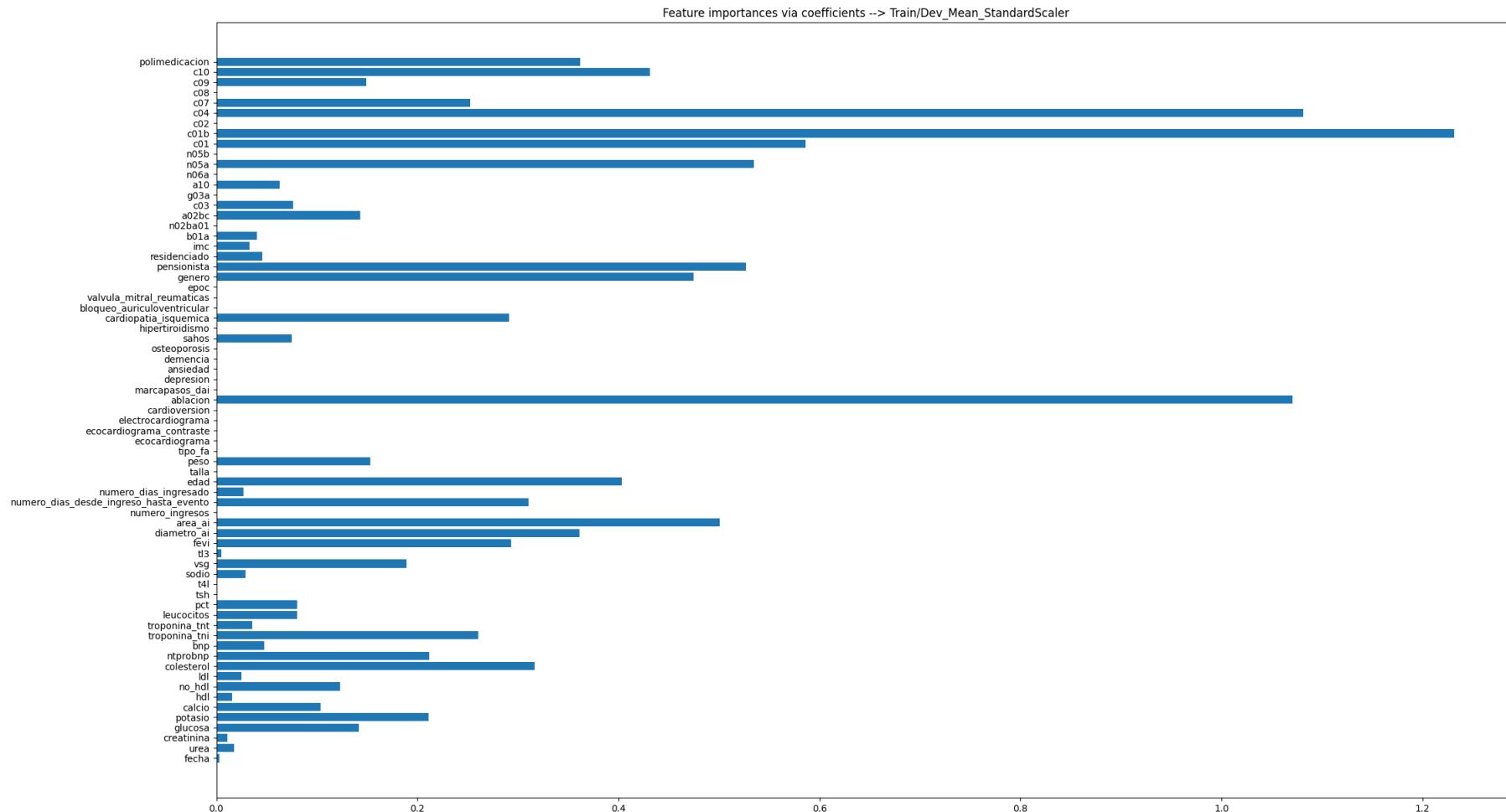
Como ejemplo, se va a aplicar este método de selección de variables sobre el conjunto de datos de entrenamiento con imputación de *missing values* mediante la media aritmética y estandarización mediante *StandardScaler*. En la Figura 4.21 se muestran los coeficientes de impureza de Gini asignados a las variables y se seleccionarán solamente aquellas que superen la media de la impureza (un total de 38 variables): *fecha*, *urea*, *creatinina*, *glucosa*, *potasio*, *calcio*, *hdl*, *no_hdl*, *ldl*, *colesterol*, *ntprobnp*, *tsh*, *t4l*, *sodio*, *vsg*, *t3l*, *fevi*, *diametro_ai*, *area_ai*, *numero_ingeridos*, *numero_dias_desde_ingreso_hasta_evento*, *numero_dias_ingresado*, *edad*, *talla*, *ecocardiograma*, *electrocardiograma*, *ablacion*, *genero*, *pensionista*, *a02bc*, *c03*, *n06a*, *n05b*, *c01b*, *c07*, *c08*, *c09*, *c10*.

■ *SelectKBest*:

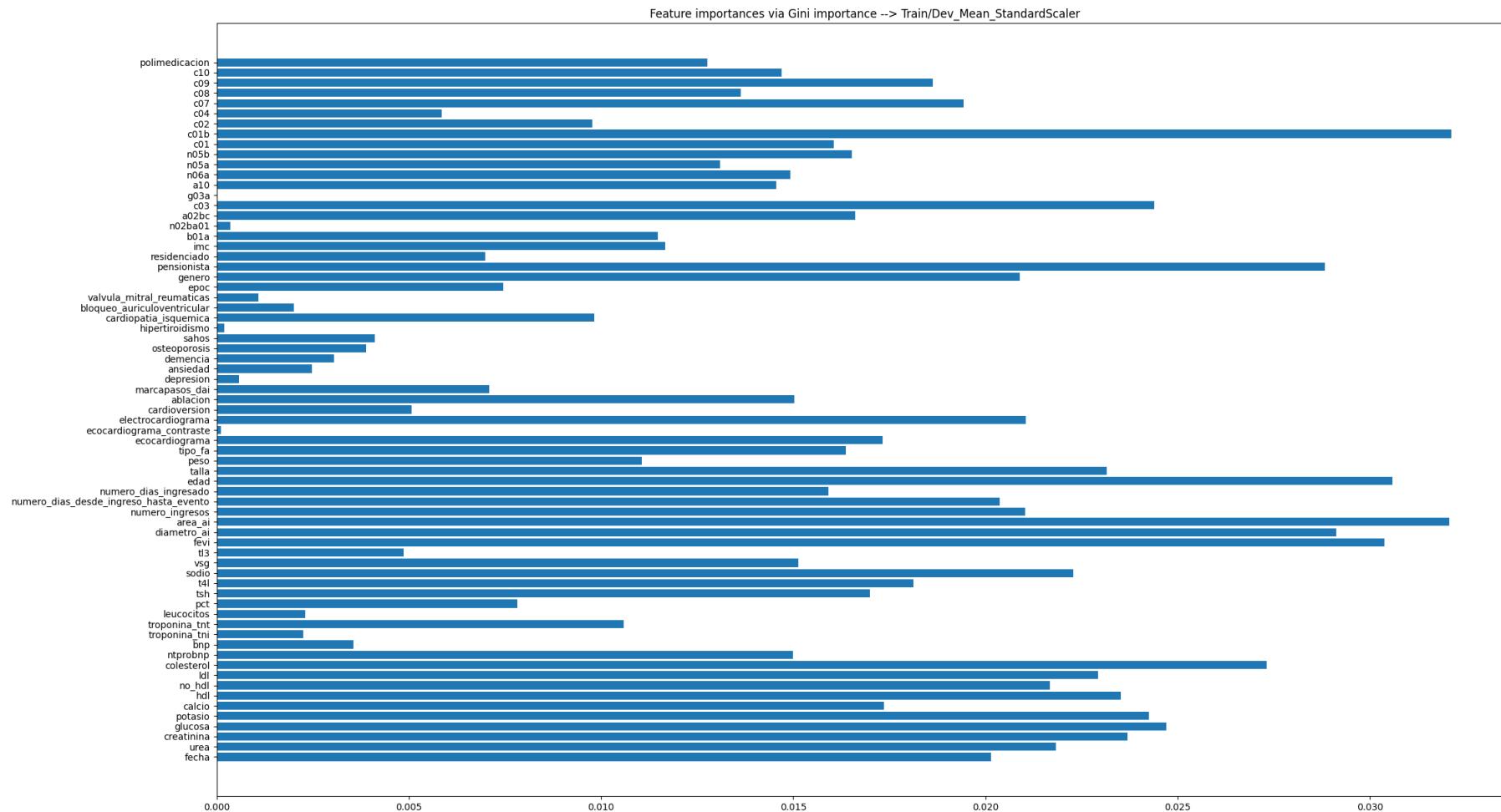
Se trata de una selección de variables univariada basada en test estadísticos. Se han calculado las estadísticas chi-cuadrado (χ^2) de las variables y se seleccionarán las variables con mayor valor en la prueba χ^2 . Tras probar diferentes números de variables a mantener, se seleccionarán las 25 variables con mayor valor χ^2 .

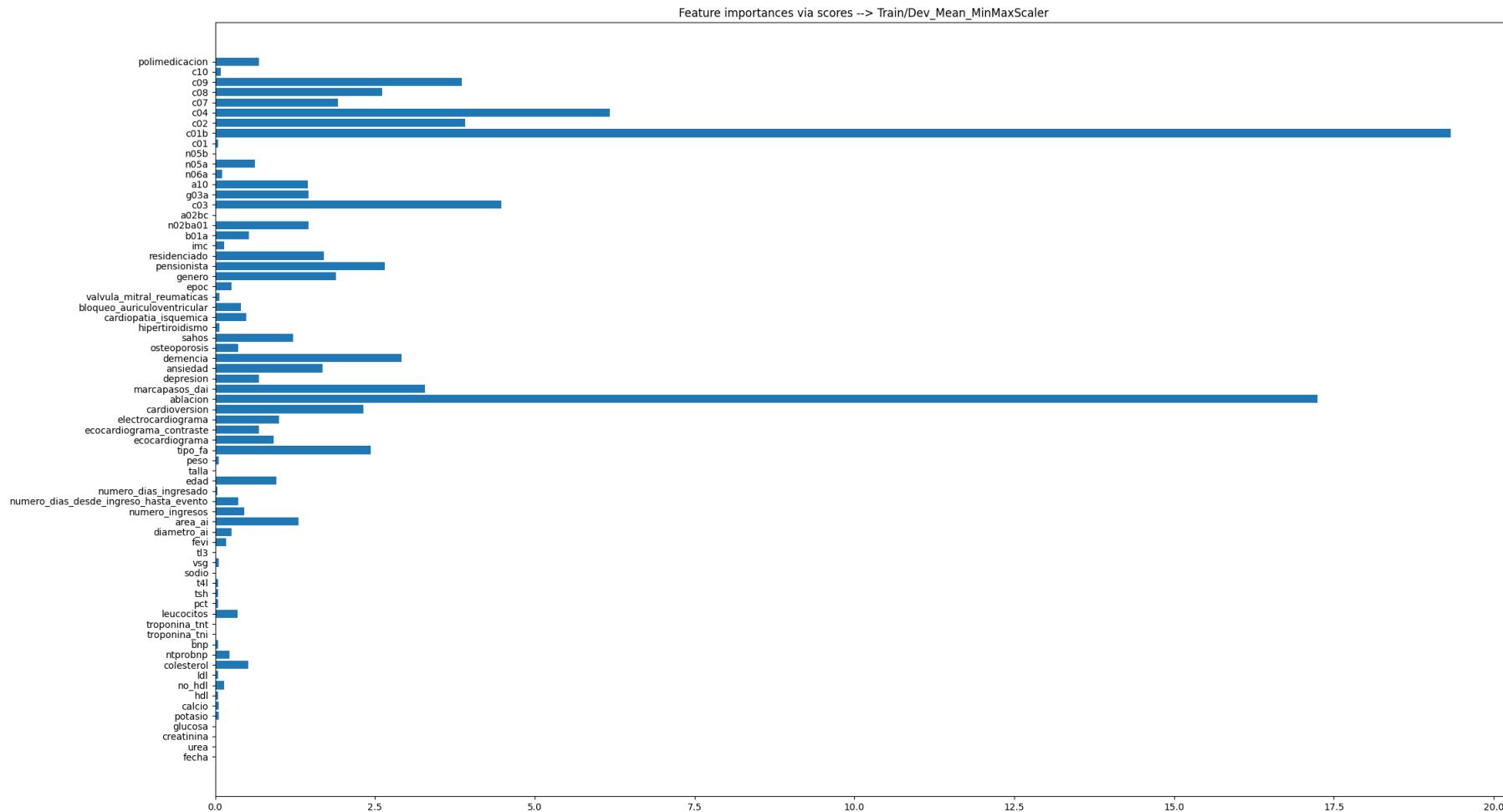
Como ejemplo, vamos a aplicar este método de selección de variables sobre el conjunto de datos de entrenamiento con imputación de *missing values* mediante la media aritmética y estandarización mediante *MinMaxScaler*, ya que todos los valores numéricos tienen que ser no negativos para la prueba χ^2 . En la Figura 4.22 se muestra el valor tras la prueba χ^2 de cada una de las variables. Las 25 variables seleccionadas con mayor valor son las siguientes: *area_ai*, *edad*, *tipo_fa*, *ecocardiograma*, *electrocardiograma*, *cardioversion*, *ablacion*, *marcapasos_dai*, *ansiedad*, *demencia*, *sahos*, *genero*, *pensionista*, *residenciado*, *n02ba01*, *c03*, *g03a*, *a10*, *c01b*, *c02*, *c04*, *c07*, *c08*, *c09*, *polimedicion*.

Tras aplicar estos cinco métodos de selección de variables, por cada conjunto de entrenamiento que teníamos, creamos cinco. Después del entrenamiento de modelos con diferentes algoritmos de ML y su respectiva evaluación, se decidirá qué método de selección de variables es el que mejores resultados da.



4.20. Figura: Coeficientes de regularización L1 (Lasso) en las variables del *dataset* Train/Dev_Mean_StandardScaler

4.21. Figura: Coeficientes de impureza de Gini en las variables del *dataset* Train/Dev_Mean_StandardScaler



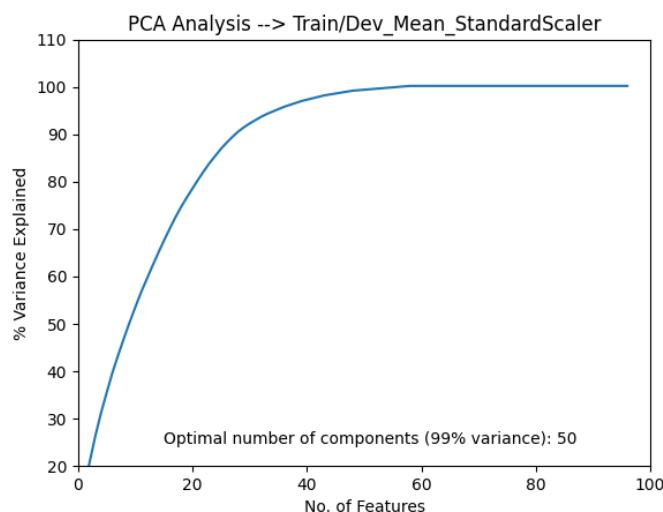
4.22. Figura: Coeficientes de la prueba χ^2 en las variables del *dataset* Train/Dev_Mean_MinMaxScaler

4.3.2.3.2 Dimension reduction (reducción de dimensiones)

En los métodos de reducción de la dimensionalidad del espacio de caracterización de las instancias, no se seleccionan y eliminan algunas de las variables, sino que se transforma el espacio vectorial a un espacio de dimensiones reducidas.

Uno de los métodos de reducción de dimensionalidad lineal más conocidos, y el que estudiaremos en este proyecto, es PCA (*Principal Component Analysis*). Es un procedimiento estadístico que transforma ortogonalmente mediante transformaciones lineales las n dimensiones numéricas originales de un conjunto de datos en un nuevo conjunto de m dimensiones llamadas componentes principales, donde $n > m$. Como resultado de la transformación, el primer componente principal tiene la mayor varianza posible. Cada componente principal subsiguiente tiene la mayor varianza posible bajo la restricción de que es orthogonal a los componentes principales precedentes, es decir, no está correlacionado con ellos. Mantener solo los primeros $m < n$ componentes principales reduce la dimensionalidad de los datos, al tiempo que conserva la mayor parte de la información de los datos, es decir, la varianza en los datos. Tras aplicar PCA, las nuevas variables que caracterizan el conjunto de datos no son las establecidas en un principio, por lo que se pierde la capacidad de interpretación.

A la hora de aplicar PCA a un conjunto de datos de entrenamiento, hemos decidido que se van a conservar todos los componentes principales (variables) con los que se mantenga el 99 % o más de la varianza de los datos. Como ejemplo, en la Figura 4.23 se muestra el porcentaje de varianza explicada a medida que se va reduciendo la cantidad de componentes principales seleccionados, en el conjunto de datos de entrenamiento con imputación de *missing values* mediante la media aritmética y estandarización mediante *StandardScaler*. En este caso, 50 componentes principales mantienen el 99 % de varianza en los datos, por lo que el conjunto de datos estará caracterizado por esas 50 nuevas variables. Como consecuencia, hemos reducido la dimensionalidad del conjunto de datos de 100 a 50 manteniendo el 99 % de varianza en los datos.



4.23. Figura: Porcentaje de varianza explicada dependiendo del número de componentes principales seleccionados

Tras el entrenamiento de modelos y su evaluación, se comprobará si un conjunto de datos con la dimensionalidad reducida mediante PCA mejora los resultados obtenidos.

4.3.2.4. Importancia/Correlación de las variables

De la misma manera que en la sección anterior se ha visto la selección de variables, también nos interesa saber la importancia de cada variable a la hora de hacer las predicciones en un modelo entrenado.

De esta manera, se podrá observar el valor de qué variable tiene más influencia en las predicciones. En esta sección hemos utilizado diferentes algoritmos para analizar su influencia, y se analizará sobre el conjunto de entrenamiento y evaluación, como se muestra en la Figura 4.15.

4.3.2.4.1 *Feature importance* (importancia de las variables)

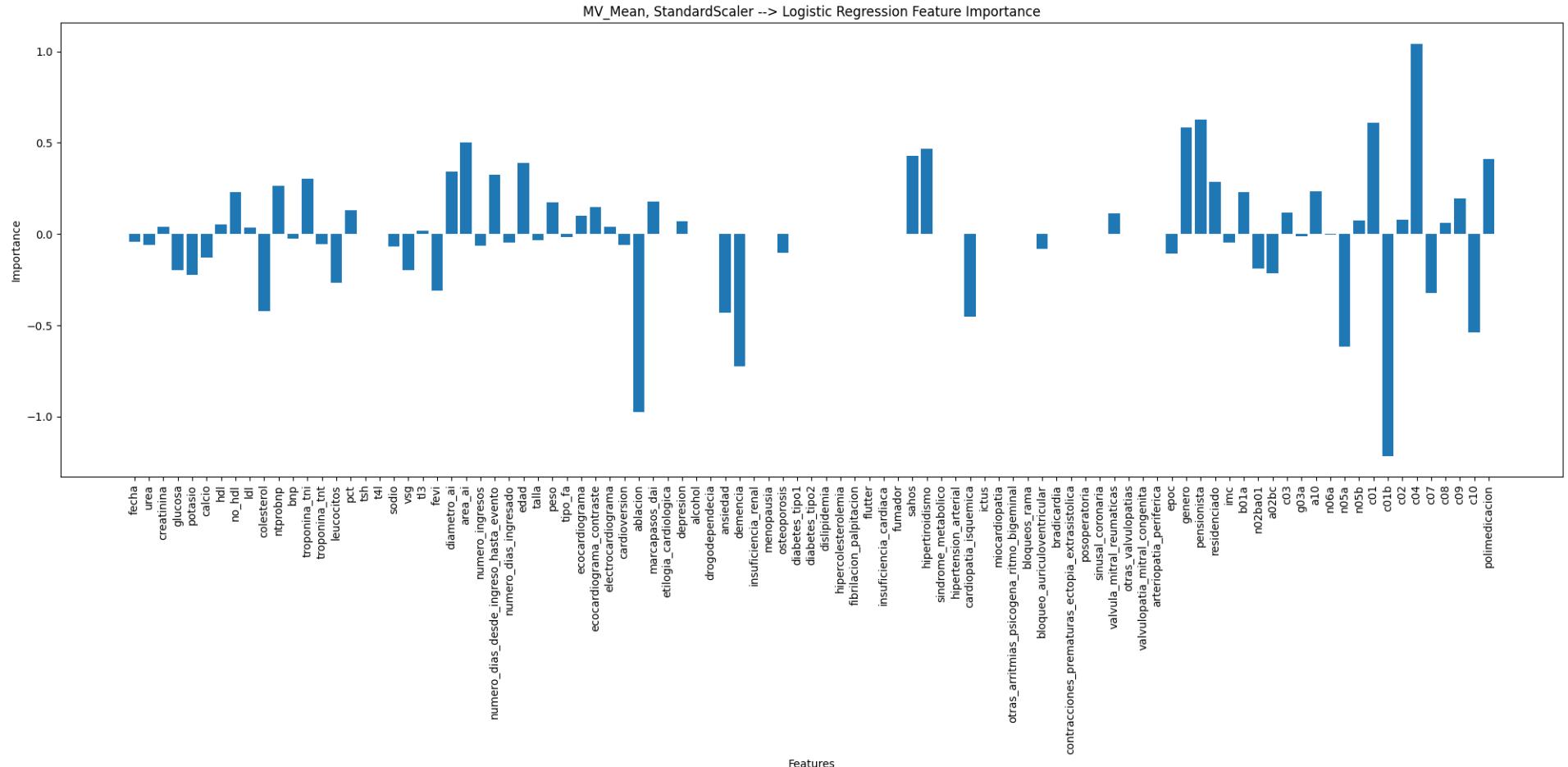
Los algoritmos de aprendizaje automático lineal se ajustan a un modelo donde la predicción es la suma ponderada de los valores de entrada. Todos estos algoritmos encuentran un conjunto de coeficientes para usar en la suma ponderada y hacer una predicción. Estos coeficientes se pueden utilizar directamente como una puntuación de importancia de las variables. Se ha analizado la importancia de las variables con los siguientes algoritmos:

- ***LogisticRegression***: se entrena un modelo mediante este algoritmo y se asigna a las variables un coeficiente de la función de decisión. La puntuación positiva de una variable indica que predice la clase 1, mientras que la puntuación negativa indica que predice la clase 0.
- ***CART***: se entrena un modelo mediante el algoritmo *DecisionTreeClassifier*, el cual asigna a cada variable una puntuación de importancia (*Gini importance*).
- ***RandomForest***: se entrena un modelo mediante el algoritmo *RandomForestClassifier*, el cual asigna a cada variable una puntuación de importancia (*Gini importance*).
- ***XGBoost***: se entrena un modelo mediante el algoritmo *XGBClassifier*, el cual asigna a cada variable una puntuación de importancia (*Gini importance*).

Otra técnica para calcular la importancia relativa de las variables en varias iteraciones es *Permutation Feature Importance*. Primero, un modelo se ajusta al conjunto de datos y se usa para hacer predicciones. Las predicciones se hacen para cada variable. Después, todo este proceso se repite varias veces, en nuestro caso faremos 10 iteraciones. El resultado es una puntuación de importancia media para cada variable de entrada (y la distribución de puntuaciones dadas las repeticiones). Se requiere especificar una métrica como base de la puntuación de importancia, por ejemplo, *accuracy* o *AUC*.

- ***Permutation_KNN***: se ha usado el algoritmo *KNeighborsClassifier*.
- ***Permutation_RandomForest***: se ha usado el algoritmo *RandomForestClassifier*.

A continuación, en la Figura 4.24 se muestran las importancias de las variables tras entrenar un modelo de regresión logística con el conjunto de entrenamiento con imputación de *missing values* mediante la media aritmética y estandarización mediante *StandardScaler*. Las variables con coeficiente de importancia negativa como *abacion* tienen mayor influencia en la predicción de la clase 0 (no recurrencia de FA), mientras que las variables con coeficientes positivos como *pensionista* tienen mayor influencia en la predicción de la clase 1 (recurrencia de FA).

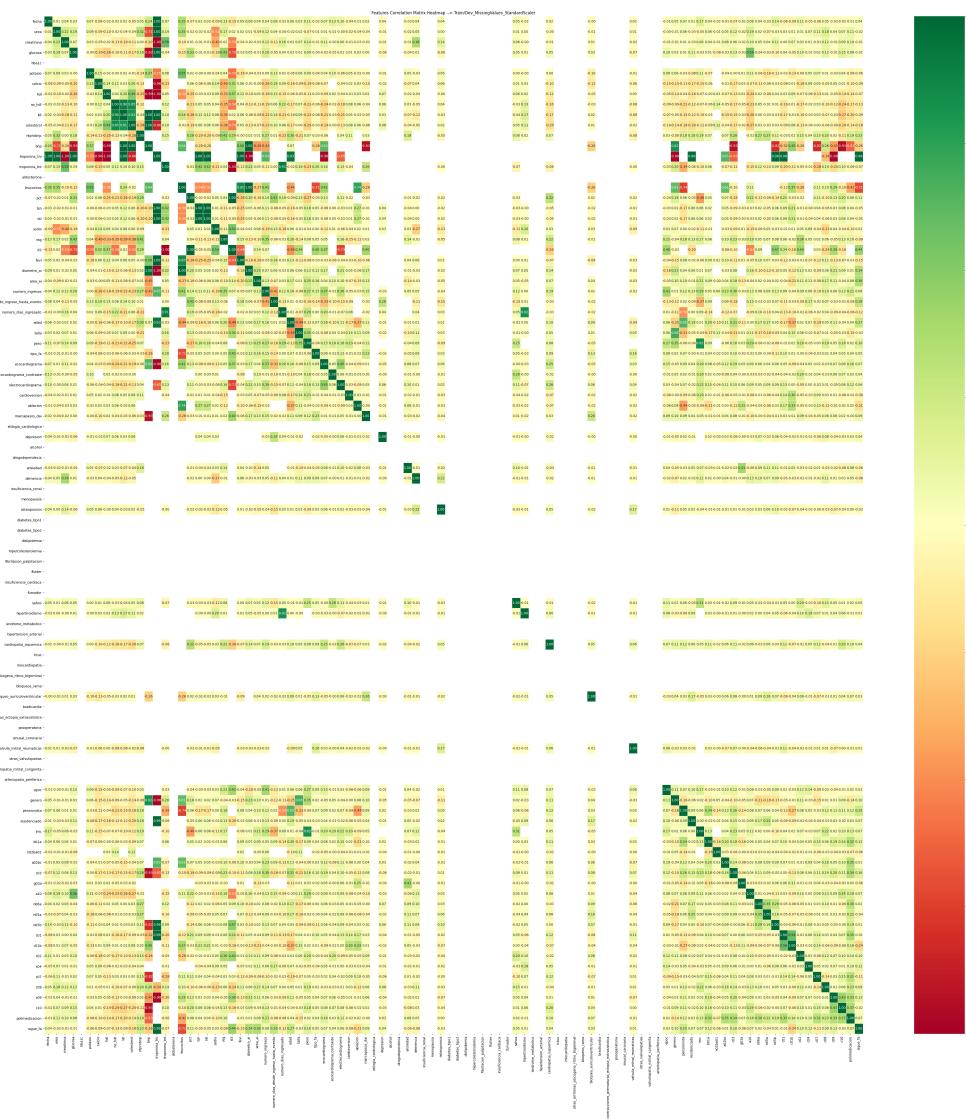


4.24. Figura: Importancia de las variables en un modelo de *LogisticRegression* sobre el dataset Train/Dev_Mean_StandardScaler

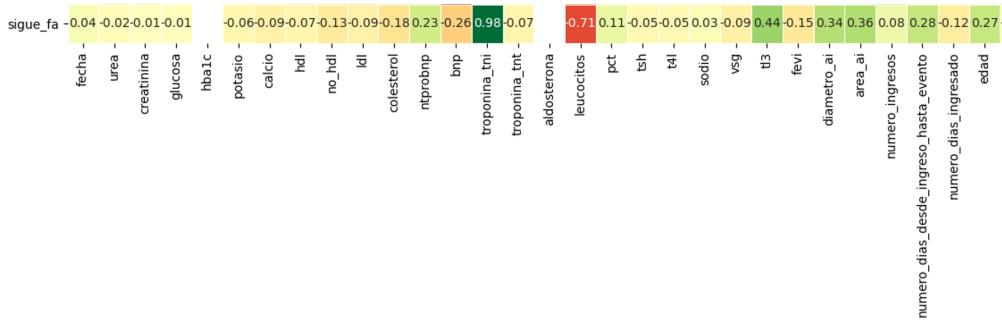
4.3.2.4.2 Correlación de las variables (*heatmaps*)

La correlación entre pares de variables se puede visualizar de una forma sencilla mediante *heatmaps*. Los *heatmaps* son mapas de calor con la estructura de una matriz en la que cada celda muestra la correlación entre dos variables del conjunto de datos. La correlación es positiva cuando el aumento de una variable aumenta la variable destino, o negativa cuando el aumento de una variable disminuye el valor de la variable de destino, o viceversa. El mapa de calor facilita la identificación de las variables que están más relacionadas con la variable se salida (recurrencia o no de FA).

En la Figura 4.25 se muestra el *heatmap* de correlación de variables del conjunto de datos de entrenamiento con *missing values* y estandarización mediante *StandardScaler*. Al haber tantas variables se ha ampliado la correlación de la clase (*sigue_fa*) con algunas de las variables en la Figura 4.26. En esta imagen se puede ver que variables como *troponina_tni*, *tl3*, *diametro_ai*, *area_ai* o *edad* están más correlacionadas con la clase (*sigue_fa*) positiva (recurrencia de FA).



4.25. Figura: *Heatmap* de correlación del *dataset* Train/Dev_MissingValues_StandardScaler



4.26. Figura: Correlación de algunas variables con la clase *sigue_fa* en el dataset Train/-Dev_MissingValues_StandardScaler

4.3.2.5. Entrenamiento de modelos

En las secciones anteriores se ha creado el conjunto de datos y se le han aplicado técnicas de preprocesado de datos y selección de variables o reducción de dimensiones. Por cada combinación de los anteriores métodos se han creado conjuntos diferentes para el entrenamiento de modelos de aprendizaje automático supervisado. Tras la evaluación de los modelos entrenados, se elegirá la combinación de métodos de preprocesado óptima con la que se consigan mejores resultados.

Resumiendo, primero se han conseguido 5 conjuntos de entrenamiento dependiendo del método de imputación de *missing values*:

- Sin imputación.
- Imputación de la media aritmética.
- Imputación de la mediana.
- Imputación de un valor predicho por un modelo.
- Imputación de un valor diferente a todos los *missing values*.

A continuación, por cada conjunto de datos de los anteriores se han aplicado (o no) 5 técnicas de estandarización/normalización:

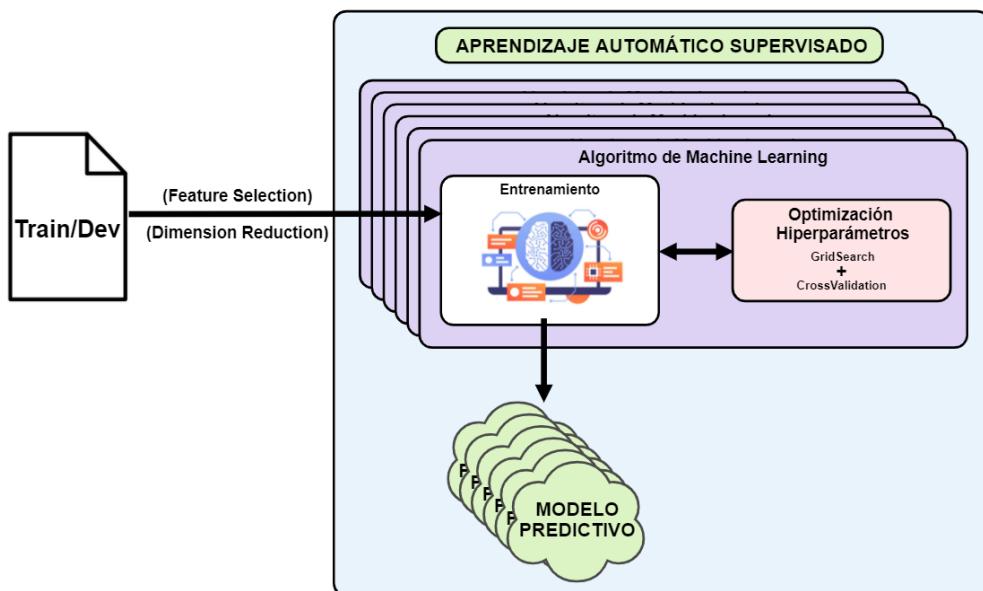
- Sin estandarización/normalización.
- *StandardScaler*.
- *MinMaxScaler*.
- *MaxAbsScaler*.
- *QuantileTransformer*.
- *Normalizer*.

Finalmente, a cada conjunto de los preprocesados anteriormente se le han aplicado (o no) 6 métodos de selección de variables o reducción de dimensionalidad:

- Sin selección de variables ni reducción de dimensionalidad.
- *RFE*.
- *Lasso_SelectFromModel*.
- *Lasso_SFS*.
- *Trees_SelectFromModel*.
- *SelectKBest*.
- *PCA*.

Con todas estas variantes del conjunto de datos de entrenamiento se han entrenado modelos predictivos de recurrencia de FA. Para ello, se ha probado con diferentes algoritmos de aprendizaje automático supervisado. Dependiendo del algoritmo, también se ha llevado a cabo una optimización de sus hiperparámetros más importantes. En la Figura 4.27 puede verse el diseño de esta fase de aprendizaje automático.

Para la optimización de hiperparámetros de los algoritmos se ha utilizado la técnica *Grid Search* junto con *Cross Validation*. Se ha implementado en Python mediante el objeto *GridSearchCV* ofrecido por Scikit-learn. *Grid Search* es una búsqueda exhaustiva de valores de parámetros especificados para un estimador. Es necesario determinar una lista con los parámetros que quieren optimizarse y sus valores a probar. Se entrena un modelo por cada combinación de valores a probar en estos parámetros, y a través de una evaluación mediante el esquema *Cross Validation*, se consiguen los valores que mejores resultados ofrecen.



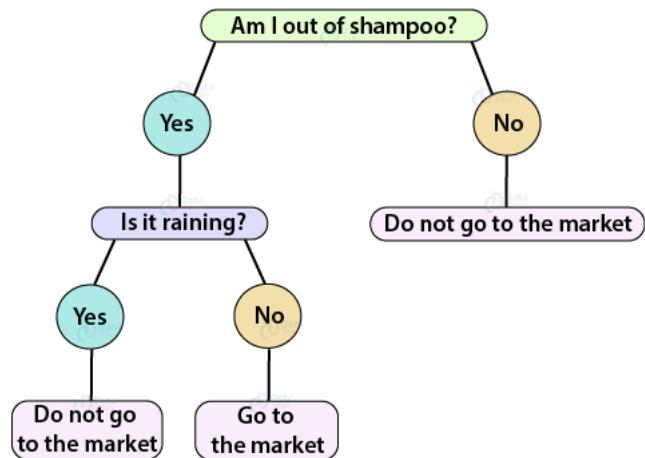
4.27. Figura: Diseño del entrenamiento de modelos

A continuación se da una lista de los algoritmos de aprendizaje automático supervisado que se han implementado (y sus hiperparámetros a optimizar, dependiendo del caso):

- ***DecisionTree*:**

Los árboles de decisión (*Decision Trees*) son un método de aprendizaje supervisado no paramétrico que se utiliza para clasificación y regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo mediante el aprendizaje de reglas de decisión simples inferidas de las características de los datos. Un árbol puede verse como una aproximación constante por partes, donde el valor de cada variable va llevando a una predicción u otra (ver Figura 4.28). En Python se ha utilizado el objeto *DecisionTreeClassifier* de Scikit-learn.

Decision Trees Example



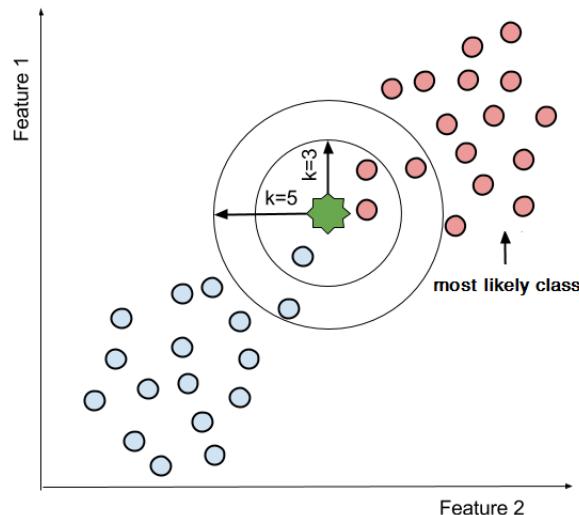
4.28. Figura: Ejemplo del algoritmo *DecisionTree*

- ***KNN (k-Nearest Neighbors):***

Es un método de aprendizaje supervisado (o no supervisado) no paramétrico y se basa en encontrar un número predefinido de instancias de entrenamiento más cercanas en distancia al nuevo punto, las cuales se denominan vecinos, y predecir la etiqueta a partir de ellos. El número de vecinos es normalmente una constante definida por el usuario. Al aplicar KNN en tareas de clasificación, las etiquetas de salida se calculan a partir de un voto de mayoría simple entre los vecinos más cercanos de cada punto, es decir, a un punto de consulta se le asigna la clase que tiene más representantes dentro de los vecinos más cercanos del punto (ver Figura 4.29). En Python se ha utilizado el objeto *KNeighborsClassifier* de Scikit-learn.

Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

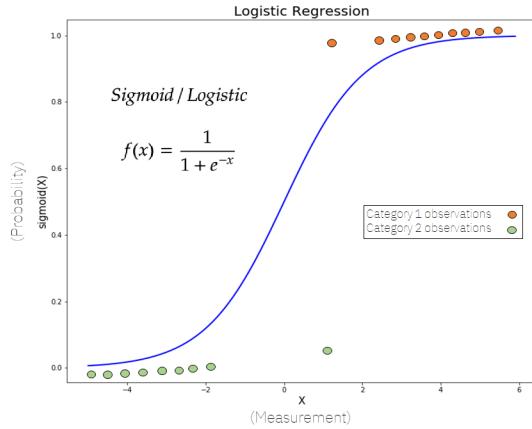
- *n_neighbors*: Número de vecinos a usar. Se ha probado con valores en el rango [1, 30].
- *p*: Distancia entre dos puntos (instancias). Se ha probado con la distancia Manhattan (1) y Euclídea (2).



4.29. Figura: Ejemplo del algoritmo *KNN*

■ ***LogisticRegression:***

La regresión logística (*Logistic Regression*), a pesar de su nombre, es un modelo lineal para clasificación en lugar de regresión. En este modelo, las probabilidades que describen los posibles resultados de un solo ensayo se modelan utilizando una función logística (ver Figura 4.30). En Python se ha utilizado el objeto *LogisticRegression* de Scikit-learn.



4.30. Figura: Ejemplo del algoritmo *LogisticRegression*

■ ***NaïveBayes:***

Los métodos *Naïve Bayes* son un conjunto de algoritmos de aprendizaje supervisados basados en la aplicación del teorema de Bayes. Se asume la independencia condicional entre cada par de características dado el valor de la variable de clase. El teorema de Bayes establece la siguiente relación, dada la variable de clase y y el vector de características dependiente x :

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)}$$

Asumiendo la independencia condicional entre variables

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

para cualquier i , esta relación se simplifica a

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)}$$

Por lo que se puede usar la siguiente regla de clasificación:

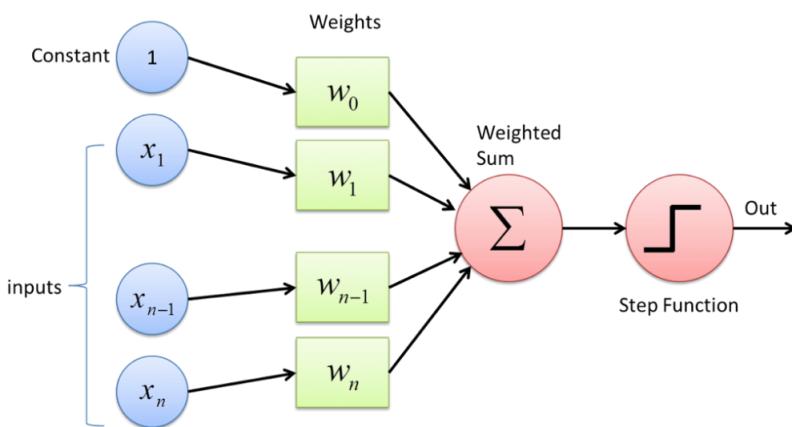
$$\begin{aligned} P(y | x_1, \dots, x_n) &\propto P(y) \prod_{i=1}^n P(x_i | y) \\ &\Downarrow \\ \hat{y} &= \arg \max_y P(y) \prod_{i=1}^n P(x_i | y), \end{aligned}$$

En Python se ha utilizado el objeto *GaussianNB* de Scikit-learn, donde se asume que la probabilidad de las características es Gaussiana:

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

■ **Perceptron:**

El perceptrón es la red neuronal más básica que existe de aprendizaje supervisado. El funcionamiento del perceptrón es muy sencillo, simplemente lee los valores de entrada, suma todos las entradas de acuerdo a unos pesos y el resultado lo introduce en una función de activación que genera el resultado final (ver Figura 4.31). El entrenamiento del perceptrón no es más que determinar los pesos sinápticos y el umbral con el que la entrada se ajuste mejor a la salida. Para la determinación de estas variables, se sigue un proceso adaptativo. El proceso comienza con valores aleatorios y se van modificando estos valores según la diferencia entre los valores deseados y los calculados por la red. En Python se ha utilizado el objeto *Perceptron* de Scikit-learn.



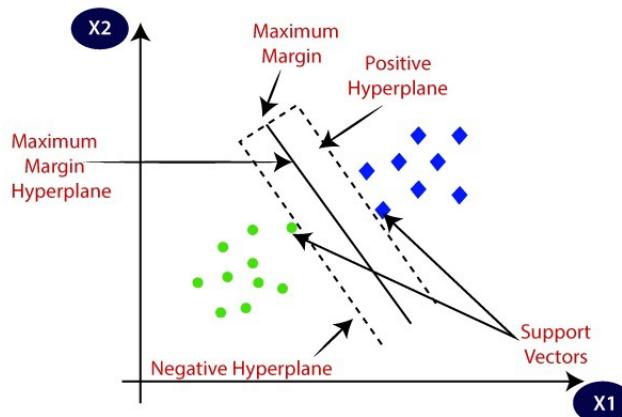
4.31. Figura: Ejemplo del algoritmo *Perceptron*

■ **SVM (Support Vector Machines):**

Las máquinas de vectores de soporte (SVM) son un conjunto de algoritmos de aprendizaje supervisado. Una SVM es un modelo que representa los puntos de la muestra en el espacio, separando las clases en 2 espacios (en clasificación binaria) lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama *vector soporte*. Cuando las nuevas instancias se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas con una u otra clase. Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta que puede ser utilizado en problemas de clasificación o regresión (ver Figura 4.32). En Python se ha utilizado el objeto *SVC* de Scikit-learn.

Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

- *C*: Parámetro de regularización. La fuerza de la regularización es inversamente proporcional a *C*. Debe ser estrictamente positiva. La penalización es de L2 al cuadrado. Se ha probado con los siguientes valores {0.1, 1, 10, 100}.
- *kernel*: Especifica el tipo de kernel que se utilizará en el algoritmo. Se ha probado con los siguientes kernels {*rbf*, *poly*, *sigmoid*}.
- *gamma*: Coeficiente de kernel para *rbf*, *poly* y *sigmoid*. Se ha probado con los siguientes valores {1, 0.1, 0.01, 0.001}.
- *degree*: Grado de la función del kernel polinómica (*poly*). Ignorado por todos los demás kernels. Se ha probado con los siguientes valores {2, 3, 4, 5, 6, 7, 8, 9, 10}.



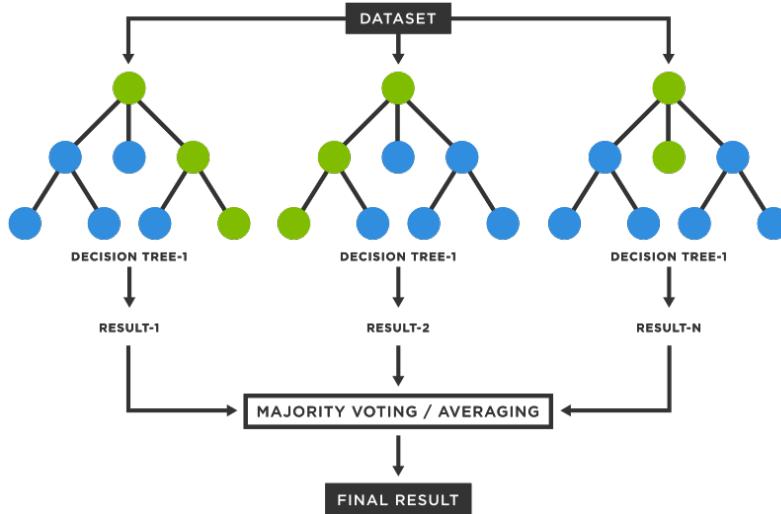
4.32. Figura: Ejemplo del algoritmo *SVM*

■ *RandomForest*:

Un *Random Forest* es un meta-estimador que se ajusta a varios clasificadores de árboles de decisión en varias submuestras del conjunto de datos y usa promedios para mejorar la precisión predictiva y controlar el sobreajuste (ver Figura 4.33). En Python se ha utilizado el objeto *RandomForestClassifier* de Scikit-learn.

Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

- *n_estimators*: Cantidad de árboles en el bosque. Se ha probado con los siguientes valores {100, 300, 500}.
- *max_depth*: Profundidad máxima del árbol. Se ha probado con los siguientes valores {5, 8, 15, 25, 30}.
- *min_samples_split*: Número mínimo de muestras necesarias para dividir un nodo interno. Se ha probado con los siguientes valores {2, 5, 10, 15, 100}.
- *min_samples_leaf*: Número mínimo de muestras necesarias para estar en un nodo hoja. Un nodo de división a cualquier profundidad solo se considerará si deja al menos *min_samples_leaf* muestras de entrenamiento en cada una de las ramas izquierda y derecha. Se ha probado con los siguientes valores {1, 2, 5, 10}.



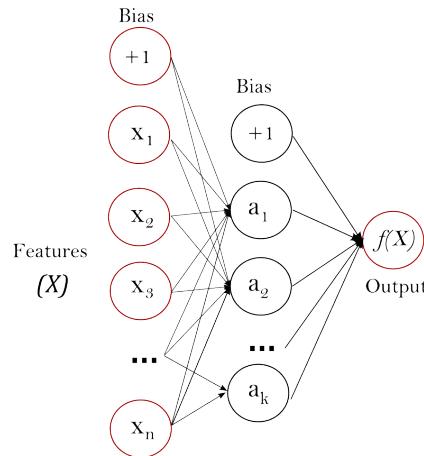
4.33. Figura: Ejemplo del algoritmo *RandomForest*

■ **MLP (Multilayer Perceptron):**

El Perceptrón Multicapa (MLP) es un algoritmo de aprendizaje supervisado que dado un conjunto de características $X = x_1, x_2, \dots, x_m$ y una variable de salida y , puede aprender un aproximador de función no lineal para clasificación o regresión. Es diferente de la regresión logística, ya que entre la capa de entrada y la de salida puede haber una o más capas no lineales, llamadas capas ocultas. La Figura 4.34 muestra un MLP de una capa oculta con salida escalar. En Python se ha utilizado el objeto *MLPClassifier* de Scikit-learn.

Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

- *hidden_layer_sizes*: El i -ésimo elemento representa el número de neuronas en la i -ésima capa oculta. Se ha probado con los siguientes valores $\{(50, 50, 50), (50, 100, 50), (100)\}$.
- *activation*: Función de activación para la capa oculta. Se ha probado con los siguientes valores $\{\tanh, \text{relu}\}$.
 - *tanh*: la función tangente hiperbólica, devuelve $f(x) = \tanh(x)$.
 - *relu*: la función de unidad lineal rectificada, devuelve $f(x) = \max(0, x)$.
- *solver*: Método para la optimización del peso. Se ha probado con los siguientes valores $\{\text{sgd}, \text{adam}\}$.
 - *sgd*: descenso de gradiente estocástico.
 - *adam*: optimizador estocástico basado en gradiente propuesto por Kingma, Diederik y Jimmy Ba.
- *alpha*: Parámetro de penalización L2 (término de regularización). Se ha probado con los siguientes valores $\{0.0001, 0.05\}$.
- *learning_rate_init*: Tasa de aprendizaje inicial utilizada. Controla la proporción de cambio al actualizar los pesos. Se ha probado con los siguientes valores $\{0.001, 0.005, 0.01\}$.
- *learning_rate*: Gestor del *learning rate* para las actualizaciones de los pesos. Solo se usa cuando el *solver* es *sgd*. Se ha probado con los siguientes valores $\{\text{constant}, \text{adaptive}\}$.
 - *constant*: es un *learning rate* constante dado por el parámetro *learning_rate_init*.
 - *adaptive*: mantiene el *learning rate* constante en *learning_rate_init* siempre que la pérdida de entrenamiento (*loss*) siga disminuyendo. Cada vez que no disminuya en dos iteraciones (*epochs*) consecutivas, la tasa de aprendizaje actual se divide por 5.



4.34. Figura: Ejemplo del algoritmo *MLP*

■ *Bagging:*

Esta técnica consiste en crear diferentes modelos usando muestras aleatorias con reemplazo, y después, combinar o ensamblar los resultados. Primero, se divide el conjunto de entrenamiento en distintos subconjuntos (*bootstrapping*), obteniendo como resultado diferentes muestras aleatorias con las siguientes características:

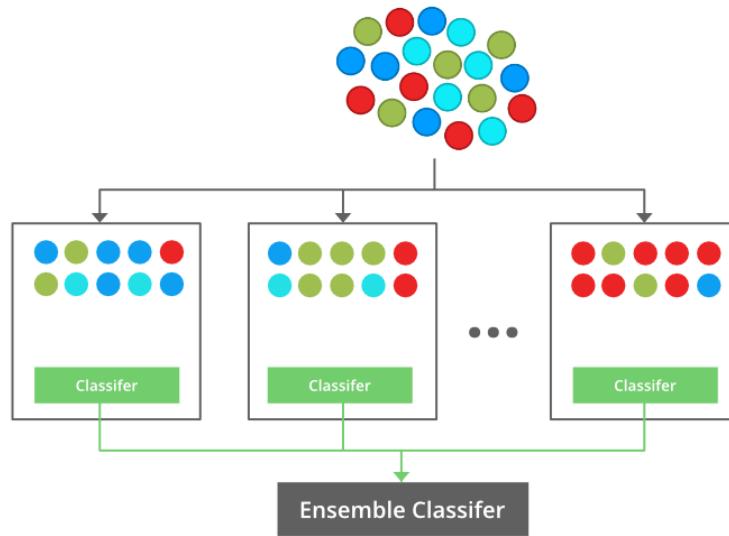
- Muestra uniforme (misma cantidad de instancias en cada conjunto).
- Muestreo con reemplazo (las instancias pueden repetirse en el mismo subconjunto).
- El tamaño de la muestra es igual al tamaño del conjunto de entrenamiento, pero no contiene todas las instancias, ya que algunas se repiten.

Después, se crea un modelo predictivo con cada subconjunto, obteniendo modelos diferentes. Finalmente, se construye o ensambla un único modelo predictivo, que es el promedio de todos los modelos (ver Figura 4.35).

Se necesita establecer un algoritmo para el entrenamiento de los modelos base. En este caso, hemos elegido *Logistic Regression*. En Python se ha utilizado el objeto *BaggingClassifier* de Scikit-learn.

Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

- *n_estimators*: Número de estimadores base. Se ha probado con los siguientes valores {10, 50, 100}.
- *max_samples*: Proporción de instancias a extraer del conjunto total para entrenar cada estimador base. Se ha probado con los siguientes valores {0.1, 0.2, 0.5, 1.0}.
- *max_features*: Proporción de características a extraer del conjunto total para entrenar cada estimador base. Se ha probado con los siguientes valores {0.1, 0.2, 0.5, 1.0}.



4.35. Figura: Ejemplo del algoritmo *Bagging*

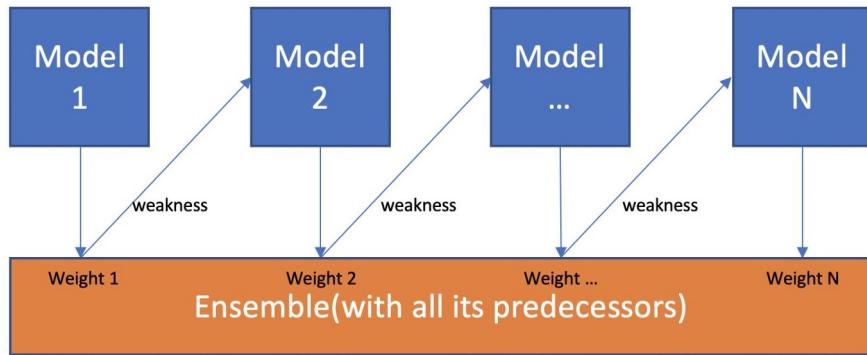
■ *AdaBoost*:

Es un algoritmo basado en *Boosting*, un método que pretende mejorar el desempeño de cualquier algoritmo de aprendizaje supervisado mediante la combinación de los resultados de varios estimadores base para obtener un clasificador final robusto. Una de las técnicas más populares de *Boosting* es el algoritmo *Boosting Adaptativo (AdaBoost)*. Este algoritmo, mediante un entrenamiento iterativo de los estimadores base, le asigna mayor importancia a los datos mal clasificados anteriormente, y de esta manera obtiene un nuevo clasificador (ver Figura 4.36). Por lo tanto, consigue adaptarse y obtener mejores resultados aumentando la precisión del algoritmo.

Al igual que con *Bagging*, se necesita establecer un algoritmo para el entrenamiento de los modelos base. También hemos elegido *Logistic Regression*. En Python se ha utilizado el objeto *AdaBoostClassifier* de Scikit-learn.

Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

- *n_estimators*: Número máximo de estimadores con los que finaliza el entrenamiento iterativo. En caso de un ajuste perfecto, el procedimiento de aprendizaje se detiene antes de tiempo. Se ha probado con los siguientes valores {10, 50, 100}.
- *learning_rate*: Peso aplicado a cada clasificador en cada iteración de entrenamiento. Una tasa de aprendizaje más alta aumenta la contribución de cada clasificador. Se ha probado con los siguientes valores {0.01, 0.1, 0.5, 1.0}.



4.36. Figura: Ejemplo del algoritmo *AdaBoost*

■ ***XGBoost (Extreme Gradient Boosting)*:**

Es un algoritmo basado en *boosting* con árboles de decisión y minimización del error mediante descenso de gradiente (*Gradient Boosting*), optimizado a través del procesamiento paralelo, poda de árboles, manejo de *missing values* y regularización para evitar sobreajustes/sesgos (*overfitting/bias*). En Python se ha utilizado el objeto *XGBClassifier*.

Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

- *learning_rate*: Tasa de aprendizaje para la actualización de los pesos. Se ha probado con los siguientes valores {0.01, 0.1}.
- *max_depth*: Profundidad máxima del árbol para los estimadores base. Se ha probado con los siguientes valores {3, 6, 10}.
- *min_child_weight*: Suma mínima de peso de instancia en un nodo hijo. Se ha probado con los siguientes valores {1, 3, 5}.
- *subsample*: Proporción de submuestra de la instancia de entrenamiento. Se ha probado con los siguientes valores {0.5, 0.7}.
- *colsample_bytree*: Proporción de submuestras de columnas al construir cada árbol. Se ha probado con los siguientes valores {0.5, 0.7}.
- *n_estimators*: Número de árboles. Equivalente al número de iteraciones de *boosting*. Se ha probado con los siguientes valores {100, 500, 1000}.

■ ***LightGBM (Light Gradient Boosted Machine)*:**

Al igual que *XGBoost*, es un algoritmo de *Gradient Boosting* que hace uso de algoritmos de aprendizaje basados en árboles de decisión y descenso de gradiente. También es insensible a *missing values*. Mientras que otros algoritmos de árboles de decisión crecen horizontalmente, el algoritmo LightGBM crece verticalmente, lo que significa que crece en forma de hojas y no por niveles. LightGBM elige la hoja con grandes pérdidas para crecer. Puede reducir más pérdidas que un algoritmo de crecimiento por niveles cuando crece la misma hoja. En Python se ha utilizado el objeto *LGBMClassifier*.

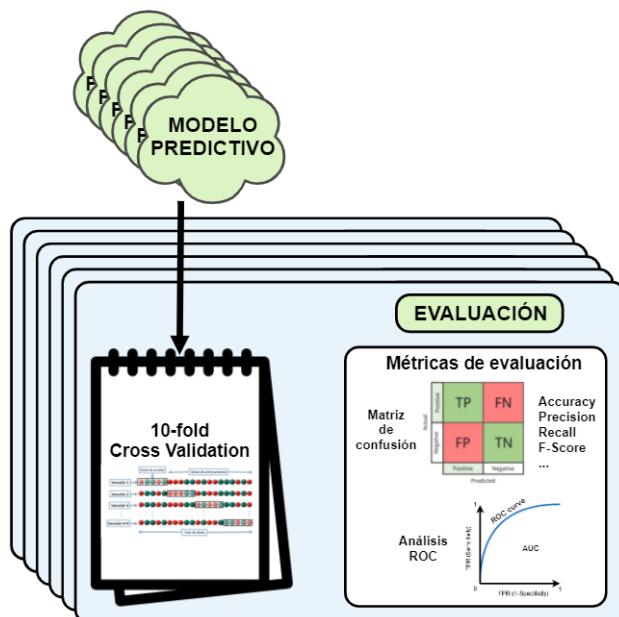
Los hiperparámetros a optimizar en este algoritmo han sido los siguientes:

- *learning_rate*: Tasa de aprendizaje para la actualización de los pesos. Se ha probado con los siguientes valores {0.01, 0.1}.
- *max_depth*: Profundidad máxima del árbol para los estimadores base. Se ha probado con los siguientes valores {3, 6, 10}.

- *num_leaves*: Máximo de hojas de árboles para los estimadores base. Se ha probado con los siguientes valores {5, 10, 50}.
- *min_child_samples*: Número mínimo de datos necesarios en un nodo hijo (hoja). Se ha probado con los siguientes valores {20, 50, 100}.
- *n_estimators*: Número de árboles. Equivalente al número de iteraciones de *boosting*. Se ha probado con los siguientes valores {100, 500, 1000}.

4.3.2.6. Evaluación de modelos

Esta fase es la encargada de evaluar el rendimiento de los modelos predictivos de la sección anterior. Por cada modelo entrenado se ha realizado una evaluación mediante el esquema *k-fold Cross Validation*. Como resultado de la evaluación se obtienen las métricas que nos permiten hacernos una idea del rendimiento ofrecido por el modelo predictivo y compararlo con otros de algoritmos diferentes. En la Figura 4.37 se muestra el diseño de esta fase.



4.37. Figura: Diseño de la evaluación de los modelos

4.3.2.6.1 Esquema de evaluación: *k-fold Cross Validation*

Cross Validation es un procedimiento de remuestreo que se utiliza para evaluar modelos de aprendizaje automático en una muestra de datos limitada (nuestro conjunto de datos Train/Dev).

El procedimiento tiene un solo parámetro llamado *k* que se refiere al número de grupos en los que se dividirá una muestra de datos Train/Dev. Por ello, el procedimiento *k-fold Cross Validation*. Cuando se elige un valor específico para *k*, se puede usar en lugar de *k* en la referencia al modelo, por ejemplo, si *k* = 10 se convierte *10-fold Cross Validation*.

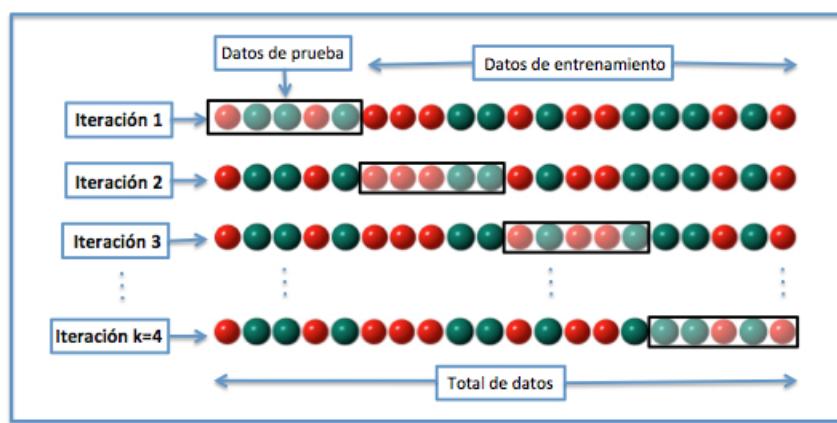
Es un método popular porque es fácil de entender y generalmente da como resultado una estimación menos sesgada o menos optimista del rendimiento del modelo que otros métodos, como una única división de entrenamiento/evaluación (*Hold-out*).

El procedimiento general de una evaluación con *Cross Validation* es el siguiente (ver Figura 4.38):

1. Mezclar el conjunto de datos de forma aleatoria.
2. Dividir el conjunto de datos en *k* grupos.

3. Para cada grupo único:
 - a) Tomar el grupo como un conjunto de datos de desarrollo/evaluación (Dev).
 - b) Tomar los grupos restantes como un conjunto de datos de entrenamiento (Train).
 - c) Entrenar un modelo con el conjunto de entrenamiento (Train) y evaluarlo en el conjunto de desarrollo/evaluación (Dev).
 - d) Conservar los resultados de las métricas de evaluación y descartar el modelo.
 4. Resumir el rendimiento del modelo a través de los resultados obtenidos en cada iteración, por ejemplo, calculando la media aritmética y la desviación estándar de las métricas.

Es importante destacar que cada ejemplo en la muestra de datos se asigna a un único grupo y permanece en ese grupo durante toda la duración del procedimiento. Esto significa que a cada muestra se le da la oportunidad de ser usada en la evaluación (conjunto Dev) una única vez y usada para entrenar el modelo (conjunto Train) $k - 1$ veces.



4.38. Figura: Ejemplo 4-fold Cross Validation

En el caso de nuestro proyecto, hemos evaluado los modelos con *10-fold Cross Validation*. El conjunto de datos al completo estaba formado por 523 ejemplos, y al extraer el 80 % de esos datos para crear el conjunto Train/Dev se seleccionaron de forma aleatoria 418 ejemplos. Como la k seleccionada es 10, el conjunto Train/Dev se ha dividido en 10 grupos de $41,8 \approx 42$ ítems para la evaluación con *10-fold Cross Validation*.

Sin embargo, para algunos algoritmos de aprendizaje supervisado más complejos se ha usado *5-fold Cross Validation*, ya que el tiempo de entrenamiento y evaluación en estos casos era excesivamente largo. Por lo tanto, en estos casos el conjunto Train/Dev se ha dividido en 5 grupos de $83,6 \approx 84$ ítems. Los algoritmos en los que se ha hecho división en 5 grupos han sido los siguientes: *SVM*, *RandomForest*, *MLP*, *Bagging*, *AdaBoost*, *XGBoost* y *LightGBM*.

4.3.2.6.2 Métricas de evaluación

Las métricas de evaluación (o figuras de mérito) se utilizan para valorar el rendimiento de un modelo de aprendizaje automático y su objetivo es estimar la precisión de la generalización de un modelo sobre los datos futuros (no vistos/fuera de muestra).

A continuación se muestran las métricas utilizadas en este proyecto:

- **Matriz de confusión:** Es una representación matricial de los resultados de las predicciones de cualquier prueba binaria que se utiliza para describir el rendimiento del modelo de clasificación sobre un conjunto de datos de prueba cuyos valores reales se conocen (ver Figura 4.39). Cada predicción puede ser uno de cuatro resultados, basado en cómo coincide con el valor real:

- *True Positive (TP)*: Predicho *positivo* (1) y *positivo* (1) en realidad.

- *True Negative (TN)*: Predicho *negativo (0)* y *negativo (0)* en realidad.
- *False Positive (FP)*: Predicho *positivo (1)* y *negativo (0)* en realidad.
- *False Negative (FN)*: Predicho *negativo (0)* y *positivo (1)* en realidad.

Los errores de las predicciones se acumulan en FP y FN. En nuestro estudio, el error FP (error de tipo I) significa que se ha predicho la recurrencia de FA cuando en realidad a revertido, mientras que el error FN (error de tipo II) significa que se ha predicho la no recurrencia de FA en un paciente que permanece en la arritmia. En los estudios médicos los errores de tipo II son los más graves y han que intentar minimizarlos.

		Predicted Values	
		Negative	Positive
Actual Values	Negative	TN True Negative	FP False positive
	Positive	FN False Negative	TP True Positive

4.39. Figura: Matriz de confusión con 2 etiquetas de clase

- **Accuracy (exactitud)**: Indica el número de ítems clasificados correctamente en comparación con el número total de ejemplos. Hay que tener en cuenta que la métrica de exactitud tiene limitaciones, ya que no funciona bien con las clases desequilibradas que pueden tener muchos ejemplos de la misma clase e incluir algunas otras clases.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

- **Precision (precisión)**: Representa el número de verdaderos positivos que son realmente positivos en comparación con el número total de valores positivos predichos.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall (exhaustividad)**: Muestra la cantidad de verdaderos positivos que el modelo ha clasificado en función del número total de valores positivos.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F-score (F1)**: Esta métrica es la combinación de las métricas de *precision* y *recall*, y sirve de compromiso entre ellas. La mejor puntuación *F1* es igual a 1 y la peor a 0.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- **Sensitivity (sensibilidad)**: Indica la capacidad de nuestro estimador para dar como casos positivos los casos realmente positivos, es decir, la proporción de ejemplos positivos (con recurrencia de FA) correctamente identificados. Es la misma métrica que *recall*.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

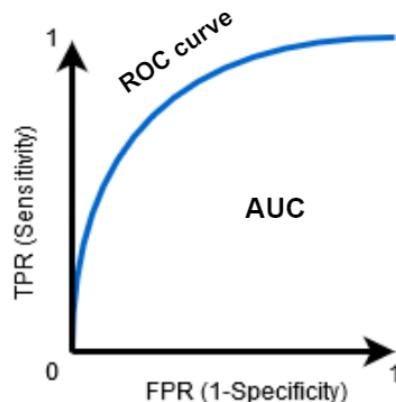
- **Specificity (especificidad)**: Indica la capacidad de nuestro estimador para dar como casos negativos los casos realmente negativos, es decir, la proporción de ejemplos negativos (sin recurrencia de FA) correctamente identificados.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

- **ROC curve y AUC**: El área bajo curva (AUC) es una de las métricas más utilizadas para la evaluación. El AUC de un clasificador es igual a la probabilidad de que el clasificador clasifique un ejemplo positivo elegido al azar más alto que un ejemplo negativo elegido al azar.

ROC es una curva de probabilidad y AUC representa el grado o medida de separabilidad. Indica cuánto es capaz el modelo de distinguir entre las clases. Cuanto mayor sea el AUC, mejor será el modelo para predecir clases de valor 0 como 0 y clases de valor 1 como 1. Por analogía, cuanto mayor sea el AUC, mejor será el modelo para distinguir entre pacientes con la enfermedad y sin enfermedad.

El AUC (área bajo la curva) es una métrica de gran utilidad al no variar dependiendo de la distribución de la clase.



4.40. Figura: Ejemplo de curva ROC y AUC

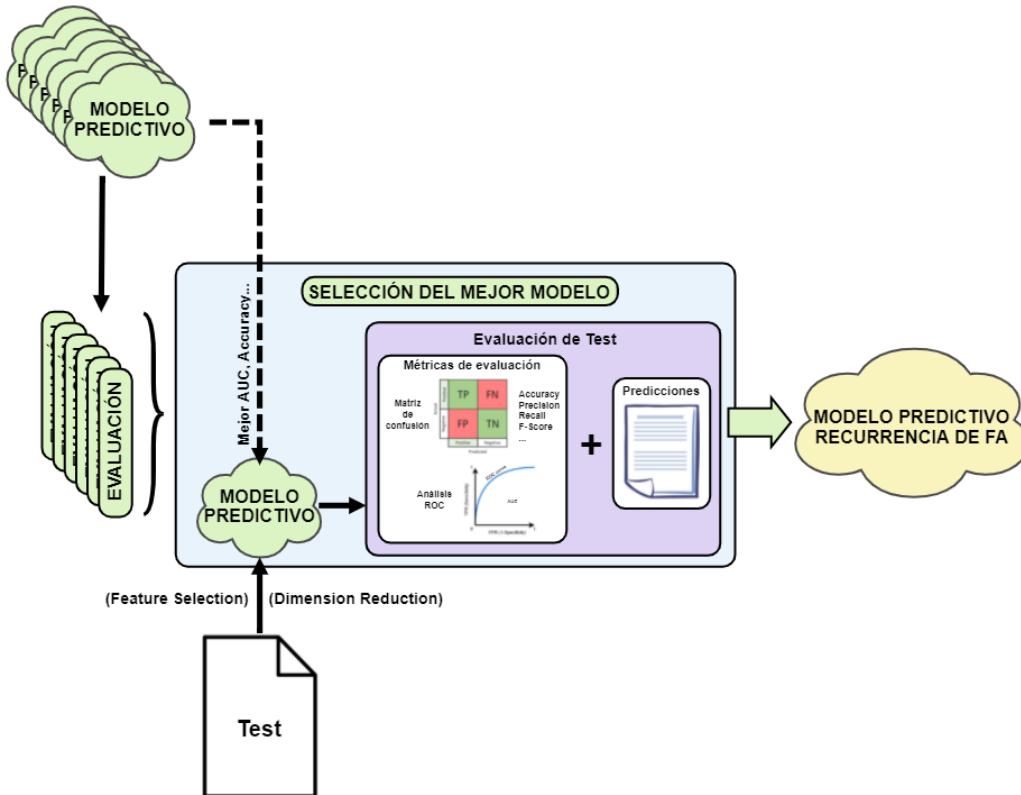
4.3.2.7. Selección del modelo óptimo

En las secciones anteriores se han entrenado y evaluado los modelos predictivos con diferentes algoritmos de clasificación y con variaciones del conjunto de datos de entrenamiento inicial, dependiendo del método de imputación de *missing values*, estandarización/normalización y selección de variables o reducción de dimensiones.

Por lo tanto, para esta fase disponemos de un conjunto de modelos predictivos entrenados y sus respectivas evaluaciones. El objetivo ahora ha sido seleccionar el modelo con mejor rendimiento, y para ello, ha habido que hacer una comparación de las métricas de evaluación. Como se ha comentado al hacer el análisis del conjunto de datos (ver sección *Creación del dataset*), la distribución de la clase no está totalmente balanceada, por lo que se ha escogido la métrica AUC como la referencia para comparar los resultados de las evaluaciones de los modelos.

De esta manera obtenemos el modelo predictivo de recurrencia de FA con mejores resultados en el área bajo la curva (AUC), y el último paso sería probarlo en una muestra nunca antes vista (conjunto Test). Se calculan las métricas de evaluación en el conjunto Test: *accuracy*, *precision*, *recall*, *F1*, *sensitivity*, *specificity*, *AUC*... Y por último, se muestran las predicciones de las intancias del Test (con la probabilidad de la predicción ofrecida por el modelo) (ver Figura 4.42).

El diseño de esta fase de selección del mejor modelo predictivo de recurrencia de fibrilación auricular se muestra en la Figura 4.41.



4.41. Figura: Diseño de la selección del modelo óptimo

Predictions on Test set:						
Instance: 779	Class: 0.0(no AF recurrence)	Predicted: 0.0(no AF recurrence)	Probability: 93.83%	Success: YES		
Instance: 1066	Class: 0.0(no AF recurrence)	Predicted: 0.0(no AF recurrence)	Probability: 80.51%	Success: YES		
Instance: 497	Class: 1.0(AF recurrence)	Predicted: 0.0(no AF recurrence)	Probability: 64.7%	Success: YES		
Instance: 880	Class: 0.0(no AF recurrence)	Predicted: 1.0(AF recurrence)	Probability: 98.57%	Success: NO		
Instance: 89	Class: 0.0(no AF recurrence)	Predicted: 0.0(no AF recurrence)	Probability: 80.59%	Success: YES		
Instance: 111	Class: 1.0(AF recurrence)	Predicted: 1.0(AF recurrence)	Probability: 65.23%	Success: YES		
Instance: 582	Class: 1.0(AF recurrence)	Predicted: 1.0(AF recurrence)	Probability: 64.04%	Success: YES		
Instance: 503	Class: 0.0(no AF recurrence)	Predicted: 0.0(no AF recurrence)	Probability: 54.23%	Success: YES		
Instance: 569	Class: 1.0(AF recurrence)	Predicted: 1.0(AF recurrence)	Probability: 81.74%	Success: YES		
Instance: 380	Class: 0.0(no AF recurrence)	Predicted: 0.0(no AF recurrence)	Probability: 66.2%	Success: YES		

4.42. Figura: Ejemplo de predicciones en 6 instancias del conjunto Test

5. Resultados

En este apartado se van a exponer y analizar los resultados obtenidos tras la implementación del diseño realizado en la sección *Diseño y desarrollo de la solución*. Tras haber entrenado modelos predictivos con diferentes algoritmos y conjuntos de datos dependiendo de la imputación de *missing values*, estandarización/normalización y métodos de *feature selection*, el objetivo es seleccionar el mejor modelo basándose en los resultados de las métricas de evaluación, concretamente, en el área bajo la curva (AUC).

De esta manera, obtendremos el modelo predictivo óptimo de recurrencia de FA a partir de nuestros datos iniciales. También se va a realizar un análisis de las variables predictoras más relevantes, es decir, aquellas que sean más determinantes a la hora de predecir la recurrencia de FA o la vuelta a un ritmo sinusoidal normal.

5.1. Resultados de las evaluaciones de los modelos

Como ya se ha dicho anteriormente, se han entrenado modelos predictivos de recurrencia de FA utilizando estos algoritmos de ML: *DecisionTree*, *KNN*, *LogisticRegression*, *NaiveBayes*, *Perceptron*, *SVM*, *RandomForest*, *MLP*, *Bagging*, *AdaBoost*, *XGBoost* y *LightGBM*.

Por otro lado, se han utilizado diferentes conjuntos de datos dependiendo de los métodos de imputación de *missing values*, estandarización/normalización y *feature selection* utilizados.

Para cada método de preprocessado utilizado, se ha creado una tabla con los resultados de las métricas AUC obtenidas de cada modelo sobre el conjunto de desarrollo/evaluación y prueba. En los Cuadros 5.1 y 5.2 se muestran los resultados tras aplicar RFE como método de selección de variables en el conjunto de datos. Si observamos la primera fila del Cuadro 5.1, vemos que, utilizando el algoritmo *DecisionTree* y con una estandarización de las variables numéricas mediante *StandardScaler*, el mejor AUC en el conjunto de desarrollo/evaluación es de 0.65 al imputar los *missing values* con la media aritmética, y el mejor AUC en el conjunto de prueba es de 0.66 al imputar los *missing values* mediante la media también.

Por cada algoritmo se ha resaltado en color azul el AUC más alto obtenido, y en color amarillo el AUC más alto de toda la tabla. Tras calcular los resultados para todos los algoritmos, vemos que el mejor AUC encontrado sobre el conjunto de desarrollo/evaluación es de **0.77** en los dos siguientes casos:

- Algoritmo ***SVM***, estandarización mediante ***StandardScaler*** e imputación con la **mediana**.
- Algoritmo ***RandomForest***, estandarización mediante ***StandardScaler*** e imputación con la **mediana**.

		TRAIN/DEV	
		AUC	MV imputation
DecisionTree	StandardScaler	0.65	Mean
	MinMaxScaler	0.63	Median
	MaxAbsScaler	0.63	Diff. value
	QuantileTransformer	0.62	Mean
	Normalizer	0.90	Diff. value
<hr/>			
KNN	StandardScaler	0.72	Mean
	MinMaxScaler	0.71	Median
	MaxAbsScaler	0.70	Predicted
	QuantileTransformer	0.71	Mean
	Normalizer	0.66	Mean/Median
<hr/>			
Logistic Regression	StandardScaler	0.76	Mean/Median
	MinMaxScaler	0.76	Mean/Med/Pred
	MaxAbsScaler	0.76	Predicted
	QuantileTransformer	0.76	Mean/Med/Pred
	Normalizer	0.70	Mean/Med/Pred
<hr/>			
NaiveBayes	StandardScaler	0.70	Mean/Median
	MinMaxScaler	0.73	Mean
	MaxAbsScaler	0.72	Predicted
	QuantileTransformer	0.75	Mean
	Normalizer	0.67	Mean/Med/Pred/Diff
<hr/>			
Perceptron	StandardScaler	0.72	Mean
	MinMaxScaler	0.76	Predicted
	MaxAbsScaler	0.76	Predicted
	QuantileTransformer	0.73	Mean/Med/Pred
	Normalizer	0.66	Predicted
<hr/>			
SVM	StandardScaler	0.77	Median
	MinMaxScaler	-	-
	MaxAbsScaler	-	-
	QuantileTransformer	0.73	Mean
	Normalizer	-	-
<hr/>			
RandomForest	StandardScaler	0.77	Median
	MinMaxScaler	-	-
	MaxAbsScaler	-	-
	QuantileTransformer	0.75	Diff. value
	Normalizer	-	-
<hr/>			
MLP	StandardScaler	0.75	Mean/Median
	MinMaxScaler	-	-
	MaxAbsScaler	-	-
	QuantileTransformer	0.75	Median/Predicted
	Normalizer	-	-

5.1. Cuadro: Mejores resultados en AUC tras *feature selection* con RFE (I).

		TRAIN/DEV	
		AUC	MV imputation
XGBoost	StandardScaler	0.76	Mean
	MinMaxScaler	0.73	Mean/Median
	MaxAbsScaler	0.73	Median
	QuantileTransformer	0.74	Diff. value
	Normalizer	0.67	Diff value
<hr/>			
LightGBM	StandardScaler	0.75	Median
	MinMaxScaler	0.74	Mean
	MaxAbsScaler	0.73	MV/Predicted
	QuantileTransformer	0.74	Diff. value
	Normalizer	0.68	Mean/Med/Pred
<hr/>			
Bagging	StandardScaler	0.76	Mean/Median
	MinMaxScaler	0.76	Median
	MaxAbsScaler	0.75	Predicted
	QuantileTransformer	0.75	Mean/Med/Pred
	Normalizer	0.69	Mean/Med/Diff
<hr/>			
AdaBoost	StandardScaler	0.76	Mean/Median
	MinMaxScaler	0.76	Median
	MaxAbsScaler	0.75	Predicted
	QuantileTransformer	0.75	Mean/Med/Pred
	Normalizer	0.69	Mean/Med/Diff

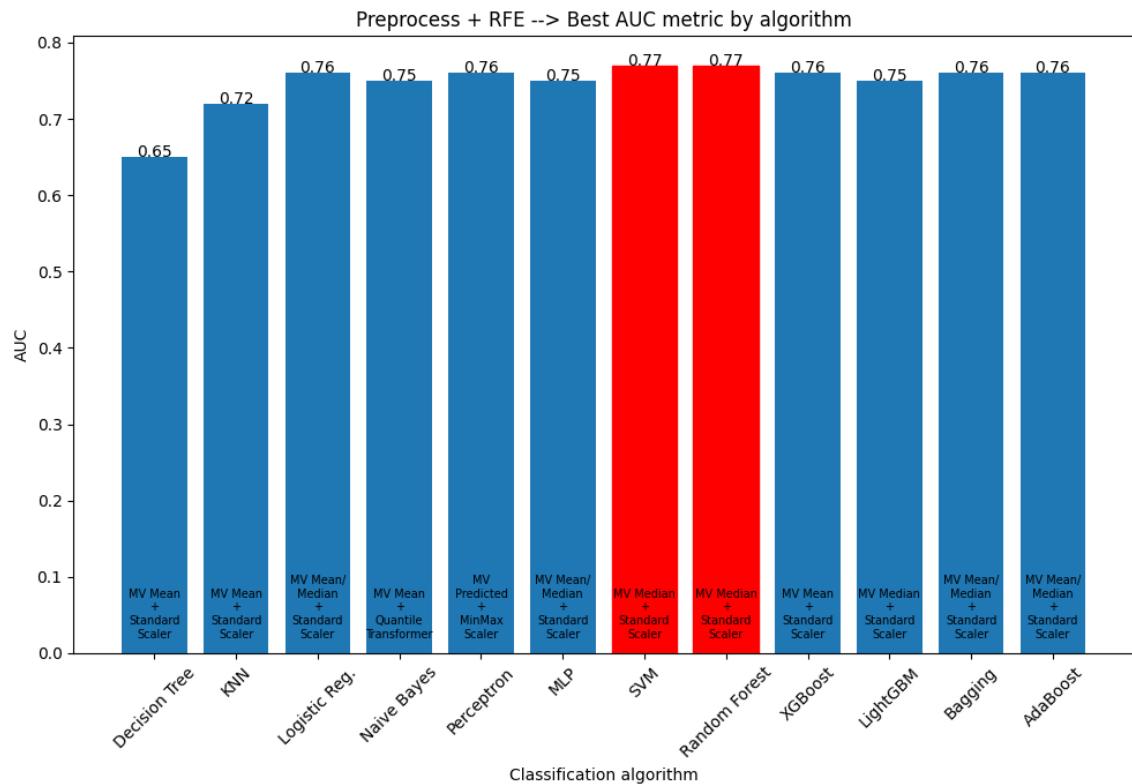
5.2. Cuadro: Mejores resultados en AUC tras *feature selection* con RFE (II).

NOTA (I): estas son las abreviaturas utilizadas en la imputacion de *missing values*:

- *Mean*: media aritmética de los valores de la variable.
- *Median / Med*: mediana de los valores de la variable.
- *Predicted / Pred*: valores predichos mediante regresión lineal.
- *Diff. values / Diff*: valor diferente para todos los *missing values*.
- *MV*: no se ha aplicado ninguna técnica de imputación (solo posible en los algoritmos *XGBoost* y *LightGBM*).

NOTA (II): en algunas pruebas de la tabla, en los algoritmos *SVM*, *RandomForest*, y *MLP*, no hay resultados, ya que no se han entrenado esos modelos porque el tiempo de entrenamiento y optimización de hiperparámetros era muy alto.

En la Figura 5.1 se muestran en forma de gráfico de barras los mejores resultados de la tabla anterior en la métrica AUC por algoritmo. Los resultados son similares para la mayoría de algoritmos y en rojo aparecen resaltados los valores más altos de AUC en los algoritmos *SVM* y *RandomForest* (**0.77**).

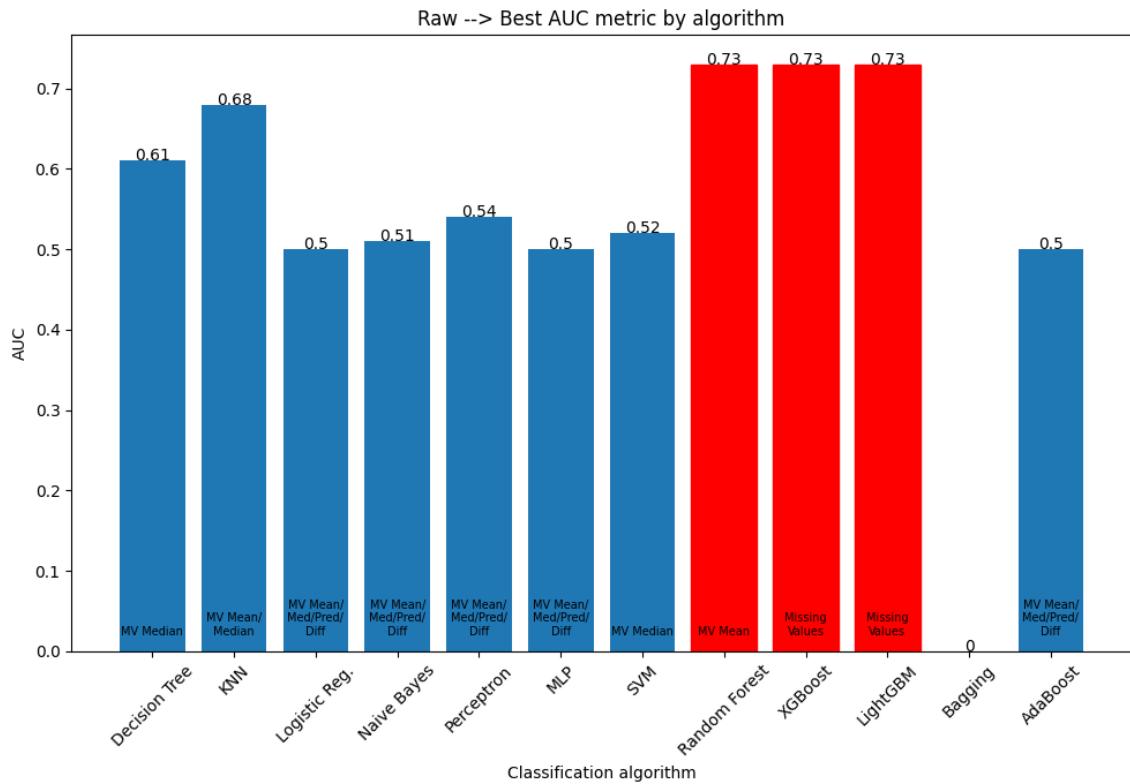


5.1. Figura: Mejores AUC por algoritmo tras *feature selection* con *RFE*

Este mismo análisis de resultados se ha hecho utilizando el resto de conjuntos de datos con otras técnicas de selección de variables o reducción de dimensionalidad aplicadas, incluso sin aplicar ninguna de ellas o ningún método de estandarización/normalización.

- *Raw*: sin ningún método de estandarización/normalización ni *feature selection* o reducción de dimensionalidad.
- *Preprocess*: solamente aplicando métodos de estandarización/normalización pero sin *feature selection* o reducción de dimensionalidad.
- *Preprocess + RFE*: aplicando métodos de estandarización/normalización y *feature selection* mediante *RFE*.
- *Preprocess + Lasso_SelectFromModel*: aplicando métodos de estandarización/normalización y *feature selection* mediante *Lasso_SelectFromModel*.
- *Preprocess + Lasso_SFS*: aplicando métodos de estandarización/normalización y *feature selection* mediante *Lasso_SFS*.
- *Preprocess + Trees_SelectFromModel*: aplicando métodos de estandarización/normalización y *feature selection* mediante *Trees_SelectFromModel*.
- *Preprocess + SelectKBest*: aplicando métodos de estandarización/normalización y *feature selection* mediante *SelectKBest*.
- *Preprocess + PCA*: aplicando métodos de estandarización/normalización y reducción de dimensionalidad mediante *PCA*.

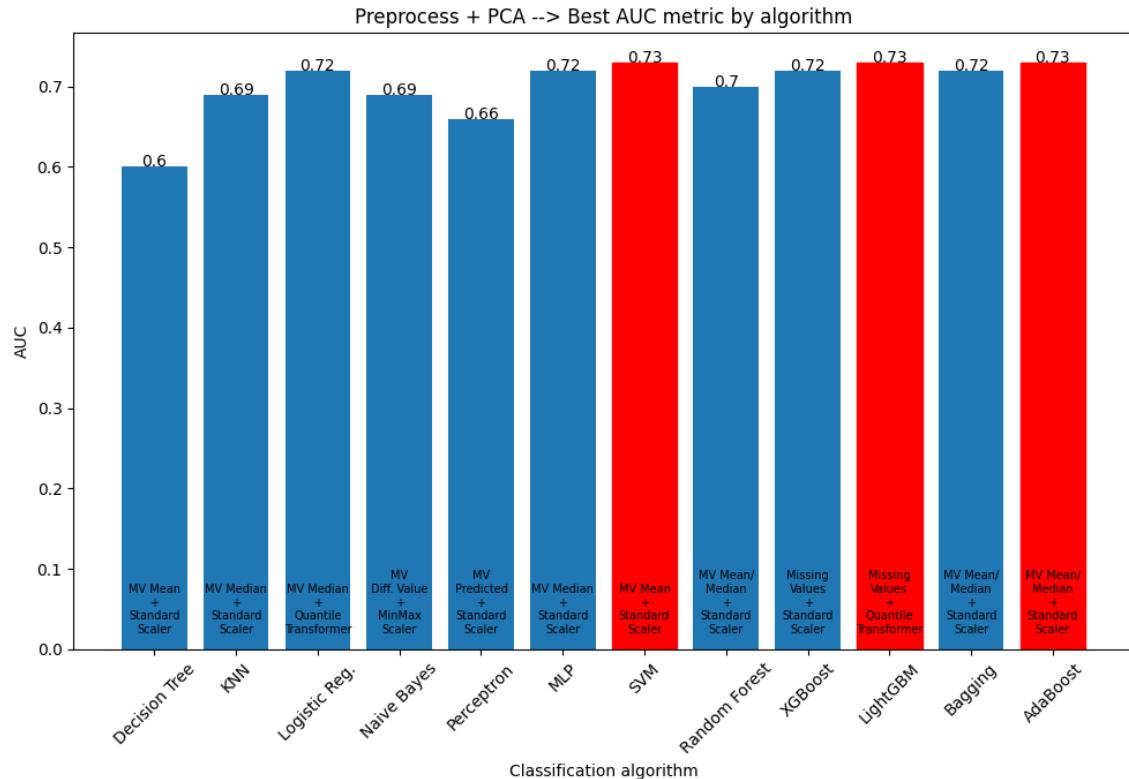
A continuación, se muestra un ejemplo de los resultados obtenidos con el conjunto de datos *Raw* (ver Figura 6.2), y se puede comprobar como los métodos de estandarización/normalización y *feature selection* ayudan a mejorar el rendimiento de los modelos predictivos.



5.2. Figura: Mejores AUC por algoritmo sobre los conjuntos de datos *Raw*

Si se comparan los resultados sobre el conjunto *Raw* (ver Figura 6.2) con los obtenidos sobre el conjunto con preprocesado de datos y *feature selection* mediante *RFE* (ver Figura 5.1), se ve como la métrica AUC mejora considerablemente tras aplicar estas últimas técnicas. En el conjunto *Raw* ninguno de los algoritmos alcanza el 0.77 de AUC obtenido con el conjunto de datos con *RFE*. Sin embargo, hay algoritmos como *RandomForest*, *XGBoost* y *LightGBM* que son capaces de ofrecer resultados bastante buenos sin la necesidad de un previo preprocesado de datos.

Otra duda surgida en el desarrollo del proyecto era si aplicar métodos de selección de variables siempre iba a ofrecer mejores resultados que aplicar un método de reducción de dimensiones del vector de rasgos como puede ser PCA (Principal Component Analysis). En la Figura 5.3 se muestran los resultados obtenidos tras el entrenamiento de modelos en un conjunto con *PCA* aplicado.



5.3. Figura: Mejores AUC por algoritmo tras reducción de dimensionalidad con *PCA*

Como se muestra en la Figura 5.3 tras aplicar *PCA* los resultados en la métrica AUC mejoran en gran parte de los algoritmos. Aun así, no se alcanzan los valores obtenidos tras *feature selection* con *RFE*.

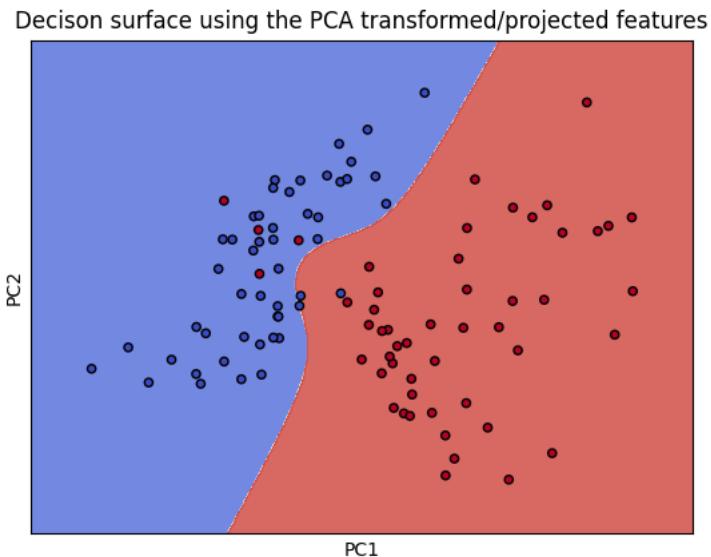
5.2. Resultados del mejor modelo predictivo

Tras analizar los resultados de todos los modelos predictivos entrenados, se va a escoger el de mayor AUC sobre el conjunto de datos de desarrollo/evaluación. En este caso, se va a elegir el modelo predictivo entrenado mediante el algoritmo ***SVM*** con el siguiente preprocesado de datos: imputación de *missing values* mediante la **mediana**, estandarización de las variables numéricas mediante **StandardScaler** y selección de variables mediante **RFE**. El área bajo la curva (AUC) obtenido con este modelo sobre el conjunto de desarrollo/evaluación es de **0.77** (ver Cuadro 5.3).

En el Cuadro 5.3 se muestran las características del modelo entrenado (algoritmo y conjunto de datos de entrenamiento) y los resultados de la evaluación mediante *5-fold Cross Validation* utilizando el conjunto Train/Dev.

Como ya se ha mencionado el algoritmo utilizado es *SVM* (*Support Vector Machines*), más concretamente, se ha utilizado *SVC* (*C-Support Vector Classification*) de la librería Scikit-learn de Python. El algoritmo consiste en encontrar un hiperplano de separación o conjunto de hiperplanos (*vector soporte*) en un espacio de dimensionalidad muy alto, separando las clases en 2 espacios con el mayor margen posible (ver sección 4.3.2.5).

En la Figura 5.4 se ve un ejemplo de un conjunto de datos reducido a dos dimensiones mediante *PCA*, en el que un algoritmo *SVM* ha creado ese hiperplano de separación para tratar de realizar una clasificación binaria separando las instancias de las dos clases (diferentes colores) con el mayor margen posible.



5.4. Figura: Ejemplo de hiperplano de separación creado con SVM

La optimización de hiperparámetros realizada en el modelo escogido ha sido la siguiente (ver Cuadro 5.3):

- **$C = 1$** : El parámetro C agrega una penalización por cada punto de datos mal clasificado. Es directamente proporcional de la distancia al límite de decisión, es decir, si C es pequeño, la penalización por puntos mal clasificados es baja y se elige un límite de decisión con un gran margen, y si C es grande, la penalización es alta y da como resultado un límite de decisión con un margen más pequeño.
- **$kernel = poly$** : El parámetro $kernel$ selecciona el tipo de hiperplano utilizado para separar los datos. Un $kernel$ de tipo $poly$ utiliza un hiperplano no lineal a través de una función polinómica.
- **$gamma = 0.1$** : El parámetro $gamma$ indica la influencia de cada uno de los ejemplos de entrenamiento en la penalización al ser mal clasificado. Se usa cuando el kernel es RBF , en este caso al ser $poly$ es irrelevante.
- **$degree = 2$** : El parámetro $degree$ es el grado de la función polinómica del hiperplano de separación. En este caso se usa una función polinómica de segundo grado.

A continuación se va a hacer un análisis de los resultados de la evaluación (ver Cuadro 5.3). Para comenzar, se ha obtenido un **AUC** de **0.77**, que como se sabe, cuanto mayor sea su valor (máximo de 1), mejor será el modelo para distinguir entre pacientes con recurrencia o no de FA.

Por otro lado, se ha obtenido una exactitud (**accuracy**) de **0.70**. No es una métrica muy válida al tratarse de un conjunto de datos no equilibrado totalmente en cuanto a la clase, ya que hay más ejemplos con recurrencia de FA. Es por ello que al revisar la matriz de confusión o figuras de mérito como *precision*, *recall*, *specificity* o *sensitivity*, se ve que el modelo aprende a clasificar mejor pacientes con recurrencia de FA que pacientes sin recurrencia.

Teniendo en cuenta la sensibilidad (**sensitivity**), vemos que la proporción de ejemplos positivos (con recurrencia de FA) correctamente clasificados es del **86 %**. Mientras que, en cuanto a la especificidad (**specificity**), la proporción de ejemplos negativos (sin recurrencia de FA) correctamente clasificados es del **47 %**. El algoritmo no llega a clasificar ni la mitad de pacientes sin recurrencia de FA de manera correcta: en 80 pacientes predice correctamente que no hay recurrencia, y en 90 predice incorrectamente que sí la hay. Por lo tanto, este es un aspecto del modelo que nos gustaría mejorar.

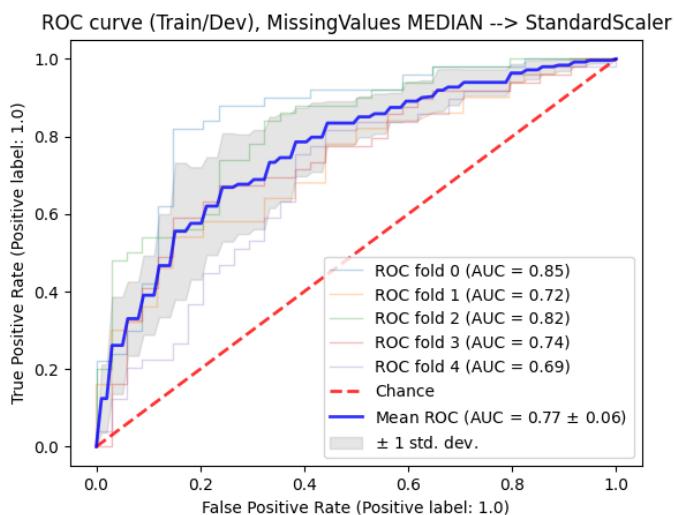
MODEL				
MISSING VALUES IMPUTATION			Median	
PREPROCESSING			StandardScaler	
FEATURE SELECTION			RFE	
ALGORITHM	Hyperparameters Tuning		SVM	
	C	1		
	gamma	0.1		
	kernel	poly		
	degree	2		

CONFUSION MATRIX				
		Predicted Label		
		no AF recurrence	AF recurrence	
True Label	no AF recurrence	80	90	
	AF recurrence	34	214	

EVALUATION METRICS				
	Precision	Recall	F1-score	
no AF recurrence	0.70	0.47	0.56	
AF recurrence	0.70	0.86	0.78	
Weighted Avg.	0.70	0.70	0.69	
<hr/>				
Accuracy	0.70			
Specificity	0.47			
Sensitivity	0.86			
AUC	0.77			

5.3. Cuadro: Resultados tras evaluación con *5-fold Cross Validation* en Train/Dev (I)

En la Figura 5.5 se muestran las curvas ROC en cada una de 5 evaluaciones (*5-fold*) realizadas sobre el conjunto Train/Dev y sus AUC correspondientes. También se muestra la media de las curvas ROC y la media del área bajo la curva (AUC) con su desviación estándar (**0.77 ± 0.06**).

5.5. Figura: Curva ROC y AUC tras evaluación con *5-fold Cross Validation* en Train/Dev

5.2.1. Modelo óptimo con mejoras

En este apartado se muestran las evaluaciones tras unas modificaciones en el modelo predictivo de la sección anterior, con el objetivo de mejorar los resultados obtenidos.

Se han creado 4 variantes del conjunto de datos utilizado en busca de una mejora en los resultados:

- Eliminación de variables predictoras con un porcentaje alto de *missing values*. Se ha obtenido un AUC de **0.78**.
- Eliminación de variables predictoras con un porcentaje alto de *missing values* y discretización de la variable *area_ai*: ≤ 20 , $(20,30]$, $(30,40]$ y > 40 . Se ha obtenido un AUC de **0.75**.
- Eliminación de variables predictoras con un porcentaje alto de *missing values* y discretización de la variable *edad*: ≤ 20 , $(20,40]$, $(40,60]$, $(60,70]$, $(70,80]$, $(80,90]$, $(90,100]$ y > 100 . Se ha obtenido un AUC de **0.72**.
- Eliminación de variables predictoras con un porcentaje alto de *missing values* y discretización de las variables *area_ai* y *edad*. Se ha obtenido un AUC de **0.77**.

De las 4 variaciones realizadas en el conjunto de datos de entrenamiento, vemos como al eliminar variables con un gran número de *missing values* el **AUC** resultante supera ligeramente al 0.77 obtenido en la sección anterior (pasa a un **0.78**). Tras varias pruebas realizadas, se ha obtenido este resultado **eliminando las variables con un 85 % o más de missing values** en el conjunto de entrenamiento: *albumina*, *hba1c*, *bnp*, *troponina_tni*, *troponina_tnt*, *dimero_d*, *fibrinogeno*, *aldosterona*, *leucocitos*, *pct* y *tl3*.

Por lo tanto, con este cambio, el mismo preprocesado realizado anteriormente (imputación de *missing values* mediante la mediana, estandarización mediante *StandardScaler* y selección de variables mediante *RFE*) y el algoritmo *SVM* con la misma optimización de parámetros, se obtienen los resultados mostrados en el Cuadro 5.4.

En este modelo, también mejoran ligeramente las métricas **accuracy (0.71)**, **precision (0.71)**, **recall (0.71)** y **F-score (0.70)**. En cuanto a la **sensibilidad** y **especificidad**, se reduce un poco el porcentaje de acierto en instancias con recurrencia de FA clasificadas correctamente (pasa de un 86 % a un **83 %**), pero aumenta el porcentaje de acierto en pacientes sin recurrencia de FA clasificados correctamente (pasa de un 47 % a un **54 %**). Esto puede deberse a que algunas de las variables eliminadas por el alto número de *missing values* como *troponina_tni* favorecían a un mayor entrenamiento en predicciones positivas (recurrencia de FA), ya que los valores imputados tenían gran correlación con la recurrencia. Con este nuevo modelo se consigue clasificar correctamente más de la mitad de pacientes sin recurrencia de FA (92 de 170).

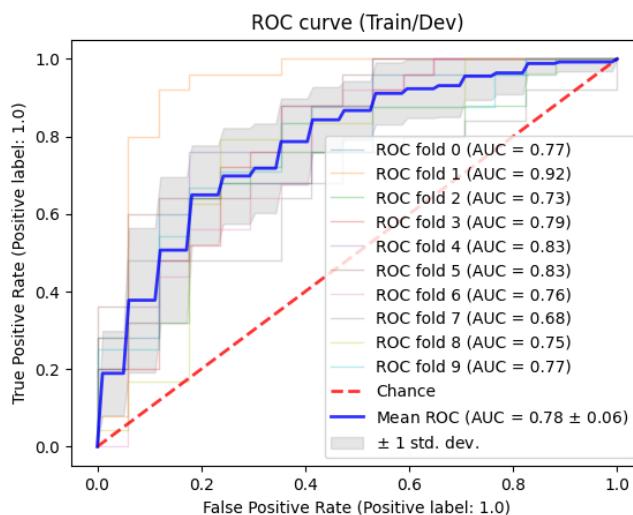
MODEL			
MISSING VALUES IMPUTATION		Median	
PREPROCESSING		StandardScaler	
FEATURE SELECTION		RFE	
ALGORITHM	SVM		
	Hyperparameters Tuning	C	1
		gamma	0.1
		kernel	poly
		degree	2

CONFUSION MATRIX			
		Predicted Label	
		no AF recurrence	AF recurrence
True Label	no AF recurrence	92	78
	AF recurrence	43	205

EVALUATION METRICS			
	Precision	Recall	F1-score
no AF recurrence	0.68	0.54	0.60
AF recurrence	0.72	0.83	0.77
Weighted Avg.	0.71	0.71	0.70
Accuracy	0.71		
Specificity	0.54		
Sensitivity	0.83		
AUC	0.78		

5.4. Cuadro: Resultados tras evaluación con *10-fold Cross Validation* en Train/Dev (II)

En la Figura 5.6 se muestran las curvas ROC en cada una de 10 evaluaciones (*10-fold*) realizadas sobre el conjunto Train/Dev y sus AUC correspondientes. También se muestra la media de las curvas ROC y la media del área bajo la curva (AUC) con su desviación estándar (**0.78 ± 0.06**).



5.6. Figura: Curva ROC y AUC tras evaluación con *10-fold Cross Validation* en Train/Dev

5.2.1.1. Resultados en el conjunto Test

En este apartado se van a mostrar los resultados obtenidos al clasificar las instancias del conjunto Test con el modelo anterior. El conjunto Test está formado por ejemplos totalmente nuevos, nunca antes

vistos en el entrenamiento. Al igual que el conjunto de entrenamiento, este también necesita una fase de preprocesado y ha sido realizada de acuerdo a los métodos aplicados en el conjunto de entrenamiento:

- Eliminación del conjunto Test de las mismas variables descartadas en el conjunto entrenamiento que superaban el 85 % de *missing values*.
- Imputación de la mediana del conjunto de entrenamiento en las variables numéricas del Test con *missing values*.
- Estandarización de las variables numéricas en el conjunto Test utilizando el escalado realizado en el conjunto de entrenamiento mediante *StandardScaler*.
- Eliminación del conjunto Test de las mismas variables descartadas en el conjunto entrenamiento mediante *RFE*.

A continuación se muestran los resultados de la evaluación sobre el conjunto Test (ver Cuadro 5.5). La mayoría de métricas presentan valores ligeramente más bajos a los obtenidos en la evaluación sobre el conjunto de desarrollo/evaluación, lo que es habitual ya que estos datos son completamente nuevos para el modelo predictivo.

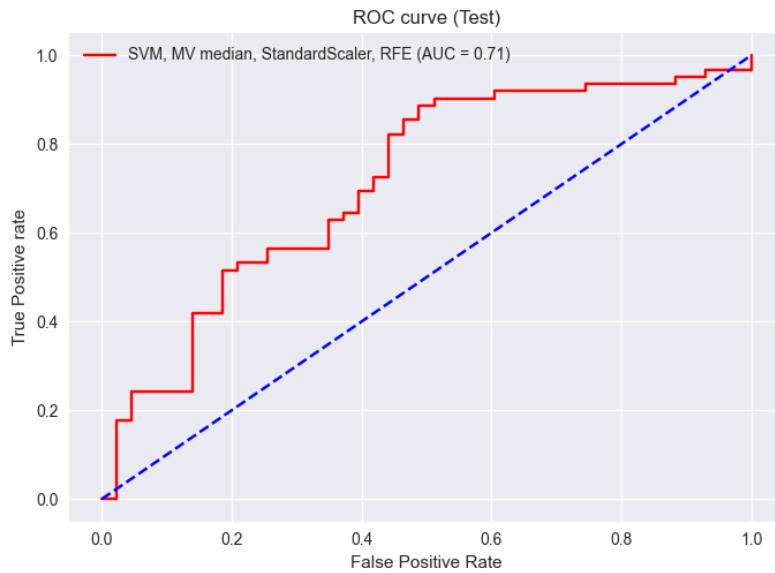
Entre otros, vemos un área bajo la curva (**AUC**) de **0.71** y un **accuracy** de **0.70**. La **especificidad** aumenta respecto a la obtenida en el conjunto de evaluación (**0.56**).

CONFUSION MATRIX (Test)			
		Predicted Label	
		no AF recurrence	AF recurrence
True Label	no AF recurrence	24	19
	AF recurrence	13	49

EVALUATION METRICS (Test)			
	Precision	Recall	F1-score
no AF recurrence	0.65	0.56	0.60
AF recurrence	0.72	0.79	0.75
Weighted Avg.	0.69	0.70	0.69
Accuracy	0.70		
Specificity	0.56		
Sensitivity	0.79		
AUC	0.71		

5.5. Cuadro: Resultados tras evaluación en Test

En la Figura 5.7 se muestra la curva ROC y el AUC correspondiente (**0.71**) tras la evaluación sobre el conjunto de Test.



5.7. Figura: Curva ROC y AUC tras evaluación en Test

Por último, en la Figura 5.8 se pueden observar las predicciones realizadas a 10 de las instancias pertenecientes al conjunto Test. Tomando como ejemplo la instancia con identificador 779, se da la siguiente información acerca de su predicción:

- Clase real de la instancia: 0 (no hay recurrencia de FA)
- Clase predicha por el modelo: 0 (no hay recurrencia de FA)
- Probabilidad de que la predicción sea correcta: 93.83 %
- Éxito, se ha predicho la clase real correctamente.

Predictions on Test set:

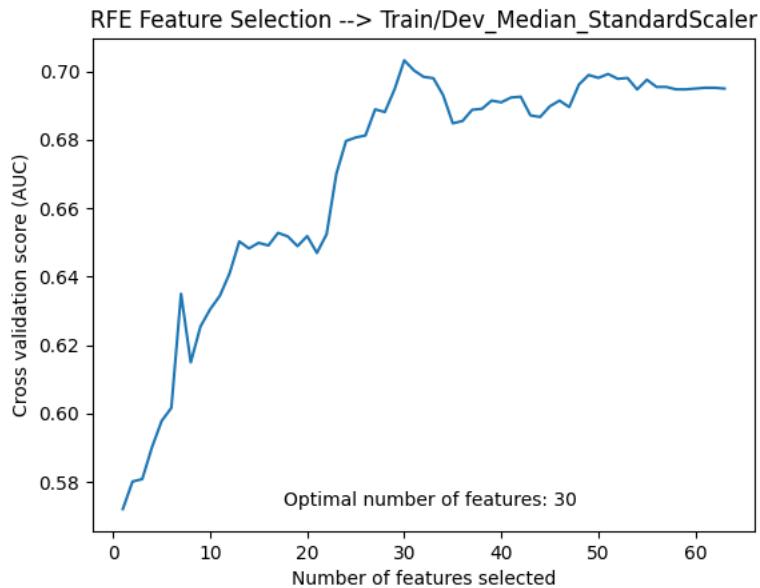
Instance:	Class:	Predicted:	Probability:	Success:
779	0.0(no AF recurrence)	0.0(no AF recurrence)	93.83%	YES
1066	0.0(no AF recurrence)	0.0(no AF recurrence)	80.51%	YES
497	1.0(AF recurrence)	0.0(no AF recurrence)	64.7%	YES
880	0.0(no AF recurrence)	1.0(AF recurrence)	98.57%	NO
89	0.0(no AF recurrence)	0.0(no AF recurrence)	80.59%	YES
111	1.0(AF recurrence)	1.0(AF recurrence)	65.23%	YES
582	1.0(AF recurrence)	1.0(AF recurrence)	64.04%	YES
503	0.0(no AF recurrence)	0.0(no AF recurrence)	54.23%	YES
569	1.0(AF recurrence)	1.0(AF recurrence)	81.74%	YES
380	0.0(no AF recurrence)	0.0(no AF recurrence)	66.2%	YES

5.8. Figura: Predicción sobre 10 instancias de Test

5.2.1.2. Variables predictoras seleccionadas

Anteriormente se han mencionado las variables descartadas por tener un 85 % o más de *missing values*, y en este apartado, se va a hacer un análisis de las variables seleccionadas por el método de *feature selection RFE* aplicado al conjunto de entrenamiento utilizado en el modelo predictivo óptimo.

El algoritmo *RFE* entrena modelos predictivos descartando en cada iteración la variable menos importante. Los modelos se han entrenado con un algoritmo *SVM* con kernel lineal. Finalmente, se ha elegido el número de variables con las que el modelo ofrecía un área bajo la curva (AUC) más alto. En el gráfico de la Figura 5.9 se muestra como con **30 variables** se consigue el AUC más alto.



5.9. Figura: Métrica AUC respecto al número de variables seleccionadas

Las 30 variables más relevantes seleccionadas por el método *RFE* son las siguientes: *potasio*, *no_hdl*, *colesterol*, *ntproBNP*, *vsg*, *fevi*, *diametro_ai*, *area_ai*, *numero_dias_desde_ingreso_hasta_evento*, *numero_dias_ingresado*, *edad*, *ablacion*, *ansiedad*, *demencia*, *sahos*, *hipertiroidismo*, *cardiopatia_isquemica*, *valvula_mitral_reumaticas*, *genero*, *pensionista*, *residenciado*, *n05a*, *n05b*, *c01*, *c01b*, *c02*, *c04*, *c09*, *c10*, y *polimedicion*.

Con la información de estas variables el modelo trata de predecir si un paciente va a sufrir recurrencia de fibrilación auricular o no. El significado de cada una de las variables seleccionadas es el siguiente:

- Variables demográficas:
 - La **edad** del paciente en el momento del primer evento de FA. Es una variable que se relaciona de forma directa con la incidencia de FA, por lo que es razonable que también sea importante para la permanencia o recurrencia de la arritmia.
 - El **género**. Ya era una variable importante en la incidencia de FA al ser mas frecuente en varones y relacionarse con mayor permanencia en el género femenino.
 - La situación de trabajo (**pensionista o no**) y el estado de dependencia (**ingresado en residencia o no**). Hasta ahora estas variables no se habían asociado de forma directa con el desarrollo o permanencia de FA. Son variables relacionadas con la edad, ya que ser pensionista o estar ingresado en una residencia es mucho más frecuente en personas con mayor edad. Esta podría ser una de las razones por la que el algoritmo otorga una mayor importancia a estas variables.
- Historia médica personal previa al diagnóstico de FA:
 - **Ansiedad.** La presencia de este diagnóstico ya se tenía en cuenta en la incidencia de FA, y nuestro modelo también parece tenerla en cuenta para la predicción de recurrencia de la misma.
 - **Demencia.** Ya se conocía que los pacientes con FA eran más propensos a desarrollar demencia, y esta enfermedad también parece estar relacionada inversamente con la progresión o recurrencia de la arritmia. También se conocía la asociación de la demencia con personas de mayor edad.
 - **SAOS.** Es una patología respiratoria ya incluida en otros *scores* de predicción de riesgo de recurrencia de FA, y nuestro modelo también la trata como una variable importante.
 - **Hipertiroidismo.** Es otra patología relacionada con el desarrollo de FA y que el modelo predictivo de recurrencia también parece tenerla en cuenta.

- Historia cardiovascular específica previa al diagnóstico de FA:

- **Cardiopatía isquémica.**
- **Enfermedades de la válvula mitral reumáticas.**

Son dos variables ya incluidas por otros *scores* de predicción de riesgo de FA.

- Hospitalizaciones:

- **Cuántos días antes del evento de FA fue el último ingreso.**
- **Cuántos días ingresó.**

El modelo predictivo de recurrencia de FA da importancia a estas variables acerca de aspectos hospitalarios, ya que podrían ser determinantes la gravedad y cercanía de ingresos previos al evento de FA.

- Pruebas de laboratorio:

- **Potasio.**
- **No HDL.**
- **Colesterol.**
- **NT-proBNP.**
- **VSG.**

De entre todas las pruebas de laboratorio incorporadas al modelo inicial, estas han sido las seleccionadas con mayor influencia para la predicción de recurrencia de FA.

- Tratamientos realizados (fármacos) desde un año antes del evento de FA y posteriormente:

- Se han seleccionado los antipsicóticos **N05A** y las benzodiacepinas **N05B**.
- Estos fármacos del aparato cardiovascular han sido escogidos como buenos predictores de recurrencia de FA: **C01, C02, C04, C09 y C10**.
- Los fármacos **C01B** se consideran determinantes en la predicción de recurrencia o reversión de la arritmia, ya que son antiarrítmicos clase I o III utilizados para el control de ritmo.
- La **polimedición**, la cual se da cuando el paciente ha sido tratado con 5 o más fármacos de los analizados, ha sido escogida como una variable relevante.

- Procedimientos en el momento del evento de FA:

- La **FEVI**, el **diámetro de aurícula izquierda** y el **área de aurícula izquierda** obtenidos en los procedimientos realizados (ecocardiogramas) en el momento del diagnóstico de FA son valores a tener en cuenta también para la predicción de recurrencia de la arritmia.

- Intervenciones tras el evento de FA:

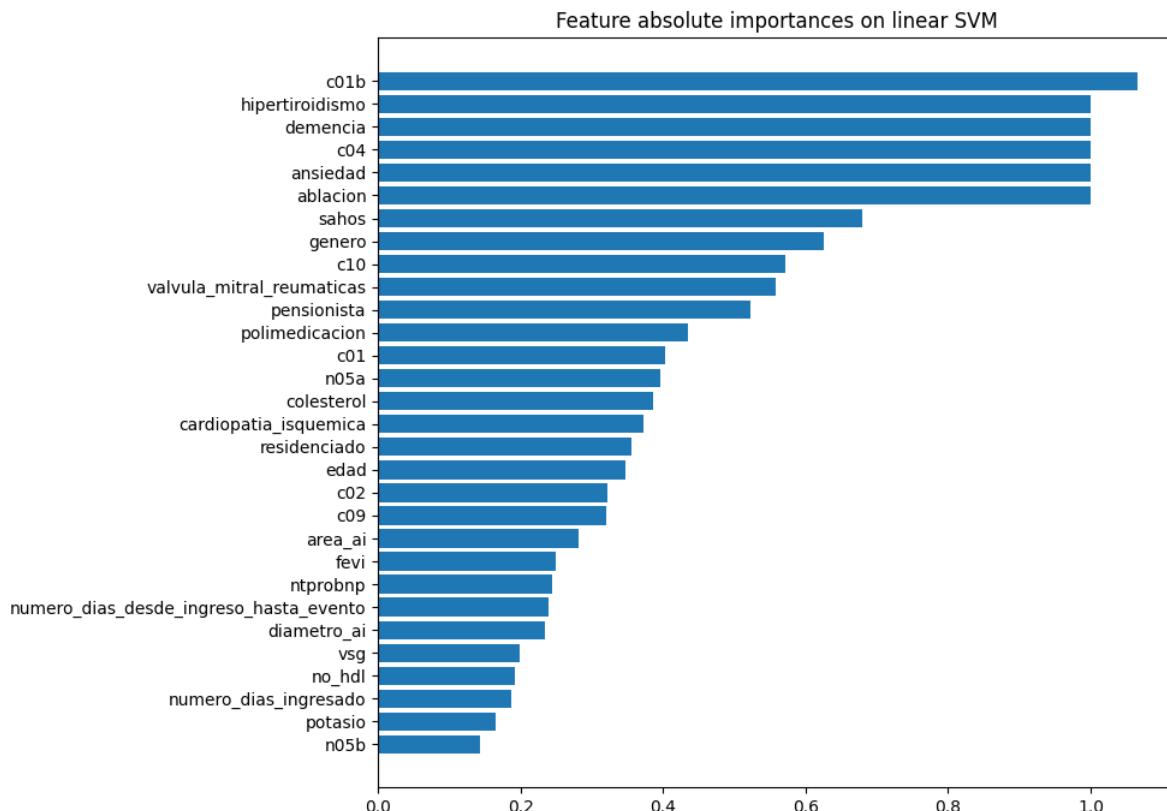
- La realización de una **ablación** por radiofrecuencia (por catéter) tras el evento de FA ha sido considerada una variable a tener en cuenta para la predicción de recurrencia o reversión de la arritmia.

Para determinar la contribución de cada una de las variables en el modelo entrenado se ha utilizado un *SVM* con un *kernel* lineal. No es posible obtener la contribución de las variables en otros *kernels* (como el polinómico de grado 2), ya que los datos son transformados a otro espacio.

En un *SVM* lineal el hiperplano separador resultante está en el mismo espacio que los rasgos (*features*) iniciales. Por lo tanto, sus coeficientes pueden ser interpretados como los pesos de las dimensiones iniciales. En otros *kernels*, sin embargo, el hiperplano separador resultante existe en otro espacio (resultado del espacio original transformado por el *kernel*), por lo que sus coeficientes no están directamente relacionados con el espacio inicial.

Es por ello que para tratar de obtener la contribución de cada una de las variables en las predicciones del modelo, se ha entrenado otro modelo *SVM* pero con un *kernel* lineal. Los resultados de este nuevo modelo son prácticamente iguales al de *kernel* polinómico de grado 2 (AUC de 0.78 también), por lo que nos puede servir para hacernos una idea de la importancia de las variables predictoras.

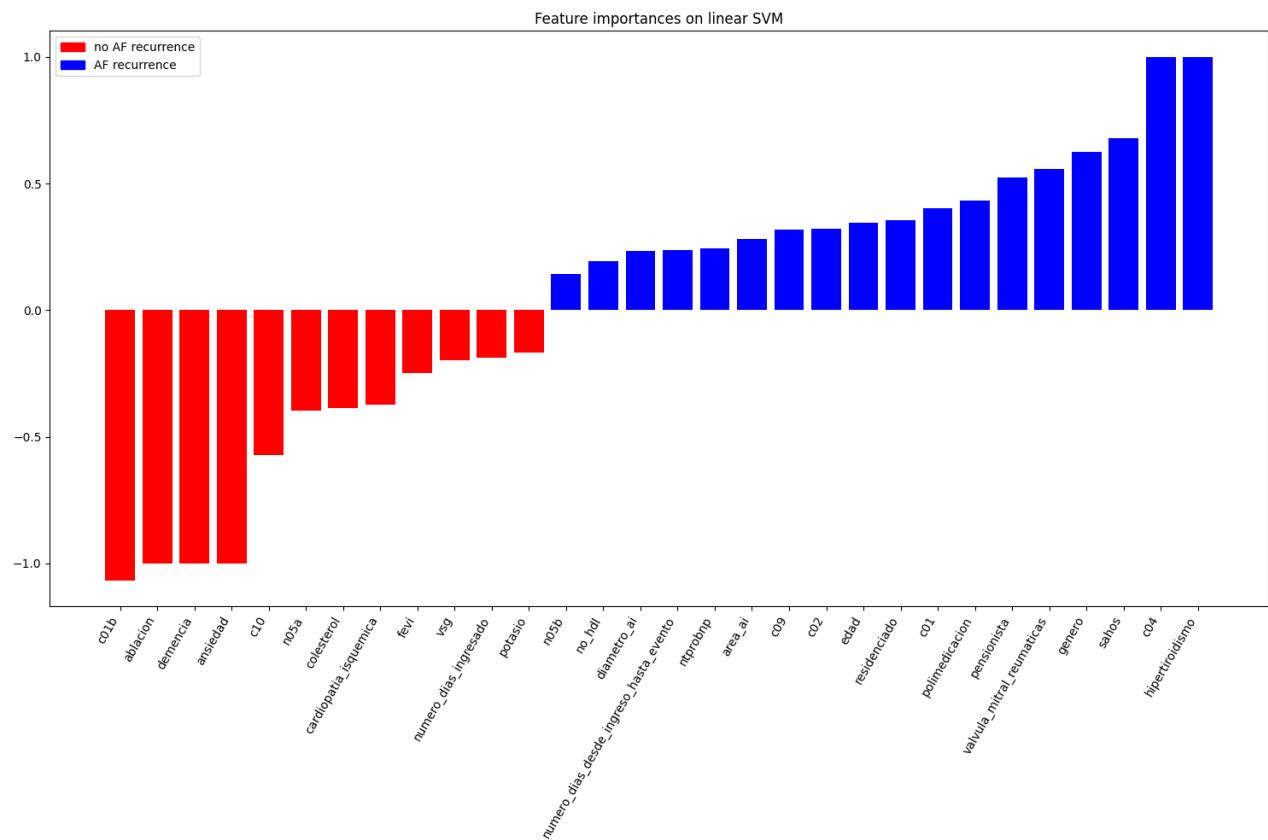
En la Figura 5.10 se muestran los coeficientes de los pesos (importancia) asignados por el algoritmo a cada variable a la hora de realizar las predicciones. En el gráfico aparecen los valores absolutos de las importancias, es decir, la importancia de cada variable ya sea para predecir la clase positiva (recurrencia de FA) como la clase negativa (no recurrencia de FA).



5.10. Figura: Coeficientes (absolutos) de importancia de las variables en un modelo *SVM* lineal

En la Figura 5.11, en cambio, se muestran los coeficientes de importancia de las variables, pero respecto a la contribución en la predicción de una clase concreta. Las variables con coeficientes negativos (color rojo) contribuyen a la predicción de la clase negativa (no recurrencia de FA), mientras que las variables con coeficientes positivos (color azul) contribuyen a la predicción de la clase positiva (recurrencia de FA).

Este gráfico muestra la importancia de una variable para la predicción negativa o positiva de la clase, los coeficientes son los pesos otorgados por el algoritmo para crear la función del hiperplano separador y no tienen nada que ver con los valores que toman las variables en los ítems de entrenamiento.

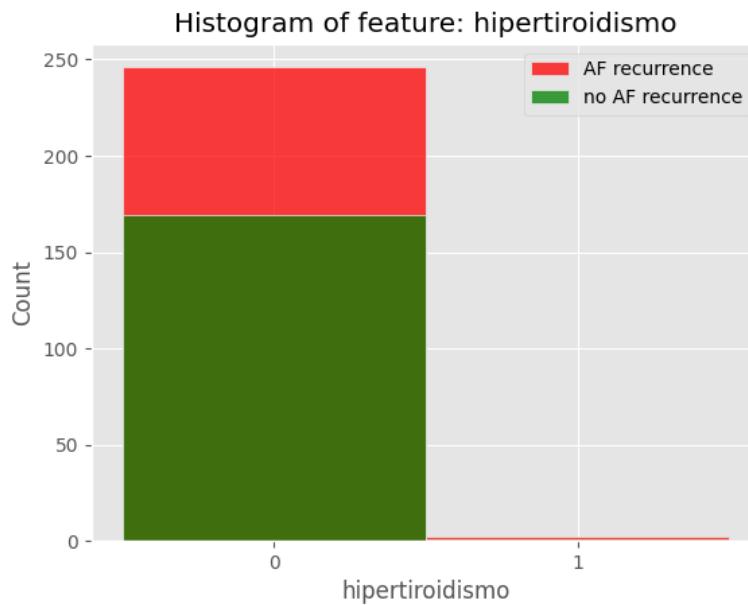


5.11. Figura: Coeficientes de importancia de las variables en un modelo *SVM* lineal

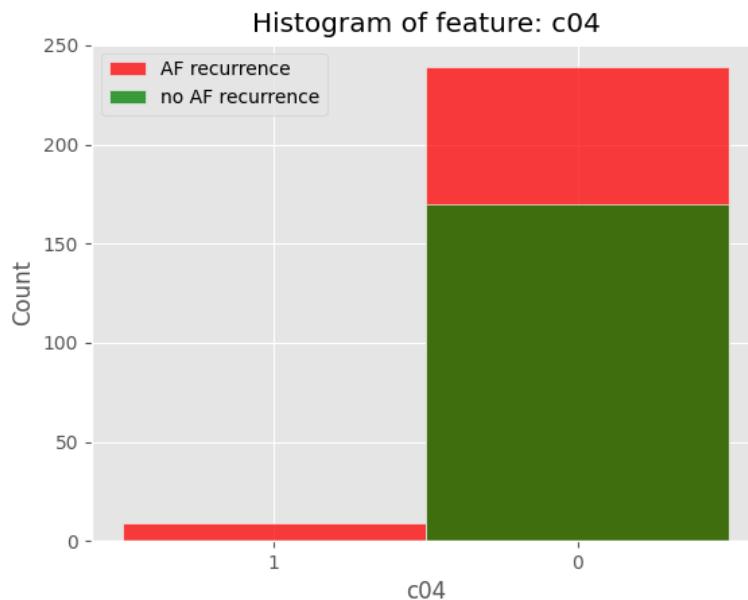
En el gráfico de la Figura 5.11, si analizamos las variables que más contribuyen a la predicción negativa (no recurrencia de FA), vemos que se encuentran *c01b* (farmacos antiarrítmicos C01B) y *ablation*. Desde el punto de vista médico, tiene sentido que esas variables tengan una gran influencia en la predicción de no recurrencia de fibrilación auricular, ya que se tratan de tratamientos y procedimientos realizados en la estrategia de control de ritmo cardiaco. Sin embargo, la aparición de diagnósticos como *demenzia*, *ansiedad* o *cardiopatia_isquemica* entre las variables más importantes para la predicción de no recurrencia de FA, no tiene por qué significar que los pacientes que hayan sufrido estas enfermedades tengan menos probabilidad de recurrencia, sino todo lo contrario, la no aparición de estas enfermedades puede favorecer a la vuelta del ritmo sinusal normal.

Entre las variables que más contribuyen a una predicción positiva (recurrencia de FA), nos encontramos con tales como *edad*, *pensionista*, *residenciado*, *genero*, *diametro_ai*, *area_ai* y *sahos*. Las tres primeras (edad, pensionista o no, e ingresado en una residencia o no) pueden tener gran influencia en la recurrencia de FA, ya que habitualmente se desarolla y permanece en personas de edad avanzada. Además el diámetro y área de la aurícula izquierda son buenos indicadores del desarrollo y permanencia de recurrencia de FA. El género está comprobado que influye en la aparición de la arritmia, y se ve que también es de gran interés para la predicción de recurrencia o permanencia. La enfermedad respiratoria SAOS también aparece como una de las variables predictoras más influyentes de recurrencia de FA.

Por último, la aparición de variables como *hipertiroidismo* o *c04* con tanta importancia puede deberse a que si miramos la distribución de estas mismas respecto a la clase en el conjunto de entrenamiento utilizado (ver Figuras 5.12 y 5.13), todos los ejemplos que contienen valor de 1 en esas variables (aunque sean muy pocos) han sufrido recurrencia de fibrilación auricular, y es por eso que el algoritmo le otorga tanta importancia, aun no siendo quizás desde el punto de vista clínico las variables más relevantes para la predicción de recurrencia de FA.



5.12. Figura: Distribución de la variable *hipertiroidismo*

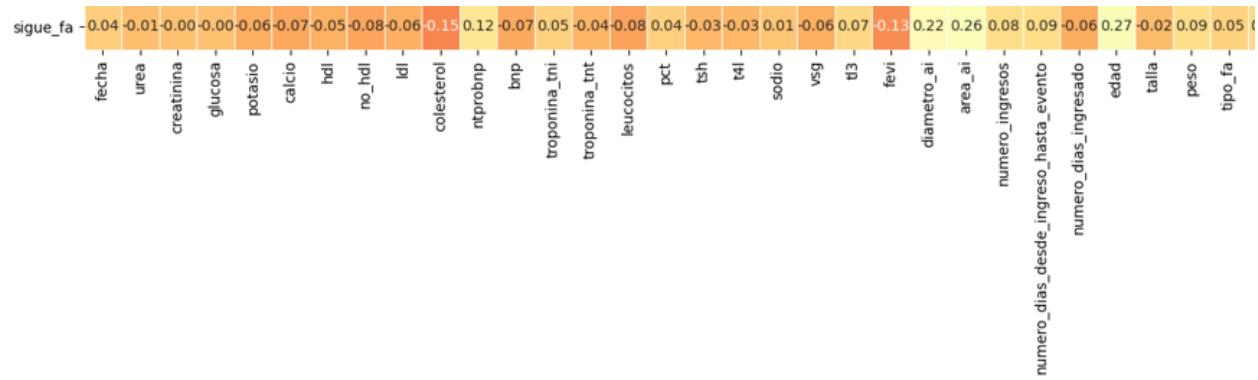


5.13. Figura: Distribución de la variable *c04*

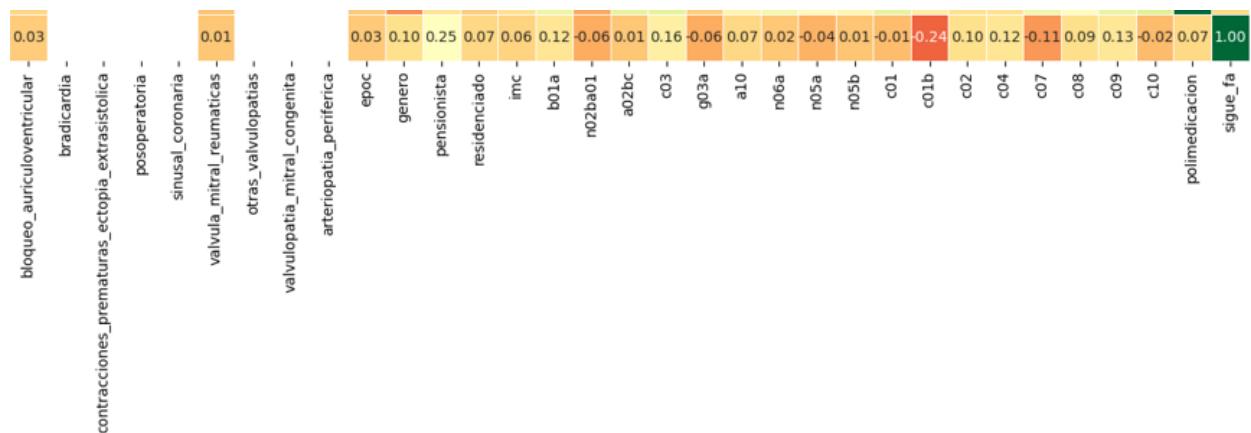
Por otro lado, si comparamos las variables seleccionadas con las de mayor importancia según la matriz de correlación entre pares de variables (*heatmap* de correlación), podemos ver que la mayoría de las variables utilizadas por el modelo son las que mayor correlación expresan con la clase *sigue_fa* en el *heatmap* (ver Figuras 5.14 y 5.15).

Las 5 variables con mayor correlación con la clase positiva (recurrencia de FA) según el *heatmap* son: *edad*, *area_ai*, *pensionista*, *diametro_ai* y *c09*. Todas de ellas han sido seleccionadas por *RFE* y son utilizadas en el modelo óptimo con el algoritmo *SVM*.

En cambio, las 5 variables con mayor correlación con la clase negativa (no recurrencia de FA) según el *heatmap* son: *c01b*, *ablation* (valor -0.21, no aparece en las imágenes), *colesterol*, *fevi* y *c07*. Todas menos *c07* han sido seleccionadas por *RFE* y son utilizadas en el modelo óptimo con el algoritmo *SVM*.



5.14. Figura: Métrica AUC respecto al número de variables seleccionadas (I)



5.15. Figura: Métrica AUC respecto al número de variables seleccionadas (II)

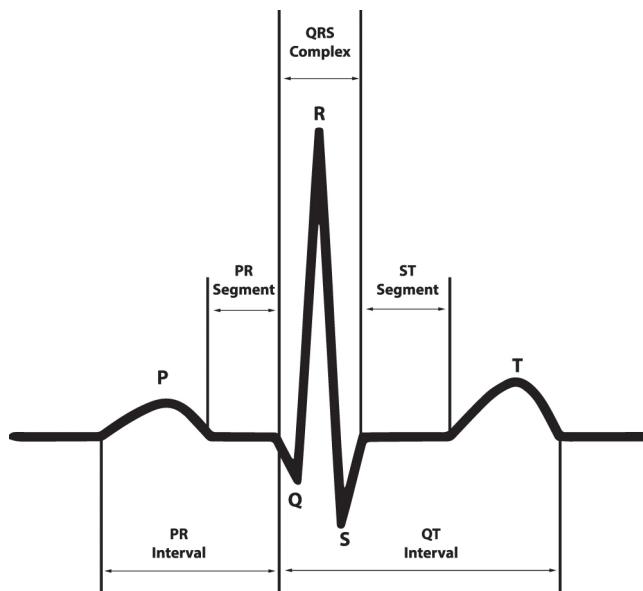
6. Tarea secundaria: PRAFAI en base a ECGs

6.1. Introducción y objetivos

Se ha definido una tarea secundaria relacionada con la tarea principal del Trabajo Fin de Grado, la cual complementa la idea del proyecto de investigación PRAFAI. Se trata de crear otro modelo predictivo de recurrencia o permanencia de FA en pacientes que ya han padecido un primer evento de esta arritmia, pero en este caso, utilizando la información de los propios electrocardiogramas de los pacientes, en vez de las características demográficas y la historia clínica electrónica de los pacientes.

Un electrocardiograma (ECG) es un gráfico de voltaje con respecto al tiempo que refleja las actividades eléctricas de la despolarización del músculo cardíaco seguida de la repolarización durante cada latido del corazón. El gráfico de ECG de un latido normal (ver Figura 6.1) consta de una secuencia de ondas, una onda P que presenta el proceso de despolarización auricular, un complejo QRS que denota el proceso de despolarización ventricular y una onda T que representa la repolarización ventricular. Otras porciones de la señal incluyen los intervalos PR, ST y QT.

Especificamente, un ECG de 12 derivaciones (12-lead ECG), el cual es el tipo de ECG que se va a tratar en esta tarea, proporciona una instantánea de las ondas desde 12 ángulos o vistas diferentes del corazón. Se coloca un electrodo en cada una de las cuatro extremidades y otros seis en localizaciones específicas del tórax, lo que hace un total de 10 electrodos en la piel del paciente. Son las derivaciones bipolares de los miembros I, II, III; las derivaciones aumentadas de los miembros aVR, aVL, aVF, y las derivaciones precordiales V1 a V6. Las derivaciones analizan una parte específica de la superficie del corazón y pueden ayudar a determinar qué parte del corazón ha sufrido una lesión, así como el origen y el flujo del impulso.



6.1. Figura: Ejemplo de onda de un ECG y los segmentos de la derivación II que presentan un ciclo cardíaco normal.

El objetivo ideal de este proyecto sería construir el modelo predictivo de recurrencia de FA añadiendo los ECGs de los 523 pacientes utilizados en el modelo ya realizado teniendo en cuenta las características demográficas e historia clínica electrónica de estos mismos. De esta manera, podría darse una predicción de recurrencia o vuelta a ritmo sinusal en pacientes tras un primer evento de FA desde dos modelos con perspectivas diferentes pero relacionadas entre sí: una en base a las características personales y clínicas del paciente, y otra en base a la información extraída de los propios ECGs que se le han realizado.

Sin embargo, el objetivo ideal de esta tarea no va a ser posible realizarlo en los plazos de este Trabajo Fin de Grado, ya que la obtención desde la OSI (Organización Sanitaria Integrada) del hospital con el que se está trabajando de estos ECGs en el formato apropiado para un trabajo de aprendizaje automático es complicada y lleva más tiempo que la duración de este TFG, tal y como ha informado el data-manager e investigador de la OSI del hospital implicado en este proyecto.

Por lo tanto, a la espera de poder realizar este modelo predictivo de recurrencia de FA añadiendo los ECGs de los 523 pacientes de la OSI del hospital tras un primer evento de FA, se ha decidido crear este mismo modelo predictivo con conjuntos de datos formados por ECGs ya utilizados en otras investigaciones similares.

En conclusión, en esta tarea secundaria se va a replicar una tarea de predicción de recurrencia de FA utilizando ECGs de conjuntos de datos disponibles (o clasificación de FA, en caso de no disponer de ECGs de pacientes tras un primer evento de FA), la cual será extrapolable a los ECGs de pacientes con FA de la OSI del hospital de la red de salud vasca cuando consigan obtenerse.

6.2. Estado del arte

En este apartado, se han recopilado bases de datos utilizadas y estudios realizados para la predicción o clasificación de arritmias utilizando la información extraída de electrocardiogramas. Tras este análisis se espera obtener el conjunto de datos adecuado y el conocimiento suficiente para la creación del modelo predictivo adaptado a nuestra tarea.

Este análisis se encuentra en el Anexo III de la memoria y se han analizados los siguientes artículos:

- **AFibNet: An Implementation of Atrial Fibrillation Detection with Convolutional Neural Network [70]**
- **Classification of 12-lead ECGs: the PhysioNet / Computing in Cardiology Challenge 2020 [2]**
- **JAMA Cardiology - Performance of a Convolutional Neural Network and Explainability Technique for 12-Lead Electrocardiogram Interpretation [27]**
- **Circulation: Deep Neural Networks Can Predict NewOnset Atrial Fibrillation From the 12-Lead ECG and Help Identify Those at Risk of Atrial Fibrillation–Related Stroke [57]**
- **Convolutional Neural Network and Rule-Based Algorithms for Classifying 12-lead ECGs [64]**

De este análisis se han sacado varias conclusiones. La primera es que la gran mayoría de herramientas de IA que trabajan con ECGs emplean algoritmos de Deep Learning como redes neuronales convolucionales (CNNs), ya que se trata de procesamiento de imágenes. Sin embargo, el uso de estas redes neuronales reduce la interpretabilidad del modelo. En el estudio *Hughes et al.* [27], por ejemplo, se aplica una técnica de explicabilidad llamada LIME para resaltar qué segmentos del ECG contribuyen a las predicciones según lo aprendido por la CNN. En otros estudios como *Raghunath et al.* [57], también se emplean como *baseline* otros algoritmos de Machine Learning (por ejemplo, *XGBoost*).

Por otro lado, en muchos de los estudios de IA para la clasificación de ECGs (un ejemplo es *Raghunath et al. [57]*), agregan variables predictoras adicionales como la edad y el sexo del paciente a las redes neuronales. El estudio *Natarajan et al. [50]* también añade características extraídas a mano de la derivación II del ECG, características relacionadas con la variabilidad de la frecuencia cardíaca y características morfológicas.

Por ello, en nuestra tarea, se tratará de crear dos modelos: un primer *baseline* y un modelo con técnicas de aprendizaje automático avanzado como puede ser una red neuronal convolucional (CNN). A estos modelos predictivos, entrenados con la información de los ECGs, se les podrían añadir características adicionales como la edad y el sexo del paciente.

En la siguiente sección se muestran las únicas bases de datos encontradas con ECGs de pacientes que parten de una fibrilación auricular y están creadas para la predicción de reversión de la arritmia.

6.2.1. Bases de datos: AF Termination Challenge Database y Long Term AF Database

Estos dos *datasets* están creados para tareas de predicción de reversión/terminación de fibrilación auricular, es decir, contienen ECGs de pacientes con FA persistente/recurrente y ECGs en los que la arritmia va a revertir.

6.2.1.1. AF Termination Challenge Database [45].

Base de datos con la que se tiene que predecir si la FA presente va a revertir a ritmo sinusal.

Características de cada registro (ECG):

- Duración: 1min
- Frecuencia de muestreo: 128Hz
- Muestras (*samples*): 7680
- Cada registro es un segmento de un ECG a largo plazo (20-24h)

El conjunto de entrenamiento está formado por 30 ECGs:

- 20 ECGs de los conjuntos S o T, en los que la FA revierte, no permanece. En los ECGs del conjunto S, la FA termina un minuto después del final del registro; mientras que en los del conjunto T, la FA termina inmediatamente después del final del registro (son inmediatamente posteriores a los registros del conjunto S, s01 es la continuaciób de s01).
- 10 ECGs del conjunto N, en los que la FA persiste/recurre. En los ECGs de este conjunto, la FA no ha terminado durante la duración del registro a largo plazo (20-24h), al menos una hora después del segmento.

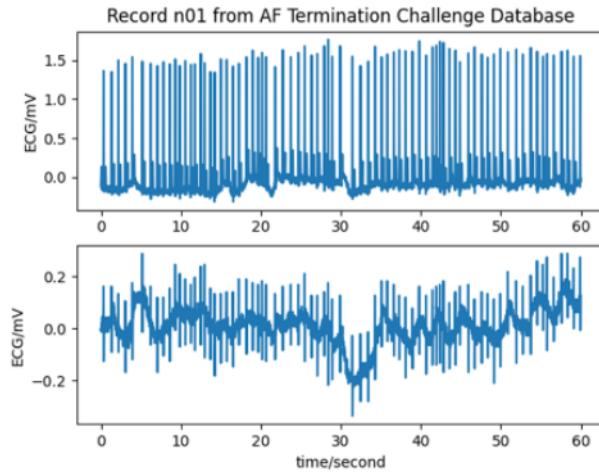
Los 30 ECGs se obtuvieron de 20 sujetos: 10 para el conjunto N y 10 para los conjuntos S y T.

Por otro lado, dispone de dos conjuntos de prueba. El conjunto de pruebas A está formado por 30 ECGs pertenecientes a 30 sujetos:

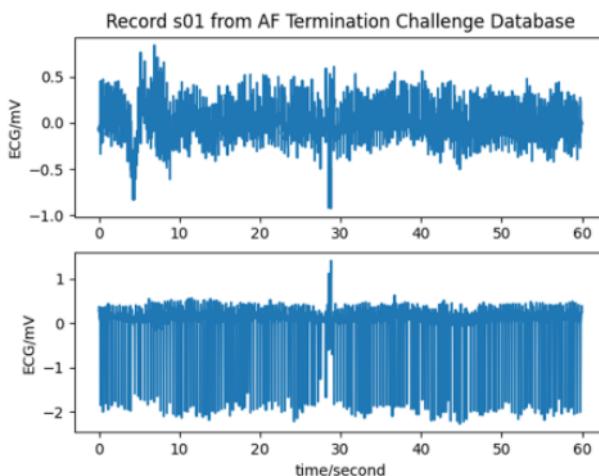
- Aproximadamente la mitad pertenecen al conjunto N (persistencia o recurrencia de FA).
- Los demás pertenecen al conjunto T (terminación de FA).

El conjunto de pruebas B esta formado por 20 ECGs pertenecientes a 10 sujetos:

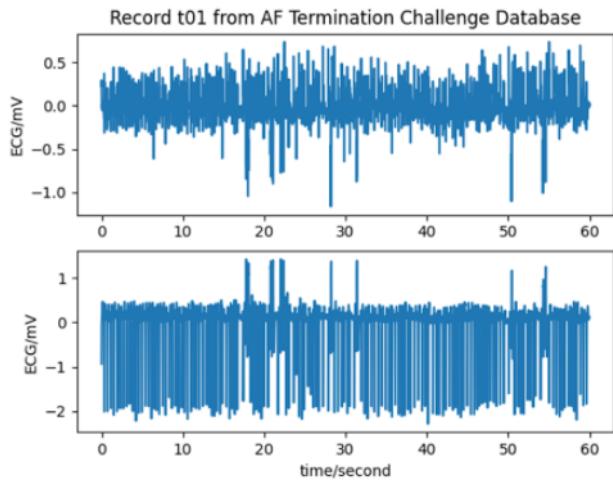
- 10 ECGs del conjunto S (terminación de FA un minuto después del final del registro).
- 10 ECGs del conjunto T (terminación inmediata de FA después del final del registro).



6.2. Figura: ECG de dos canales del conjunto N (persistencia/recurrencia de FA)



6.3. Figura: ECG de dos canales del conjunto S (terminación de FA un minuto después del final del registro)



6.4. Figura: ECG de dos canales del conjunto T (terminación inmediata de FA después del final del registro)

6.2.1.2. Long Term AF Database *Petrutiu et al.* [54].

Base de datos que incluye 84 registros de ECG a largo plazo con sujetos con FA paroxística y persistente.

Características de cada registro (ECG):

- Duración: largo plazo ($> 1h, > 2h\dots$)
- Frecuencia de muestreo: 128Hz
- Muestras (*samples*): varía dependiendo de la duración del registro
- Cada registro es un segmento de un ECG a largo plazo (24-25h)

La base de datos anterior (*AF Termination Challenge Database*) estaba formada por 80 extractos de un minuto de un subconjunto de los registros de *Long Term AF Database* (los numerados del 00 al 75); es decir, son los mismos ECGs en los dos *datasets*, solo que en *AF Termination Challenge Database* son extractos de 1min.

6.3. Tareas previas al desarrollo de la solución

Como ya se ha mencionado en el apartado de objetivos, la idea principal de esta tarea secundaria era crear un modelo predictivo de recurrencia de FA utilizando la información extraída de los ECGs de los pacientes con un primer evento de FA de la OSI del hospital de la red de salud vasca con el que estamos trabajando.

Sin embargo, por el momento no están disponibles estos datos. A pesar de no disponer de los propios ECGs de la OSI del hospital, si se conoce el tipo de electrocardiograma con el que trabajan. Se tratan de ECGs de 12 derivaciones realizados por sistemas MUSE de la empresa General Electric (GE), los cuales se exportan de este sistema en formato MUSE-XML.

Por lo tanto, antes de comenzar con el desarrollo de este proyecto, son necesarias dos tareas previas. La primera tarea consiste en encontrar una herramienta de conversión de 12-lead ECGs (electrocardiogramas de 12 derivaciones) en formato MUSE-XML. Y la segunda tarea consiste en obtener un conjunto de datos compatible con los ECGs que se reciben de la OSI del hospital.

6.3.1. Herramienta de conversión de ECGs en formato MUSE-XML

Los electrocardiogramas de 12 derivaciones de la OSI del hospital provienen de sistemas MUSE de GE. El formato en el que se obtienen estos ECGs es MUSE-XML. Sin embargo, para trabajar con esta información en una tarea de aprendizaje automático, es necesaria la conversión de los ECGs a formato CSV, con el cual obtendríamos una matriz con las muestras recogidas por cada una de las 12 derivación del ECG.

Para esta conversión de MUSE-XML a CSV se ha utilizado la herramienta **musexmlexport** de código abierto disponible en GitHub. Al tratarse de un script Python de código abierto se ha descargado y se han realizado algunos cambios para adecuarlos a nuestra tarea. Finalmente, se ha comprobado que la herramienta sí es útil para nuestro estudio al realizar la conversión con un 12-lead ECG en formato MUSE-XML de prueba.

	A	B	C	D	E	F	G	H	I	J	K	L
1	I	II	III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6
2	-24.40	24.40	48.80	-0.00	-36.60	36.60	-48.80	-141.52	-107.36	-48.80	-14.64	4.88
3	-24.40	24.40	48.80	-0.00	-36.60	36.60	-48.80	-146.40	-107.36	-48.80	-14.64	4.88
4	-29.28	24.40	53.68	2.44	-41.48	39.04	-48.80	-151.28	-112.24	-53.68	-19.52	0.00
5	-29.28	24.40	53.68	2.44	-41.48	39.04	-48.80	-151.28	-112.24	-53.68	-19.52	0.00
6	-29.28	24.40	53.68	2.44	-41.48	39.04	-48.80	-156.16	-112.24	-53.68	-19.52	0.00
7	-29.28	24.40	53.68	2.44	-41.48	39.04	-48.80	-161.04	-112.24	-53.68	-19.52	0.00
8	-29.28	24.40	53.68	2.44	-41.48	39.04	-48.80	-156.16	-112.24	-53.68	-19.52	0.00
9	-29.28	24.40	53.68	2.44	-41.48	39.04	-48.80	-156.16	-117.12	-53.68	-19.52	0.00
10	-29.28	24.40	53.68	2.44	-41.48	39.04	-48.80	-156.16	-117.12	-53.68	-19.52	0.00
11	-29.28	24.40	53.68	2.44	-41.48	39.04	-43.92	-156.16	-117.12	-53.68	-14.64	0.00
12	-29.28	24.40	53.68	2.44	-41.48	39.04	-43.92	-156.16	-117.12	-53.68	-14.64	4.88
13	-29.28	24.40	53.68	2.44	-41.48	39.04	-43.92	-156.16	-117.12	-53.68	-14.64	4.88
14	-34.16	39.04	73.20	-2.44	-53.68	56.12	-29.28	-156.16	-122.00	-39.04	0.00	29.28
15	-34.16	39.04	73.20	-2.44	-53.68	56.12	-29.28	-156.16	-122.00	-39.04	0.00	29.28
16	-34.16	39.04	73.20	-2.44	-53.68	56.12	-29.28	-156.16	-122.00	-39.04	0.00	29.28
17	-34.16	39.04	73.20	-2.44	-53.68	56.12	-29.28	-156.16	-122.00	-39.04	0.00	29.28
18	-34.16	39.04	73.20	-2.44	-53.68	56.12	-29.28	-156.16	-122.00	-39.04	0.00	29.28
19	-34.16	39.04	73.20	-2.44	-53.68	56.12	-29.28	-156.16	-122.00	-39.04	0.00	29.28
20	-34.16	39.04	73.20	-2.44	-53.68	56.12	-29.28	-156.16	-122.00	-39.04	0.00	29.28

6.5. Figura: 12-lead ECG convertido a formato CSV

En la Figura 6.5 se muestra el resultado obtenido al convertir un 12-lead ECG a formato CSV. Se obtiene una matriz de 12 columnas, cada una de ellas correspondiente a una derivación del ECG: I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5 y V6. El número de filas de la matriz corresponde a la cantidad de muestras (*samples*) registradas de la señal del ECG. En este ECG de prueba, se ha utilizado una frecuencia de muestreo (*sample rate*) de 500Hz y se ha registrado la señal del ECG durante 10s, por lo que la matriz cuenta con 5000 valores por cada derivación (5000 filas).

Es necesario tener en cuenta estas características (frecuencia de muestreo y duración del registro) en el momento que se obtengan los ECGs de los pacientes con un primer evento de FA en la OSI del hospital.

6.3.2. Elección de un conjunto de datos compatible con los ECGs de la OSI del hospital

Nuestra tarea consiste en la predicción de recurrencia de FA en pacientes con un primer diagnóstico de esta arritmia. No es lo mismo que una tarea de clasificación en la que un algoritmo debe detectar un ECG con fibrilación auricular o ritmo sinusal normal. En nuestra tarea todos los pacientes parten de una fibrilación auricular previa, por lo que los ECGs con los que entrenar el algoritmo ya muestran una FA paroxística o permanente.

Para nuestra tarea de predicción de recurrencia o reversión de la arritmia se han encontrado muy pocos conjuntos de datos públicos. Ejemplos de estos son las bases de datos *AF Termination Challenge Database* y *Long Term AF Database* descritas en la sección *Bases de datos* del Estado del arte. Sin embargo, estas bases de datos disponen de muy pocos ECGs, *AF Termination Challenge Database* solamente tiene 30 ítems para el conjunto de entrenamiento. Además, son ECGs de dos canales y su frecuencia de muestreo es de 128Hz durante 1 minuto. Teniendo en cuenta que buscamos ECGs de 12 derivaciones extraídos de sistemas MUSE, no se han considerado apropiados estos conjuntos de datos.

Por lo tanto, la base de datos más compatible con los ECGs que tratan en los sistemas MUSE de la OSI del hospital que se ha encontrado es *ECG recording from Chapman University and Shaoxing People's Hospital [79]*.

6.3.2.1. Base de datos: *ECG recording from Chapman University and Shaoxing People's Hospital [79]*

La base de datos consta de 10.646 ECGs de 12 derivaciones de diferentes pacientes, incluidos 5956 hombres y 4690 mujeres. Entre esos pacientes, el 17.15 % tenía ritmo sinusal normal y el 82.85 % tenía al menos una anomalía. Concretamente, hay 1826 ECGs etiquetados con ritmo sinusal normal (SR) y 1780 ECGs (16.72 %) etiquetados con fibrilación auricular (AFIB). La edad media de los pacientes etiquetados con SR es de 54.35 ± 16.33 años, mientras que la de los etiquetados con AFIB es de 73.36 ± 11.14 años. El 56.08 % de las personas con ritmo sinusal y el 58.48 % de los que sufren fibrilación auricular son hombres.

Los datos se adquirieron en cuatro etapas. Primero, cada sujeto se sometió a una prueba de ECG en reposo de 12 derivaciones que se tomó durante un período de 10 segundos. Los datos se almacenaron en el sistema de MUSE de GE. En segundo lugar, un médico con licencia marcó el ritmo y otras afecciones cardíacas. Otro médico con licencia realizó una validación secundaria. Si hubo un desacuerdo, un médico senior intervino y tomó una decisión final. En tercer lugar, los datos de ECG y la información de diagnóstico se exportaron desde el sistema GE MUSE a archivos XML que se codificaron con una conversión de nombres específica definida por General Electric (GE), MUSE-XML. Finalmente, se utilizó una herramienta de conversión para extraer datos de ECG e información de diagnóstico del archivo XML y transferirlos al formato CSV.

Por otro lado, todos los ECGs pasaron por un proceso de reducción de ruido ocasionado por la interferencia de la línea eléctrica, el ruido de contacto de los electrodos, los artefactos de movimiento, la contracción muscular, la desviación de la línea de base y el ruido aleatorio. Además, los ECGs se han registrado con una frecuencia de muestreo de 500Hz durante 10 segundos, por lo que tras la conversión a CSV se obtiene una matriz de 5000x12 con las señales de cada derivación (como en el 12-lead ECG de ejemplo de la Figura 6.5).

Por lo tanto, de esta base de datos obtenemos para nuestro estudio un total de 3606 ECGs de 12 derivaciones de diferentes pacientes (junto con características demográficas como la edad y el sexo), entre los que se encuentran 1826 ECGs que presentan ritmo sinusal normal y 1780 con fibrilación auricular. Han sido extraídos de sistemas MUSE como los utilizados en la OSI del hospital de la red de salud vasca con el que estamos trabajando y convertidos del formato MUSE-XML a CSV. Se ha utilizado una frecuencia de muestreo de 500Hz durante 10 segundos, por lo que se obtiene una matriz de 5000x12 teniendo en cuenta las 12 derivaciones.

6.3.2.2. Inconveniente

Los ECGs obtenidos de la base de datos *ECG recording from Chapman University and Shaoxing People's Hospital [79]* parten de pacientes con ritmo sinusal normal o muestran fibrilación auricular. Es decir, son datos útiles para una tarea de clasificación en la que el algoritmo debería clasificar los ECGs en ritmo sinusal o FA. Sin embargo, no es el tipo de ECG que buscamos para nuestra tarea de predicción de recurrencia de FA, ya que todos nuestros ECGs partirían de pacientes que ya han sufrido un primer evento de la arritmia.

Desde una perspectiva clínica, en un electrocardiograma con ritmo irregular, como con fibrilación auricular, no hay presencia de ondas P, la actividad auricular normal se ve sustituida por una actividad desorganizada, caótica, y la línea base del ECG aparece con ondas F (actividad rápida e irregular de diferente morfología). El ritmo de los QRS es totalmente irregular, con una distancia entre ellos variable.

En la Figura 6.1 de un ECG en ritmo sinusal puede apreciarse la onda P, la cual desaparece en un ECG con fibrilación auricular como el de la Figura 6.6. La onda pequeña que se ve detrás del QRS es la onda T y el trazado es muy irregular.



6.6. Figura: Ejemplo de registro ECG de un paciente en FA, en el que desaparece la onda P y la distancia entre los QRS varía.

Por lo tanto, volviendo a nuestra tarea de aprendizaje automático en la que el objetivo es predecir si pacientes tras un primer evento de FA van a recurrir en la arritmia o no, los ECGs de nuestros pacientes con los que entrenar el modelo predictivo mostrarían siempre un ritmo irregular (no hay onda P). Esto quiere decir que si entrenamos un modelo con la base de datos *ECG recording from Chapman University and Shaoxing People's Hospital*, no nos serviría posteriormente para aplicarlo a los ECGs de nuestra tarea, ya que en todos detectaría una irregularidad en el ritmo (por la ausencia de la onda P principalmente) y los clasificaría como fibrilación auricular (o recurrencia de FA).

Por esta razón crear un modelo de clasificación de FA con los datos *ECG recording from Chapman University and Shaoxing People's Hospital [79]* no sería extrapolable a nuestra tarea de predicción de recurrencia con ECGs de pacientes tras un primer evento de FA.

6.3.2.3. Solución propuesta

Tras ver que no disponemos de un conjunto de datos válido para nuestra tarea en concreto, se ha decidido hacer un **cambio de objetivo** a la espera de recibir en un futuro los ECGs necesarios de la OSI del hospital de la red de salud vasca.

Se va a crear un modelo de clasificación binaria (ritmo sinusal normal, *outcome* = 0; o fibrilación auricular, *outcome* = 1) en base a los 12-lead ECGs de la base de datos *ECG recording from Chapman University and Shaoxing People's Hospital [79]*. Para ello, se utilizarán los 1826 ECGs etiquetados con ritmo sinusal y los 1780 etiquetados con FA.

En definitiva, se dispone de un conjunto con un total de 3606 ítems. El nuevo objetivo de esta tarea será crear múltiples modelos de clasificación a medida que se va **disminuyendo la cantidad de datos de entrenamiento, hasta encontrar un empeoramiento relevante en los resultados de la evaluación del modelo**. De esta manera, se establecerá un umbral mínimo aproximado de ECGs necesarios para la creación del futuro modelo predictivo de recurrencia de FA. También **podrá utilizarse para la tarea de predicción de recurrencia futura el algoritmo empleado para la creación de este modelo de clasificación de ECGs**.

Es decir, el trabajo futuro consistiría en obtener el mínimo de ECGs necesarios de nuestra investigación de recurrencia de FA de la OSI del hospital y entrenar el modelo creado con los datos *ECG recording from Chapman University and Shaoxing People's Hospital [79]*, pero con nuestros propios ECGs. En ese caso, ya obtendríamos un modelo predictivo de recurrencia de fibrilación auricular en base a los electrocardiogramas, y no uno de clasificación de fibrilación auricular.

6.4. Diseño y desarrollo de la solución

Una vez finalizadas las dos tareas previas, se dispone de una base de datos con los ECGs apropiados para comenzar el diseño y desarrollo de la solución a la tarea. En esta sección van a diseñarse los modelos de IA necesarios para cumplir el objetivo de clasificación de ECGs, y va a desarrollarse el experimento de reducción de ítems en el conjunto de datos seleccionado.

Como se ha dicho en el apartado anterior, se va a crear un modelo clasificador de fibrilación auricular o ritmo sinusal normal utilizando los ECGs de la base de datos *ECG recording from Chapman University and Shaoxing People's Hospital* [79]. Posteriormente, se irá disminuyendo el número de ítems de entrenamiento hasta observar un notable descenso en los resultados de la evaluación del modelo.

Para comenzar, hay que decidir el algoritmo de aprendizaje supervisado que va a utilizarse para la creación del modelo. Teniendo en cuenta el análisis hecho para el estado del arte (ver sección 6.2), va a entrenarse un clasificador **XGBoost** como *baseline* y un modelo de Deep Learning mediante una **red neuronal**.

6.4.1. Baseline: **XGBoost**

Se han implementado diferentes modelos clasificadores de FA mediante el algoritmo *XGBoost*. Para ello, se han utilizado los 12-lead ECGs tras la reducción de ruido ofrecidos por la base de datos *ECG recording from Chapman University and Shaoxing People's Hospital* [79], y también la edad y el sexo de los pacientes pertenecientes a esos ECGs.

Se han creados modelos con los siguientes *inputs (features)*:

- **AS** → Inputs: **edad, sexo** (2 features)
- **RMS** → Inputs: **root mean square por cada lead** (12 features)
- **RMS-AS** → Inputs: **root mean square por cada lead + edad, sexo** (14 features)
- **I** → Inputs: **valores lead I** (5000 features)
- **I-AS** → Inputs: **valores lead I + edad, sexo** (5002 features)
- **II** → Inputs: **valores lead II** (5000 features)
- **II-AS** → Inputs: **valores lead II + edad, sexo** (5002 features)
- **III** → Inputs: **valores lead III** (5000 features)
- **III-AS** → Inputs: **valores lead III + edad, sexo** (5002 features)
- **aVR** → Inputs: **valores lead aVR** (5000 features)
- **aVR-AS** → Inputs: **valores lead aVR + edad, sexo** (5002 features)
- **aVL** → Inputs: **valores lead aVL** (5000 features)
- **aVL-AS** → Inputs: **valores lead aVL + edad, sexo** (5002 features)
- **aVF** → Inputs: **valores lead aVF** (5000 features)
- **aVF-AS** → Inputs: **valores lead aVF + edad, sexo** (5002 features)
- **V1** → Inputs: **valores lead V1** (5000 features)
- **V1-AS** → Inputs: **valores lead V1 + edad, sexo** (5002 features)
- **V2** → Inputs: **valores lead V2** (5000 features)
- **V2-AS** → Inputs: **valores lead V2 + edad, sexo** (5002 features)
- **V3** → Inputs: **valores lead V3** (5000 features)
- **V3-AS** → Inputs: **valores lead V3 + edad, sexo** (5002 features)
- **V4** → Inputs: **valores lead V4** (5000 features)
- **V4-AS** → Inputs: **valores lead V4 + edad, sexo** (5002 features)
- **V5** → Inputs: **valores lead V5** (5000 features)

- **V5-AS** → Inputs: **valores lead V5 + edad, sexo** (5002 features)
- **V6** → Inputs: **valores lead V6** (5000 features)
- **V6-AS** → Inputs: **valores lead V6 + edad, sexo** (5002 features)

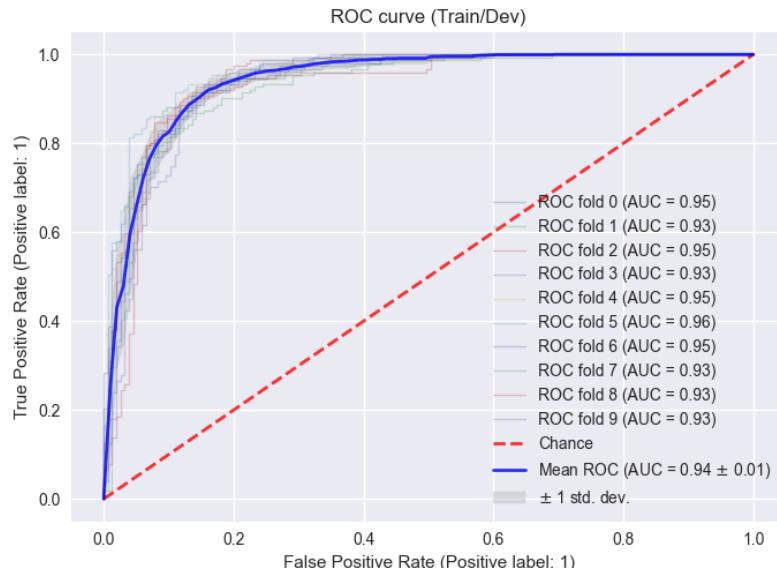
Se ha creado un primer modelo utilizando solamente la edad y el sexo de los pacientes (modelo **AS**). También se ha creado un modelo calculando la media cuadrática (*root mean square*) de las 5000 muestras recogidas por cada derivación (*lead*) del ECG (modelo **RMS**). Y por último, se han creado múltiples modelos por cada derivación con sus 5000 muestras (modelos **I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6**). En todos los modelos anteriores se ha creado una segunda versión añadiendo la edad y el sexo del paciente como dos *features* extra (modelos con terminación **-AS**).

Antes del proceso de entrenamiento del modelo, el conjunto de datos ha sido dividido en Train (70 %), Dev (15 %) y Test (15 %). El conjunto de entrenamiento ha sido estandarizado mediante *StandardScaler*, el cual escala los valores de cada variable restando la media aritmética y dividiéndola por la desviación estándar. Se aplica esa misma estandarización a los conjuntos Dev y Test.

Los modelos anteriores se han evaluado mediante el esquema *10-fold Cross Validation* sobre el conjunto de desarrollo/evaluación Dev. El modelo que mejores resultados ha obtenido ha sido **aVR**, modelo entrenado usando las 5000 muestras del **lead aVR** de los ECGs (sin incluir la edad ni el sexo del paciente). Se ha utilizado el área bajo la curva (AUC) como métrica para seleccionar el mejor modelo, y el modelo **aVR** ha obtenido un AUC de **0.94** (ver Cuadro 6.1 y Figura 6.7). Posteriormente, el modelo se ha probado sobre el conjunto Test.

aVR model	Dev	Test
AUC	0.94	0.93
Accuracy	0.88	0.86
F-Score	0.88	0.86

6.1. Cuadro: Evaluación del modelo *aVR* sobre los conjuntos Dev y Test



6.7. Figura: Curva ROC y AUC del modelo *aVR* sobre el conjunto Dev

6.4.2. Red neuronal

Al disponer de bastantes datos y de tipo numérico como son las muestras recogidas de las señales del 12-lead ECG, trabajar con modelos de Deep Learning puede ofrecer grandes ventajas. Es por ello que se

van a utilizar redes neuronales complejas.

Se han implementado diferentes modelos clasificadores de FA utilizando las CNNs creadas y empleadas en uno de los trabajos de la competición de clasificación de 12-lead ECGs de PhysioNet en 2020 [2] realizado por el **equipo TeamUIO** (*Singstad et al. [64]*, analizado *esta sección del Anexo III*). En este estudio se clasificaban 27 arritmias cardíacas basadas en los EGCs ofrecidos por la competición de PhysioNet, entre las que se encuentra la fibrilación auricular. Para nuestro estudio, vamos a aplicar su algoritmo sobre los datos *ECG recording from Chapman University and Shaoxing People's Hospital* [79] con ECGs con FA y ritmo sinusal.

Se han creado los siguientes 6 modelos, con los algoritmos implementados por el equipo TeamUIO y disponibles mediante licencia Apache 2.0 en su GitHub:

- **FCN** (Fully Convolutional Neural Network)

Como *input* solamente se utilizan los datos de los **12-lead ECGs**: matriz de 5000x12 por cada ECG (5000 muestras por cada lead). El código se encuentra [aquí](#).

- **FCN + MLP (edad, sexo)**

Como *input* se utilizan los datos de los **12-lead ECGs**: matriz de 5000x12 por cada ECG (5000 valores por cada lead); y **la edad y el sexo** del paciente del ECG a través de un perceptrón multicapa integrado en la red neuronal FCN. El código se encuentra [aquí](#).

- **Encoder** (Encoder Network)

Como *input* solamente se utilizan los datos de los **12-lead ECGs**: matriz de 5000x12 por cada ECG (5000 muestras por cada lead). El código se encuentra [aquí](#).

- **Encoder + MLP (edad, sexo)**

Como *input* se utilizan los datos de los **12-lead ECGs**: matriz de 5000x12 por cada ECG (5000 valores por cada lead); y **la edad y el sexo** del paciente del ECG a través de un perceptrón multicapa integrado en la red neuronal Encoder. El código se encuentra [aquí](#).

- **FCN + Encoder**

Como *input* solamente se utilizan los datos de los **12-lead ECGs**: matriz de 5000x12 por cada ECG (5000 muestras por cada lead). Se combinan las redes neuronales FCN y Encoder. El código se encuentra [aquí](#).

- **FCN + Encoder + MLP (edad, sexo)**

Como *input* se utilizan los datos de los **12-lead ECGs**: matriz de 5000x12 por cada ECG (5000 valores por cada lead); y **la edad y el sexo** del paciente del ECG a través de un perceptrón multicapa integrado en la red neuronal combinada FCN + Encoder. El código se encuentra [aquí](#).

Antes del entrenamiento de los modelos, de la misma manera que con el *baseline XGBoost*, el conjunto de datos se ha dividido en Train (70 %), Dev (15 %) y Test (15 %), y se ha estandarizado el conjunto de entrenamiento mediante *StandardScaler*. Además, para validar la CNN en cada iteración del entrenamiento (*epoch*), se ha utilizado el 20 % de los datos del subconjunto Train, creando el subconjunto Validation.

En todos estos modelos basados en CNNs, se ha utilizado un tamaño de lote (*batch size*) de 30 y 20 iteraciones de entrenamiento (*epochs*). Se ha aplicado un decremento de la tasa de aprendizaje (*learning rate*) mediante *reduce on plateau method* (factor de 10), en cada *epoch* que el AUC del conjunto de validación disminuye. Por último, se ha utilizado *early stopping*, cuando el AUC del conjunto de validación no mejora en dos *epochs* sucesivas termina el proceso de aprendizaje.

Los modelos anteriores se han evaluado mediante el esquema *Hold-Out* sobre el conjunto de desarrollo/evaluación Dev. El modelo que mejores resultados ha obtenido ha sido **FCN + MLP**. Al igual que con el *baseline XGBoost*, se ha utilizado la métrica AUC para seleccionar el mejor modelo, y el modelo **FCN + MLP** ha obtenido un AUC de **0.9926** (ver Cuadro 6.2).

	DEV			TEST		
	AUC	Accuracy	Loss	AUC	Accuracy	Loss
FCN	0.9885	0.9519	0.1412	0.9885	0.9527	0.1455
FCN + MLP	0.9926	0.9649	0.1095	0.9903	0.9612	0.1209
Encoder	0.9753	0.9057	0.2049	0.9687	0.9113	0.2320
Encoder + MLP	0.9693	0.8965	0.2710	0.9460	0.8762	0.3875
FCN + Encoder	0.9700	0.9150	0.2238	0.9560	0.9076	0.2755
FCN + Encoder + MLP	0.9746	0.9131	0.2186	0.9731	0.9131	0.2200

6.2. Cuadro: Evaluación de las redes neuronales sobre los conjuntos Dev y Test mediante *Hold-Out*

FCN + MLP es el modelo basado en una CNN que mejores resultados ha dado en esta tarea de clasificación de ECGs. En la evaluación anterior el modelo se ha validado mediante el esquema *Hold-Out* sobre un conjunto de desarrollo/evaluación que estaba formado por el 15 % de los datos. A continuación, para obtener unos resultados más robustos, se ha evaluado mediante *10-fold Cross Validation* utilizando el 85 % de los datos (Train + Dev) para ese proceso de entrenamiento/evaluación. Los resultados se mantienen tras esta última evaluación, el área bajo la curva (AUC) pasa de 0.9926 a **0.9917** (ver Cuadro 6.3).

	DEV		
	AUC	Accuracy	Loss
FCN + MLP	0.9917	0.9569	0.1232

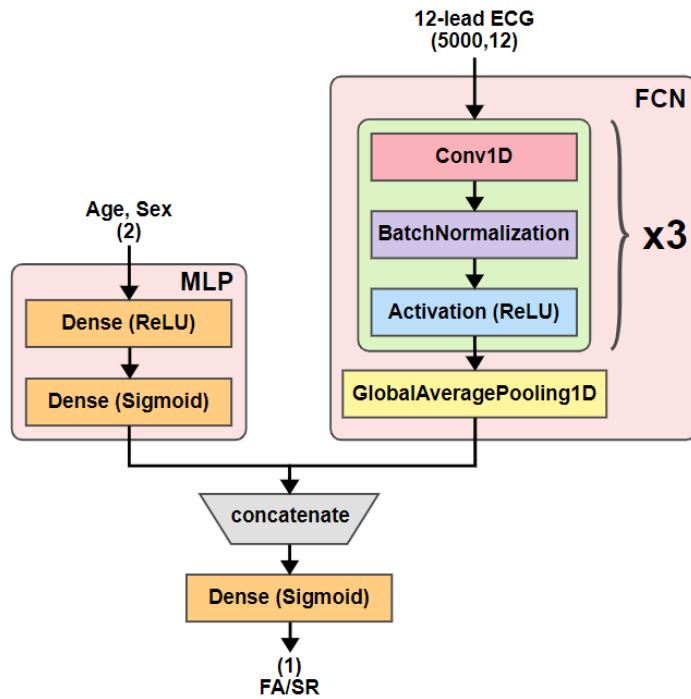
6.3. Cuadro: Evaluación del modelo FCN + MLP sobre los conjuntos Dev mediante *10-fold Cross Validation*

Por lo tanto, **FCN + MLP** es el modelo que se ha seleccionado para esta tarea de clasificación de fibrilación auricular o ritmo sinusal normal en base a la información de los 12-lead ECGs (y añadiendo la edad y el sexo de los pacientes). Como se puede observar, se ha mejorado el rendimiento del *baseline* con el que se obtenía un AUC de 0.94 sobre el conjunto de evaluación frente al 0.9917 que se consigue con esta CNN.

La arquitectura de este modelo FCN + MLP puede verse en las Figuras 6.8 y 6.9. Como ya se ha explicado, consta de dos redes neuronales, una FCN y un MLP, que se concatenan antes de la última capa. La FCN tiene una entrada de 5000x12 dimensiones que corresponde a las dimensiones de cada 12-lead ECG, mientras que el MLP tiene 2 dimensiones: la edad y el sexo del paciente.

La FCN esta formada por tres capas convolucionales de una dimensión (*Conv1D*), seguidas por capas de normalización (*BatchNormalization*) y activación (*Activation*) mediante la función ReLU. Finalmente, la FCN utiliza una capa de agrupación mediante el promedio global (*GlobalAveragePooling1D*).

El MLP esta formado por una capa oculta de 50 nodos con función de activación ReLU y una capa de salida de dos nodos con función de activación sigmoidea.



6.8. Figura: Arquitectura de la CNN FCN + MLP (I)

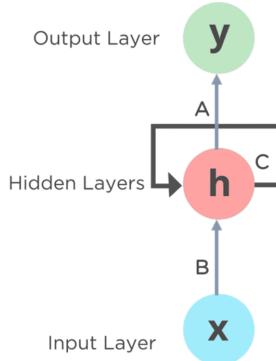
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 5000, 12)]	0	
conv1d (Conv1D)	(None, 5000, 128)	12416	input_1[0][0]
batch_normalization (BatchNorma	(None, 5000, 128)	512	conv1d[0][0]
activation (Activation)	(None, 5000, 128)	0	batch_normalization[0][0]
conv1d_1 (Conv1D)	(None, 5000, 256)	164096	activation[0][0]
batch_normalization_1 (BatchNor	(None, 5000, 256)	1024	conv1d_1[0][0]
activation_1 (Activation)	(None, 5000, 256)	0	batch_normalization_1[0][0]
conv1d_2 (Conv1D)	(None, 5000, 128)	98432	activation_1[0][0]
batch_normalization_2 (BatchNor	(None, 5000, 128)	512	conv1d_2[0][0]
input_2 (InputLayer)	[(None, 2)]	0	
activation_2 (Activation)	(None, 5000, 128)	0	batch_normalization_2[0][0]
dense (Dense)	(None, 50)	150	input_2[0][0]
global_average_pooling1d (Globa	(None, 128)	0	activation_2[0][0]
dense_1 (Dense)	(None, 2)	102	dense[0][0]
concatenate (Concatenate)	(None, 130)	0	global_average_pooling1d[0][0] dense_1[0][0]
dense_2 (Dense)	(None, 1)	131	concatenate[0][0]

6.9. Figura: Arquitectura de la CNN FCN + MLP (II)

6.4.2.1. Red neuronal recurrente (prueba)

Todas las pruebas realizadas en el apartado anterior han sido sobre redes neuronales convolucionales (CNNs). Sin embargo, proponer una red neuronal recurrente (RNN) para esta tarea de clasificación de ECGs podría ser una buena idea. Las RNNs están diseñadas para procesar datos secuenciales y reconocer patrones. Se le llama recurrente porque es capaz de hacer de manera continua la misma tarea u operación en una secuencia de entradas. Esto no se puede realizar con una CNN porque no pueden clasificar la correlación entre una entrada anterior y la siguiente.

La función de activación de las redes neuronales *feedforward* solo actúa en una dirección, hacia delante, desde la capa de entrada hacia la capa de salida, es decir, no recuerdan valores previos. Una red RNN es parecida, pero incluye conexiones que apuntan ‘hacia atrás’, una especie de retroalimentaciones entre las neuronas dentro de las capas. Como se puede ver en la Figura 6.10, la neurona recurrente h , en cada instante de tiempo (también llamado *timestep* en este contexto), recibe la entrada x de la capa anterior, así como su propia salida del instante de tiempo anterior para generar su salida y .



6.10. Figura: Ejemplo neurona recurrente

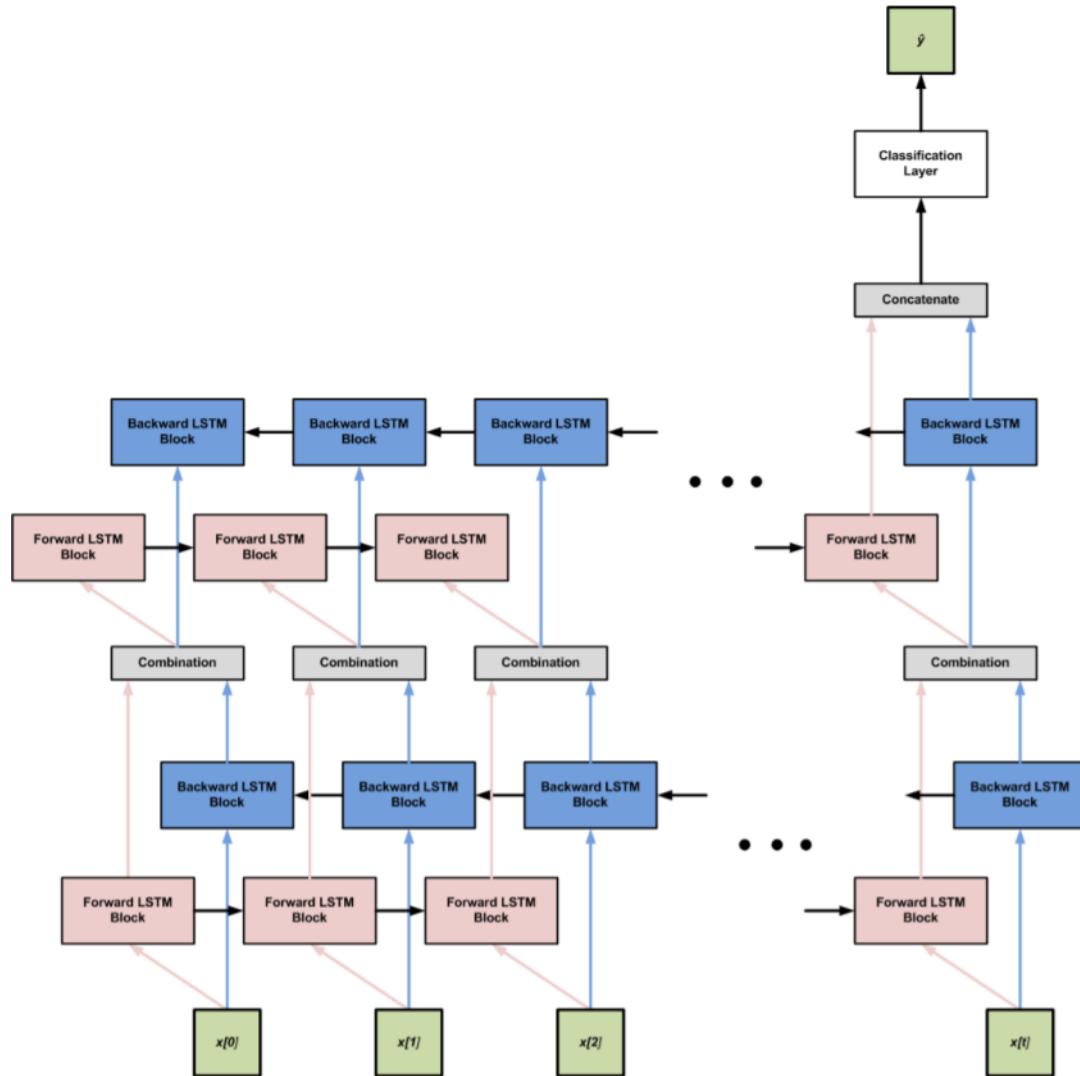
De esta manera, las neuronas recurrentes tienen en cuenta la información de estados anteriores para generar su salida, se podría decir que una neurona recurrente tiene en cierta forma memoria interna. Por ello, este tipo de redes son muy adecuadas para problemas de aprendizaje automático que involucran datos secuenciales. Gracias a su memoria interna, las RNN pueden recordar información relevante sobre la entrada que recibieron, lo que les permite ser más precisas en la predicción de lo que vendrá después manteniendo información de contexto.

Al tratarse un electrocardiograma de una secuencia de ondas en las que aparece una onda P seguida de un complejo QRS y una onda T (como se ha explicado en la sección 6.1), un ECG puede entenderse como un tipo de dato secuencial y utilizar una RNN para entrenar un clasificador podría ser útil. Los ECGs utilizados para nuestro estudio están formados por 5000 muestras de la secuencia de ondas, las cuales pueden entenderse como 5000 instantes de tiempo (*timesteps*), y esas muestras, para cada una de las 12 derivaciones del ECG.

El estudio *Classification of 12-Lead ECG Signals with Bi-directional LSTM Network [47]* utiliza una RNN para clasificar los ECGs de la base de datos disponible en *The China Physiological Signal Challenge 2018*, la cual está formada por 6877 ECGs de 12 derivaciones con diferentes arritmias entre las que se encuentra la fibrilación auricular. La RNN utilizada consta de dos capas LSTM bidireccionales y una capa convencional *fully connected* de clasificación para la salida (ver Figura 6.11).

Las neuronas recurrentes utilizadas en esta RNN son neuronas LSTM (Long Short-Term Memory) bidireccionales. Las neuronas LSTM pueden ‘recordar’ un dato relevante en la secuencia y preservarlo por varios instantes de tiempo, ya sea a corto plazo (como las neuronas recurrentes básicas) como también a largo plazo. Cada neurona bidireccional consta de un flujo hacia adelante (*forward*) y otro hacia atrás (*backward*), y en cada *timestep* se combinan las salidas de ambos flujos. La salida de la segunda capa oculta para ambos flujos en el último *timestep* se concatena y se envía a la capa de clasificación. En este estudio, la capa de clasificación utilizaba la función de activación *softmax* para obtener una salida final

de 9 dimensiones, ya que los ECGs se clasificaban entre 8 arritmias y ritmo normal. En nuestro caso, se sustituirá por una función de activación sigmoidea para clasificación binaria: FA o ritmo sinusal normal.



6.11. Figura: Arquitectura de la RNN con dos capas LSTM bidireccionales

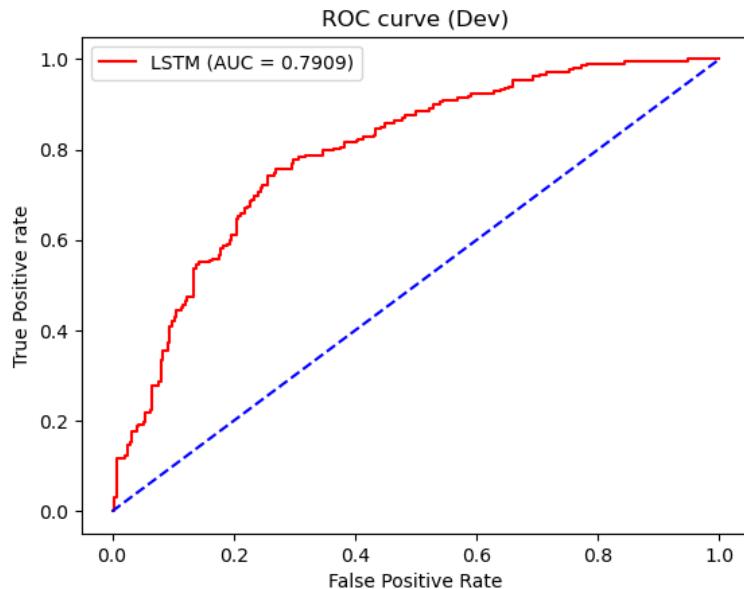
El modelo de este estudio demostró unos *F-scores* de 0.7388 y 0.7681 en la clasificación de ECGs con ritmo sinusal normal y fibrilación auricular, respectivamente.

Para nuestro estudio, vamos a aplicar su algoritmo sobre los datos *ECG recording from Chapman University and Shaoxing People's Hospital* [79] con ECGs con FA y ritmo sinusal. Al igual que en los experimentos con las CNNs, los datos se han dividido en Train (70 %), Dev (15 %) y Test (15 %) y se han estandarizado mediante *StandardScaler*.

La evaluación del modelo RNN sobre el conjunto Dev (ver Cuadro 6.4 y Figura 6.12) muestra un AUC de 0.79 y un *F-score* de 0.71, menor al obtenido en el estudio anterior. Por otro lado, comparando la RNN con las CNNs entrenadas en la sección 6.4.2, se ve que estas últimas demuestran un mayor rendimiento para la clasificación de ECGs. En la CNN seleccionada con mejor evaluación sobre el conjunto Dev (FCN + MLP), se obtenía un AUC de 0.9917, mucho mayor al 0.7909 de la RNN.

	DEV			
	AUC	Accuracy	Loss	F-score
RNN (LSTM)	0.79	0.71	0.57	0.71

6.4. Cuadro: Evaluación del modelo RNN (LSTM) sobre el conjunto Dev



6.12. Figura: Curva ROC y AUC del modelo RNN (LSTM) sobre el conjunto Dev

Como conclusión, aunque una RNN podía parecer a priori adecuada para tratar con ECGs, los cuales se basan en muestras de ondas secuenciales, las CNNs han demostrado mayor rendimiento para esta tarea de clasificación binaria de fibrilación auricular o ritmo sinusal normal. FCN + MLP ha sido finalmente el modelo escogido para el experimento realizado a continuación.

6.4.3. Experimento: reducción de ítems

Una vez seleccionado el modelo clasificador de esta tarea (red neuronal convolucional **FCN + MLP**), se ha realizado el experimento de reducción de ítems del conjunto de datos, con el objetivo de establecer un umbral mínimo de ECGs con el que se mantengan unos buenos resultados.

El *dataset* inicial está formado por 3606 ítems: 1826 con ritmo sinusal normal y 1780 con fibrilación auricular. Este total de datos se ha dividido en los subconjuntos Train (70 %, 2524 ítems), Dev (15 %, 541 ítems) y Test (15 %, 541 ítems). Por lo que el experimento consiste en ir reduciendo el número de ítems de entrenamiento e ir evaluando los modelos sobre el mismo conjunto de desarrollo/evaluación Dev, y posteriormente, realizar una segunda evaluación sobre el conjunto de prueba Test. Por otro lado, en las instancias pertenecientes al conjunto Train se realiza una división Train/Validation (80 %/20 %) para la validación de la CNN en cada *epoch*.

Tras una primera prueba reduciendo el número de ítems de entrenamiento a la mitad (1262 ítems), se ha visto que los resultados prácticamente no varían. Es por ello que se va a prestar mayor atención cuando la cantidad de datos baje de 1000. Por lo tanto, se ha decidido realizar el experimento con las siguientes cantidades de ítems de entrenamiento: **1262 (50 %), 1000, 900, 800, 700, 600, 500, 400, 300, 200, 100 y 50**.

Por cada subconjunto seleccionado del total de datos inicial, se ha realizado un muestreo aleatorio estratificado, manteniendo la distribución de los datos. Por ejemplo, en la primera reducción con 1262

ítems de entrenamiento, se selecciona de la distribución original el 50% con ritmo sinusal normal y el 50% con fibrilación auricular.

El umbral mínimo de ítems para que el modelo siga siendo robusto se va a establecer en base a métricas de evaluación como el área bajo la curva (AUC) o la exactitud (*accuracy*) obtenida sobre el conjunto de evaluación Dev. Los resultados del experimento se muestran en el siguiente apartado.

6.5. Resultados

En este apartado se muestran los resultados obtenidos al entrenar la red neuronal FCN + MLP reduciendo el número de ítems de entrenamiento del conjunto de datos *ECG recording from Chapman University and Shaoxing People's Hospital [79]*, teniendo en cuenta los ECGs con FA y ritmo sinusal normal.

El objetivo de este experimento ha sido analizar de manera aproximada a partir de cuántos ítems comienzan a empeorar de manera significativa los resultados del modelo de clasificación. De esta manera, se podría llegar a tener una estimación del número de ECGs mínimos necesarios para la creación en un futuro de un modelo predictivo de recurrencia de FA utilizando los ECGs de la OSI del hospital de la red de salud vasca.

En el Cuadro 6.5 se muestran los resultados obtenidos en las métricas AUC, *accuracy* y *loss* (pérdida de entropía cruzada binaria, *binary cross-entropy loss*) para cada uno de los conjuntos de entrenamiento reducidos. La evaluación se ha realizado sobre el mismo conjunto Dev para todos los experimentos de reducción.

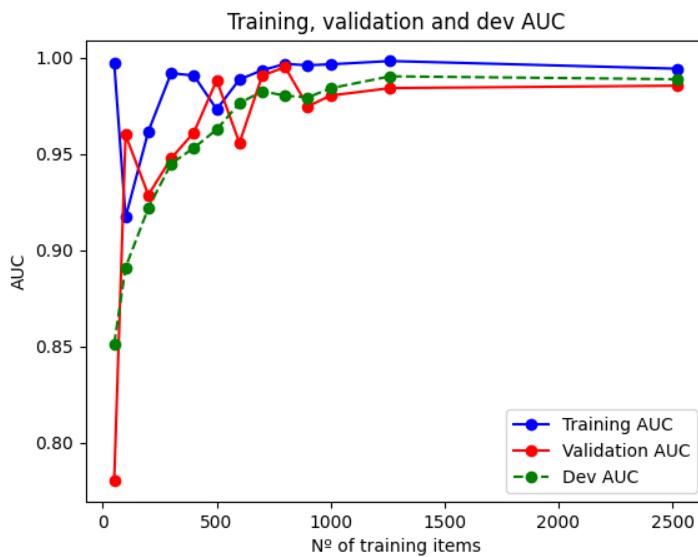
Nº ítems de entrenamiento	DEV		
	AUC	Accuracy	Loss
2524	0.9889	0.9427	0.1385
1262	0.9904	0.9612	0.1214
1000	0.9842	0.9057	0.2262
900	0.9794	0.9390	0.2037
800	0.9804	0.9390	0.1909
700	0.9827	0.9039	0.2642
600	0.9765	0.9057	0.2779
500	0.9628	0.9020	0.3042
400	0.9532	0.8595	0.4197
300	0.9450	0.8799	0.4334
200	0.9220	0.8022	0.5304
100	0.8911	0.8152	0.6115
50	0.8512	0.7098	0.6145

6.5. Cuadro: Evaluaciones de los modelos FCN + MLP al reducir el tamaño del conjunto de entrenamiento

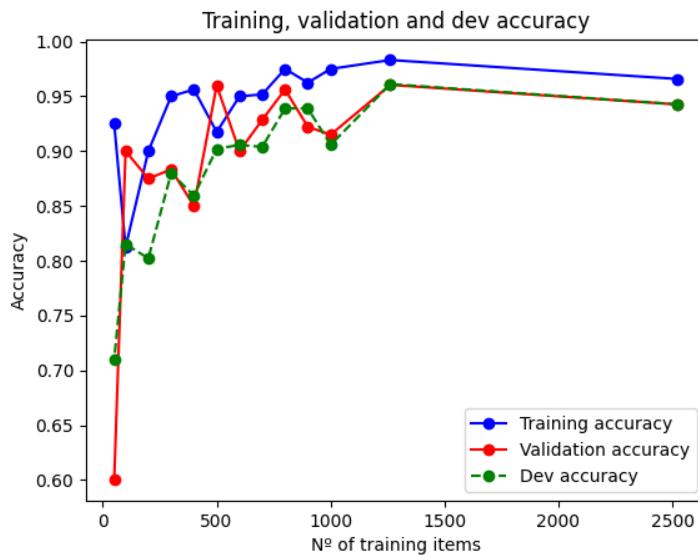
Observando las métricas sobre el conjunto de evaluación, hasta el entrenamiento con 600 ítems, el **área bajo la curva (AUC)** se mantiene prácticamente estable, variando entre 0.99 y 0.97. Es **al bajar de los 500 ítems**, cuando el AUC comienza a disminuir significativamente por cada reducción del conjunto de entrenamiento realizada. Este decremento progresivo del AUC a partir de los 500 ítems se puede observar mejor en el gráfico de la Figura 6.13. Aún así, se mantiene un AUC por encima del 0.9 hasta con solamente 200 ítems de entrenamiento (AUC = 0.92).

En cuanto a la **exactitud (accuracy)**, se mantiene mayor al 90% **por encima de los 500 ítems** de entrenamiento. Aunque con más de 1000 ítems se obtienen los mejores resultados con *accuracies* de 0.94 y 0.96, obtener un 90% de porcentaje de acierto en la clasificación de los ECGs del conjunto de evaluación sigue siendo muy buen resultado. En el gráfico de la Figura 6.14 se puede observar también el descenso de la *accuracy*.

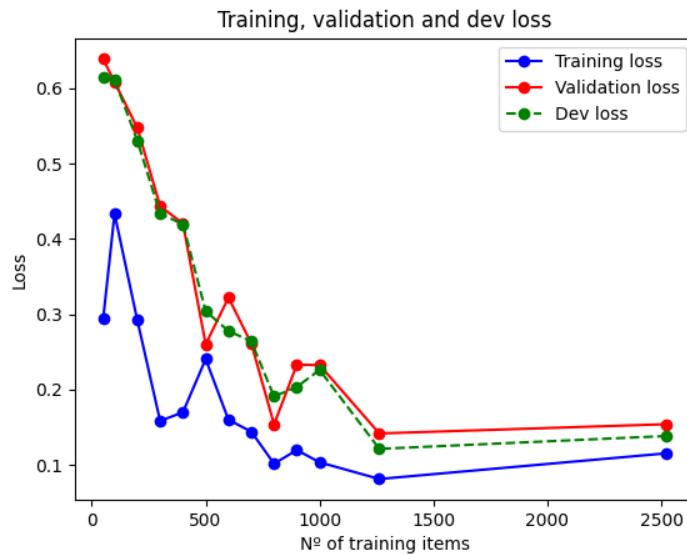
Respecto a la pérdida (*loss*), va aumentando más significativamente en cada reducción realizada, sobre todo, en las reducciones del conjunto de entrenamiento por debajo de los 700 ítems. En el gráfico de la Figura 6.15 se muestra la variación de la pérdida.



6.13. Figura: AUC respecto al número de ítems de entrenamiento



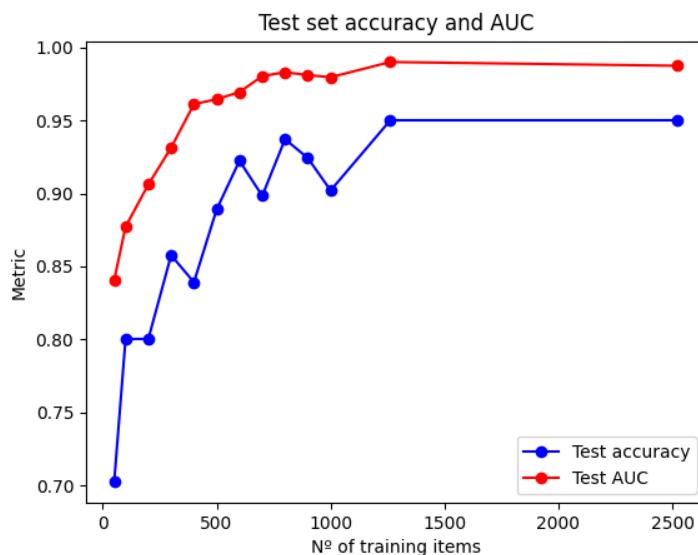
6.14. Figura: Accuracy respecto al número de ítems de entrenamiento



6.15. Figura: *Loss* respecto al número de ítems de entrenamiento

Por lo tanto, tras el análisis de la variación de las anteriores métricas sobre el conjunto de evaluación al reducir el número de ítems de entrenamiento, se puede llegar a la conclusión de que comienza a verse un empeoramiento más significativo y progresivo en los resultados al bajar de **alrededor de 500 ítems**.

Finalmente, también se han probado los modelos entrenados en cada reducción sobre el conjunto de pruebas Test (ver Figura 6.16). De la misma manera que en la evaluación con el conjunto Dev, el área bajo la curva (AUC) se mantiene estable hasta alrededor de los 500 ítems de entrenamiento, y al seguir reduciendo el conjunto, comienza a disminuir más significativamente. Respecto a la exactitud (*accuracy*), también se observa ese descenso por debajo del 0.9 al bajar de los 600-500 ítems de entrenamiento.



6.16. Figura: *Accuracy* y AUC respecto al número de ítems de entrenamiento sobre el conjunto Test

De cara a extrapolar las conclusiones obtenidas en este experimento al futuro trabajo de entrenamiento del modelo FCN + MLP con los 12-lead ECGs de pacientes con un primer evento de FA para la predicción de recurrencia de esta arritmia, un **mínimo aproximado de entre 500 y 600 ECGs** sería suficiente para entrenar un modelo robusto. Para establecer este umbral mínimo de ítems de entrenamiento se

han tenido en cuenta las métricas AUC y *accuracy*, las cuales se han considerado las más adecuadas para evaluar el rendimiento de una tarea de clasificación, y que se mantienen por encima de 0.96 y 0.90, respectivamente, en la evaluación realizada sobre el conjunto Dev.

Cabe resaltar que todos los resultados obtenidos en este experimento son estimaciones aproximadas para la futura tarea de predicción de recurrencia de FA, ya que los resultados pueden variar al entrenar el modelo con ECGs diferentes en los que el paciente parte de un primer diagnóstico de fibrilación auricular.

7. Conclusiones

El desarrollo del proyecto se ha realizado gracias al trabajo continuo (exceptuando el periodo de vacaciones) que ha permitido ir logrando los objetivos establecidos progresivamente. Se ha logrado crear una herramienta de predicción automática de recurrencia de fibrilación auricular basada en Inteligencia Artificial y entrenada con la historia clínica e información demográfica de pacientes tras un primer episodio de FA.

Sin embargo, pese a realizar una planificación previa calculando y analizando las tareas del proyecto y el tiempo a dedicar a cada una de ellas, al tratarse de un trabajo de investigación ‘real’ se han sufrido múltiples contratiempos.

Este último capítulo de la memoria se centra en el análisis final de los resultados y objetivos cumplidos del proyecto, así como los retrasos y complicaciones surgidas durante el desarrollo y una comparativa entre la planificación estimada y real. También se expone el futuro trabajo que puede presentar este proyecto y una reflexión final por parte del desarrollador principal Iker Valdelvira.

7.1. Análisis de los resultados y contratiempos

Se han cumplido los objetivos del proyecto en gran parte de manera satisfactoria:

- Se ha diseñado e implementado una herramienta basada en Inteligencia Artificial para la predicción de recurrencia de fibrilación auricular en un plazo aproximado de dos años en pacientes tras un primer episodio de la arritmia.
- Se ha creado una base de datos depurada y estructurada de pacientes con un primer evento de FA de la OSI del hospital de la red de salud vasca con el que se ha trabajado a partir de su historia clínica electrónica, la cual se ha utilizado para la creación del modelo predictivo mencionado anteriormente.
- Se ha hecho un análisis de los rasgos más influyentes en las predicciones de recurrencia de FA realizadas por el modelo predictivo diseñado.
- Se ha dado comienzo al proyecto de creación de otro modelo predictivo de recurrencia de FA utilizando la información de los propios electrocardiogramas de los pacientes. En la transcurso del proyecto no se han obtenido los ECGs de los pacientes tras un primer evento de FA de la OSI del hospital, por lo que la tarea se tuvo que modificar.

En cuanto a los resultados del modelo predictivo creado, se ha conseguido crear una herramienta capaz de predecir si un paciente que ya ha tenido un primer evento de fibrilación auricular puede recurrir en la arritmia con un **70 % de acierto** sobre el conjunto de prueba utilizado. De los 105 pacientes nunca antes vistos por el algoritmo en los que se ha probado la herramienta, 73 de las predicciones han sido correctas. 13 pacientes con recurrencia real han sido incorrectamente clasificados con reversión de la arritmia (falsos negativos), y 19 pacientes con reversión en un plazo aproximado de dos años han sido incorrectamente clasificados con recurrencia de la arritmia (falsos positivos).

Se trata de una alta tasa de acierto teniendo en cuenta el pequeño tamaño de la muestra que se disponía para realizar la investigación, un total de 523 pacientes de la OSI del hospital clasificados con recurrencia o no de FA tras un primer episodio.

También hay que tener en cuenta que se trata de un proyecto innovador ya que existen muy pocos estudios de predicción de recurrencia de FA basados en IA, y además, el nuestro considera una gran selección de rasgos acerca de la historia clínica electrónica de los pacientes.

Comparando los resultados obtenidos en nuestro proyecto con las investigaciones realizadas sobre *scores* clínicos acerca de la progresión de la arritmia en pacientes con fibrilación auricular (véase el artículo *Deng et al. [17]*), se observa que *scores* como ALARMEc [75], CHADS₂ y CHA₂DS₂-VASc [36] obtuvieron un área bajo la curva (AUC) de 0.657, 0.533 y 0.519, respectivamente, al ser evaluados sobre una muestra de pacientes con FA paroxística o FA no paroxística que se sometieron a ablación por catéter (AC) repetida. Otro estudio validó el *score* APPLE [38] en una cohorte bajo AC repetida, obteniendo un valor predictivo de recurrencia de FA con AUC de 0.617. **Nuestro estudio ha obtenido un AUC de 0.71 sobre nuestra muestra de prueba con pacientes tras un primer episodio de FA.**

Sin embargo, se demostró que el *score* MB-LATER [48], propuesto para predecir la recurrencia muy tardía (> 12 meses) de FA después de AC, obtuvo un mayor valor predictivo con AUC de 0.782 en una pequeña cohorte retrospectiva.

Por otro lado, en los *scores* mencionados en el anterior estudio, *Deng et al. [17]*, destacan la importancia de variables predictoras como el diámetro de la aurícula izquierda, el subtipo de FA, la recurrencia temprana, la edad y la insuficiencia cardiaca.

En nuestro estudio, que incluye muchos más rasgos clínicos que el resto de *scores* anteriores, son determinantes para la predicción de recurrencia de FA variables demográficas como la **edad**, el **género**, el ser **pensionista** o estar ingresado en una **residencia**, así como diagnósticos previos al primer evento de FA como la **ansiedad**, **demencia**, **SAOS**, **hipertiroidismo**, **cardiopatía isquémica** o **enfermedades de la válvula mitral reumáticas**. También son importantes los resultados de pruebas de laboratorio como el **colesterol** y la medicación con fármacos como **antipsicóticos (N05A)**, **benzodiacepinas (N05B)**, **fármacos del aparato cardiovascular (C01, C02, C04, C09, C10)**, y sobre todo, **antiarrítmicos de clase I o III (C01B)**, así como la **consumición de 5 o más fármacos** de los analizados. Por último, son importantes la **FEVI** (fracción de eyección ventricular izquierda) y el **diámetro y área de la aurícula izquierda**, así como la **ablación por catéter** tras el evento de FA.

Por último, respecto a la tarea secundaria con el objetivo de crear un mismo modelo predictivo de recurrencia de FA utilizando los electrocardiogramas (ECGs), ha tenido que modificarse la tarea al no disponer de los ECGs realizados a los pacientes de la OSI del hospital utilizados en esta investigación. No se debe olvidar que se trata de un proyecto de investigación en constante desarrollo, por lo que los contratiempos pueden estar presentes más frecuentemente.

Por ello, el objetivo de esta tarea secundaria ha cambiado en el transcurso del proyecto y se realizaron las siguientes tareas:

- Obtención de una herramienta para la conversión de ECGs de 12 derivaciones adquiridos de sistemas MUSE de GE (General Electric) a formato interpretable en Inteligencia Artificial.
- Obtención de una base de datos externa con ECGs similares a los de los pacientes utilizados en nuestro estudio. Finalmente, se ha obtenido la base de datos *ECG recording from Chapman University and Shaoxing People's Hospital [79]* creada para la clasificación de arritmias utilizando ECGs, y se ha realizado una tarea de clasificación de fibrilación auricular o ritmo sinusal normal.
- Realización de un experimento buscando el mejor algoritmo para la tarea de clasificación de ECGs, y después, reduciendo el número de ítems de entrenamiento con el objetivo de encontrar un umbral mínimo de ECGs de entrenamiento necesarios para crear un modelo robusto de clasificación de FA. Este algoritmo y umbral mínimo de ECGs serían extrapolables en cierto modo para la futura tarea de predicción de recurrencia de FA.

Tras este experimento, se ha llegado a la conclusión de que un algoritmo **FCN + MLP basado en una red neuronal convolucional (CNN)** utilizando la información de los 12-lead ECGs y la edad y el sexo de los pacientes es el más adecuado para la tarea de clasificación, y un **mínimo aproximado de entre 500 y 600 ECGs** sería suficiente para entrenar un modelo predictivo con un acierto (*accuracy*) mayor al 90 %. Estos serían el algoritmo a entrenar y el número de ECGs necesarios para la realización, en un futuro, de la herramienta de predicción de recurrencia de FA en base a los electrocardiogramas de los pacientes de la OSI del hospital.

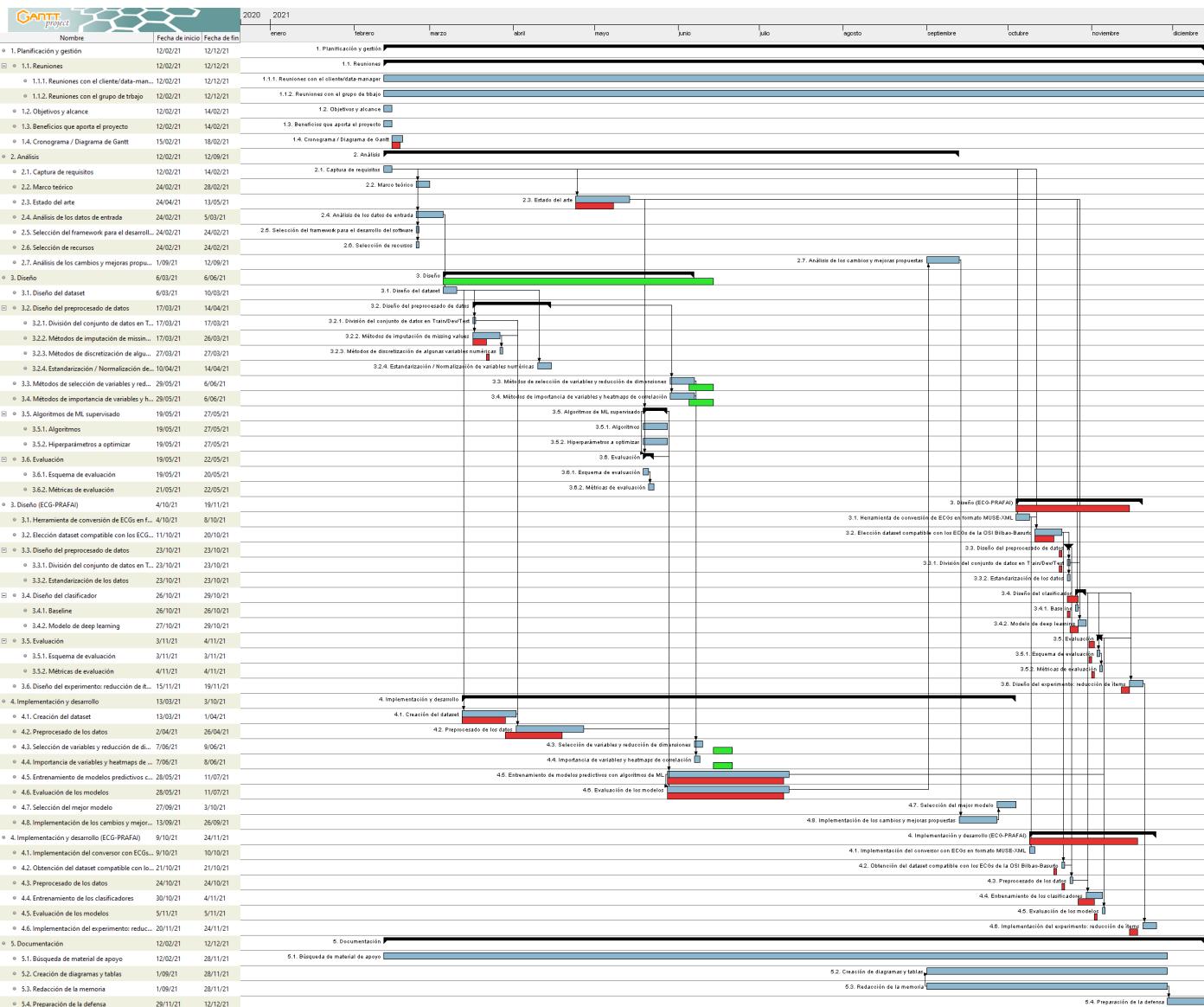
7.2. Análisis de la planificación estimada y real

Realizar y seguir una planificación exacta en un proyecto de investigación es muy difícil, más aún cuanto existen riesgos potenciales que pueden afectar en el desarrollo de un proyecto.

En la Figura 7.1 se muestra la evolución de la planificación estimada que se realizó al comienzo del proyecto y la real. En las tareas que se ha sufrido un atraso o un adelanto se muestran dos líneas de tiempo: la superior indica la duración y los plazos reales de la tarea, mientras que la inferior indica la duración y los plazos planificados. En los casos en los que se muestra una única línea de tiempo, no se ha sufrido ningún cambio respecto a la planificación inicial.

En cambio, algunas de las tareas se muestran en verde o rojo, indicando que han sido afectadas en el tiempo. Los dos colores, verde y rojo, muestran el cambio que han sufrido, ya sea un adelanto o un atraso, respectivamente. Como se puede observar, los adelantos se encuentran mayormente en las tareas de diseño, y los atrasos en las tareas de desarrollo e implementación, así como en la tarea secundaria ECG-PRAFAI. Como es habitual, la implementación del software suele tener una duración mayor a la estimada debido a la solución de errores surgidos durante las tareas de programación.

Por último, no se ha llevado un estricto control de horas por cada tarea, pero se estima un aumento por lo menos del 10 % en cuanto al esfuerzo del proyecto. Por lo tanto, si el esfuerzo estimado era de 620 horas, el esfuerzo real estará por encima de las 682 horas.



7.1. Figura: Comparativa de planificaciones en el diagrama de Gantt

7.3. Trabajo futuro

En este proyecto se ha creado un modelo predictivo de recurrencia de fibrilación auricular en un plazo aproximado de dos años utilizando la información recopilada de 523 pacientes la OSI de un hospital de la red de salud vasca que habían sufrido un primer episodio de la arritmia durante los años 2015-2018. Es un conjunto de datos muy pequeño para una tarea de Inteligencia Artificial y Machine Learning, por lo que la ampliación de esta base de datos en un futuro sería de gran ayuda a la hora de obtener un modelo predictivo más robusto y fiable. Cuanto mayor sea el conjunto de pacientes con un primer episodio de FA, más capacidad predictiva y discriminatoria adquiere el algoritmo entrenado a la hora de clasificar pacientes con o sin recurrencia de FA, y lo hace más competente en una población más heterogénea.

El uso de la Inteligencia Artificial en el ámbito clínico, y concretamente en este trabajo, en cardiología, es innovador. Herramientas como la implementada en este proyecto pueden servir de complemento a la toma de decisiones clínicas, y conocer las variables relacionadas con mayor riesgo de recurrencia de FA puede tener un impacto en la mejora de la salud de la población, lo que se denomina como medicina predictiva. Es por ello, que la herramienta de predicción podría ser utilizada por la OSI del hospital con el que se ha trabajado, y posteriormente, si se da el caso de ampliación de la base de datos con la que se entrena, podría ser implementada tanto en centros de la red de salud vasca como española.

Por otro lado, una ampliación del proyecto a medio plazo sería la creación del modelo predictivo de recurrencia de FA utilizando los ECGs de los pacientes estudiados pertenecientes a la OSI del hospital con el que se ha trabajado. Como ya se ha realizado la búsqueda de un algoritmo adecuado para trabajar en la clasificación de ECGs de 12 derivaciones y se ha establecido una cantidad mínima de datos con los que crear un modelo que ofrezca un buen rendimiento, solo faltaría obtener los electrocardiogramas de los pacientes necesarios para comenzar con el entrenamiento de la herramienta predictiva.

7.4. Reflexión personal

En lo referente a la reflexión personal, el comienzo del desarrollo del proyecto fue complicado y un tanto abrumador. Es la primera vez que se entraba a formar parte de un proyecto de investigación con una aplicación ‘real’ y altamente relacionada con un ámbito totalmente desconocido como es la medicina, y la cardiología en particular.

Es por ello que las primeras semanas resultó difícil adaptarse y comenzar a trabajar con convicción. Sin embargo, a medida que se adquirían los conocimientos necesarios y se iban realizando las tareas, la confianza con la que se trabajaba fue aumentando hasta lograr los objetivos principales establecidos al inicio. También cabe destacar que se ha logrado completar el proyecto gracias a la ayuda de los tutores y las explicaciones del cliente/data-manager en todo lo relacionado al ámbito clínico.

La experiencia y conocimientos adquiridos durante el transcurso de este proyecto se valoran de manera muy positiva, y se dan por terminados con satisfacción los estudios del Grado de Ingeniería Informática de Gestión y Sistema de Información con la finalización de esta memoria.

Anexo I:

Documento de requisitos

ID / Nombre	1 / Tipo de aprendizaje automático
Descripción	Es una tarea de aprendizaje supervisado. Se trata de clasificación binaria, en la que se deben clasificar los ejemplos de entrada como 0 (no hay recurrencia de FA) o 1 (sí hay recurrencia de FA).
ID en EDT	3.5. Algoritmos de ML supervisado

7.1. Cuadro: Requisito 1

ID / Nombre	2 / Datos de entrada y creación del <i>dataset</i> (input)
Descripción	La entrada es un conjunto de documentos en formato Excel. Un documento contiene el identificador de cada paciente a tratar como ejemplo, y la respectiva variable de salida (recurrencia o no de FA). El resto de documentos contienen los valores del vector de variables a construir para cada ejemplo. Se debe crear un conjunto de datos adecuado para aprendizaje automático donde cada ejemplo está formado por el vector de rasgos y la clase.
ID en EDT	3.1. Diseño del <i>dataset</i>

7.2. Cuadro: Requisito 2

ID / Nombre	3 / Rasgos (<i>features</i>)
Descripción	El vector de variables está compuesto por 100 rasgos (<i>features</i>) entre los que se encuentra el identificador de cada ejemplo (identificador del paciente) y la variable binaria de salida (recurrencia o no de FA). El resto de rasgos caracteriza cada ejemplo del conjunto de datos: variables demográficas (edad, género...), historia médica personal, historia cardiovascular específica, hospitalizaciones, pruebas de laboratorio, tratamientos y procedimientos.
ID en EDT	2.4. Análisis de los datos de entrada

7.3. Cuadro: Requisito 3

ID / Nombre	4 / Técnicas de preprocessado
Descripción	Se deben aplicar técnicas de preprocessado al conjunto de datos antes de entrenar un modelo predictivo. Se tendrán en cuenta, entre otras, la discretización de variables numéricas cuando sea necesario, la imputación de <i>missing values</i> para entrenar modelos mediante algoritmos que no admiten valores nulos, la división del conjunto de datos en subconjuntos Train/Dev/Test, y la estandarización o normalización de variables numéricas.
ID en EDT	3.2. Diseño del preprocessado de datos

7.4. Cuadro: Requisito 4

ID / Nombre	5 / Técnicas de selección de variables o reducción de dimensiones
Descripción	Se deben aplicar técnicas para la selección de las variables (<i>feature selection</i>) que más aportan al modelo, mejorando los resultados y reduciendo la complejidad del modelo. Se tendrán en cuenta, entre otras, técnicas mediante filtrados (<i>filters</i>) basados en la ganancia de información o en las medidas de correlación, y <i>wrappers</i> que dependen del modelo de clasificación ya que utilizan su desempeño como medida para determinar la relevancia de los rasgos. También se probará con métodos de reducción de dimensiones como PCA.
ID en EDT	3.3. Métodos de selección de variables y reducción de dimensiones

7.5. Cuadro: Requisito 5

ID / Nombre	6 / Técnicas de importancia y correlación de variables
Descripción	Se deben aplicar técnicas que muestren la importancia de cada variable a la hora de hacer las predicciones en un modelo entrenado, observando la influencia de los valores de cada variable en la predicción final. Para mostrar la correlación entre pares de variables se pueden crear matrices de correlación mediante <i>heatmaps</i> .
ID en EDT	3.4. Métodos de importancia de variables y <i>heatmaps</i> de correlación

7.6. Cuadro: Requisito 6

ID / Nombre	7 / Algoritmos de ML supervisado y selección del mejor modelo
Descripción	Se deben entrenar modelos predictivos con diferentes algoritmos de clasificación supervisada: <i>logistic regression</i> , <i>random forest</i> , <i>SVM</i> , <i>MLP</i> ... Y optimizar los valores de los hiperparámetros, dependiendo del algoritmo. Mediante las métricas de evaluación obtenidas tras el entrenamiento de los modelos, se seleccionará el modelo predictivo con mejores resultados para la predicción de recurrencia de FA.
ID en EDT	3.5. Algoritmos de ML supervisado 3.6. Evaluación 4.7. Selección del mejor modelo

7.7. Cuadro: Requisito 7

ID / Nombre	8 / Datos de entrada para ECG-PRAFAI
Descripción	La entrada debe ser un conjunto de ECGs de pacientes que han padecido un primer evento de FA. Estos ECGs son de 12 derivaciones (12-lead ECGs) y han sido extraído de sistemas MUSE, por lo que tienen formato MUSE-XML.
ID en EDT	2.4. Análisis de los datos de entrada (ECG-PRAFAI) 3.2. Elección <i>dataset</i> compatible con ECGs de la OSI (ECG-PRAFAI) 4.2. Obtención <i>dataset</i> compatible con ECGs de la OSI

7.8. Cuadro: Requisito 8

ID / Nombre	9 / Conversión de los ECGs a formato CSV
Descripción	Los ECGs de entrada están en formato MUSE-XML, y mediante una herramienta de conversión, deben convertirse a formato CSV legible para el posterior algoritmo de aprendizaje supervisado.
ID en EDT	(ECG-PRAFAI) 3.1. Herramienta de conversión ECGs en formato MUSE-XML (ECG-PRAFAI) 4.1. Implementación del conversor ECGs en formato MUSE-XML

7.9. Cuadro: Requisito 9

ID / Nombre	10 / Rasgos (<i>features</i>) de ECG-PRAFAI
Descripción	El vector de variables esta compuesto por 5000x12 valores correspondientes a la información de un ECG de 12 derivaciones con 5000 muestras recogidas de la señal del ECG a 500Hz durante 10 segundos. También pueden añadirse <i>features</i> del paciente como la edad y el sexo.
ID en EDT	2.4. Análisis de los datos de entrada (ECG-PRAFAI) 3.2. Elección <i>dataset</i> compatible con ECGs de la OSI (ECG-PRAFAI) 4.2. Obtención <i>dataset</i> compatible con ECGs de la OSI

7.10. Cuadro: Requisito 10

ID / Nombre	11 / Algoritmos para ECG-PRAFAI
Descripción	Se deben entrenar modelos predictivos de recurrencia de FA en base a los ECGs de pacientes con un primer evento de la arritmia. Se utilizarán algoritmos de <i>machine learning</i> para crear un <i>baseline</i> : XGBoost. Y redes neuronales complejas de <i>deep learning</i> : CNN.
ID en EDT	(ECG-PRAFAI) 3.4 Diseño del clasificador (ECG-PRAFAI) 3.5. Evaluación

7.11. Cuadro: Requisito 11

Anexo II: Últimos estudios relacionados con modelos predictivos de FA

Scores clínicos para los resultados del control del ritmo o la progresión de la arritmia en pacientes con fibrilación auricular: una revisión sistemática [17].

Fecha de publicación del artículo: 30/05/2017

El objetivo de este estudio de revisión es proporcionar una descripción general de los *scores* de riesgo clínico publicados relacionadas con los resultados de la estrategia de control del ritmo o la progresión de FA.

Se ha realizado una búsqueda exhaustiva mediante MEDLINE sobre los estudios que informan acerca de modelos predictivos de la recurrencia de FA después de la ablación por catéter (AC) o la progresión de FA. Se han estudiado nueve *scores* relevantes. Las predicciones se han evaluado con índices C o área bajo la curva (AUC), o índices de reclasificación o discriminación.

■ El *score* HATCH [16]

El *score* HATCH fue el primer modelo predictivo diseñado para la progresión de FA paroxística o persistente. En el estudio original, se incluyeron 1219 pacientes de la Euro Heart Survey y se observaron durante más de un año. Se utilizaron las siguientes variables predictoras: insuficiencia cardíaca (H), mayor edad (A), accidente isquémico transitorio previo (AIT) o ictus (T), enfermedad pulmonar obstructiva crónica (EPOC, C), e hipertensión (H). El *score* HATCH va de 0 a 7 puntos.

Según el modelo del estudio original, casi el 50 % de los pacientes con *score* > 5 progresaron a FA persistente, pero solo el 6 % de los pacientes con *score* 0 experimentaron esa progresión. El estudio posterior Kochhäuser *et al.* [33] sobre pacientes de FA que estaban esperando una AC determinó que el *score* HATCH tenía un pobre poder predictivo para la progresión de FA. Otro estudio, Potpara *et al.* [55], demostró que tenía un índice C de 0.6 en una cohorte de pacientes solitarios con FA durante un período de seguimiento de 12 años.

Más estudios para el análisis del valor predictivo de HATCH:

- Tang *et al.* [66]. Estudio en 488 pacientes con FA paroxística sometidos a AC. Después de un seguimiento de 27.4 ± 17.7 meses, el 69.93 % de los pacientes estaban libres de recurrencia de FA, pero el *score* HATCH no pudo predecir de manera confiable el resultado después de la AC.
- Suenari *et al.* [65]. Estudio que probó el *score* en una cohorte de 670.804 pacientes para investigar su valor predictivo de FA de nueva aparición. En esta cohorte, los pacientes tenían 20 años más que en la cohorte de derivación, sin antecedentes de FA. Durante un seguimiento

de $9,0 \pm 2,2$ años, los resultados mostraron que el *score* HATCH fue útil en la estimación y estratificación de la FA de nueva aparición.

- **El *score* ALARMEc [75]**

El objetivo en un principio era un *score* para predecir el resultado de FA tras AC en un estudio que comparó el *score* ALARMEc con los *scores* CHADS₂ y CHA₂DS₂-VASc para la estratificación del riesgo de accidente cerebrovascular.

ALARMEc incluyó cinco variables: tipo de FA, tamaño de la aurícula izquierda, insuficiencia renal, síndrome metabólico y miocardiopatía. El *score* ALARMEc se evaluó en solo 213 pacientes con FA paroxística o FA no paroxística que se sometieron a AC repetida. Después de un seguimiento de hasta 60 meses, solo la puntuación ALARMEc (AUC 0.657, P < 0,0001), pero no la puntuación CHADS₂ (AUC 0.533, P = 0.413) o CHA₂DS₂-VASc (AUC 0.519, P = 0.641), predijo los resultados después de AC.

Otro estudio, *Berkowitsch et al. [4]*, incluyó a 702 pacientes con FA y analizó cuatro de los cinco factores de riesgo en un *score* modificado. En este estudio, el síndrome metabólico y la insuficiencia renal fueron predictores independientes del resultado post-ablación de FA, pero el tipo de FA y el tamaño de la aurícula izquierda no fueron predictores. Para el resultado de la AC persistente, el síndrome metabólico no fue predictor, pero la insuficiencia renal siguió siendo un factor predictor significativo. El resultado de este estudio tuvo alguna diferencia con el anterior.

- **El *score* BASE-AF2 [7]**

El *score* BASE-AF2 ha utilizado seis variables: índice de masa corporal (IMC) $28kg/m^2$, dilatación auricular $> 40mm$, tabaquismo actual, recurrencia temprana, duración de la FA > 6 años y tipo de FA no paroxística.

Se utilizó una muestra de 236 pacientes con FA paroxística que se sometieron a crioablación. El *score* va de 0 a 6 puntos, teniendo en cuenta un *score* ≥ 3 puntos como predictor independiente de recurrencia de FA.

- **El *score* APPLE [38]**

El *score* APPLE se obtuvo a partir de una cohorte de pacientes con FA de Alemania y se probó en una cohorte de validación de los EEUU. Se han utilizado cinco variables: edad > 65 años, FA persistente, eGFR alterada ($< 60ml/min/1,73m^2$), diámetro de la AI $\geq 43mm$, y EF $< 50\%$.

Los análisis de regresión logística mostraron que los *score* APPLE, CHADS₂ y CHA₂DS₂-VASc fueron predictores significativos de recurrencia de FA entre 3 y 12 meses. Según el análisis de la curva ROC, el *score* APPLE tuvo un mejor valor predictivo en comparación con CHADS₂ y CHA₂DS₂-VASc (índice C de 0.634 frente a 0.538 y 0.542, respectivamente, ambos P < 0,001).

El estudio de validación del *score* APPLE también se llevó a cabo en una cohorte bajo AC repetida, como se recoge en el artículo *Kornej et al. [39]*. En comparación con la puntuación CHADS₂ y CHA₂DS₂-VASc, el *score* APPLE también tuvo un mejor valor predictivo de recurrencia de FA después de repetidas AC (AUC 0.617 frente a 0.577 y 0.590, respectivamente, ambos P < 0,001).

- **El *score* CAAP-AF [71]**

El *score* CAAP-AF se realizó inicialmente en una cohorte de derivación de 1125 pacientes con FA y se probó en una cohorte de validación de 937 pacientes con FA que se sometieron a la primera AC en el mismo centro. Se tuvieron en cuenta dos tipos de FA: paroxística y no paroxística. Se utilizaron las siguientes variables: arteriopatía coronaria (CAD), diámetro de la aurícula izquierda, edad, FA persistente o de larga duración, número de fármacos antiarrítmicos fallidos y sexo.

En la cohorte de validación, el porcentaje de pacientes sin FA fue idéntico al observado en la cohorte de derivación (índice C de 0.650 frente a 0.691). Es de destacar que el *score* CAAP-AF se basó en la experiencia de un solo centro y la recurrencia de la FA se detectó solo mediante Holter de 24h o 7 días, pero no mediante la implantación de un registrador de bucle.

■ El *score* MB-LATER [48]

El *score* MB-LATER se propuso para predecir la recurrencia muy tardía (> 12 meses) de FA después de la AC. Se utilizaron las siguientes variables: sexo masculino, bloqueo de rama, tamaño de la aurícula izquierda ≥ 47 mm, tipo de FA (paroxística, persistente o persistente de larga duración) y FA precoz recurrente.

El MB-LATER se derivó de una pequeña cohorte retrospectiva y se comparó con otras puntuaciones clínicas (APPLE, ALARMEc, BASE-AF2, CHADS₂, CHA₂DS₂-VASc o HATCH). En comparación con estas puntuaciones, el MB-LATER demostró un mejor valor predictivo (AUC de 0.782 frente a 0.716, 0.671, 0.648, 0.522, 0.519 o 0.583) y una mejor identificación de los pacientes con recurrencia de FA muy tardía mediante el análisis de la curva de decisión (DCA).

■ Los *scores* CHADS₂, CHA₂DS₂-VASc y R₂CHADS₂ [36]

Son *scores* de riesgo para predecir el ictus y la tromboembolia. Dado que los componentes de los factores de riesgo de estos *scores* son factores de riesgo cardiovascular comunes, pueden estar relacionados con los resultados del control del ritmo o la progresión de la arritmia.

Estas son las variables que utilizaron:

- CHADS₂: insuficiencia cardíaca congestiva, hipertensión, edad ≥ 75 años, diabetes mellitus y accidente cerebrovascular/accidente isquémico transitorio.
- CHA₂DS₂-VASc: insuficiencia cardíaca congestiva, hipertensión, edad ≥ 75 años, diabetes mellitus, accidente cerebrovascular/accidente isquémico transitorio, enfermedad vascular, 65-74 años y sexo femenino.
- R₂CHADS₂: disfunción renal, insuficiencia cardíaca congestiva, hipertensión, edad ≥ 75 años, diabetes mellitus y accidente cerebrovascular/accidente isquémico transitorio.

Según el estudio *Letsas et al.* [41], los *scores* CHADS₂ y CHA₂DS₂-VASc se asociaron con la recurrencia de la FA en pacientes con FA paroxística sometidos a un solo procedimiento de AC. CHADS₂ (AUC 0.644) y CHA₂DS₂-VASc (AUC 0.627) tuvieron el valor predictivo más alto para la recurrencia de FA.

Otro estudio, *Jacobs et al.* [29], que incluyó pacientes con FA paroxística y FA persistente, informó que los *scores* de CHADS₂ (HR 1.19, P <0.001) y CHA₂DS₂-VASc (HR 1.15, P <0.0001) fueron buenos para estratificar a los pacientes durante 5 años después de la AC, siendo el score CHA₂DS₂-VASc (HR 1.13, P = 0.001) superior al CHADS₂ para predecir la recurrencia de FA.

El estudio *Kornej et al.* [37] informó que el tipo de FA, el diámetro de la aurícula izquierda y la recurrencia temprana eran predictores significativos de la recurrencia a largo plazo después de la AC, y no los *scores* CHADS₂, 2DS₂-VASc y R₂CHADS₂. Los informes sobre el valor predictivo de estos *scores* de riesgo de accidente cerebrovascular en los resultados del control del ritmo no parecen tener resultados consistentes.

Conclusiones

- Cinco *scores* de diferentes cohortes incluyen el diámetro de la aurícula izquierda como variable predictora: ALARMEc, BASE-AF2, APPLE, CAAP-AF, y MB-LATER. Aun así, se trata de diferente manera según el modelo: el tamaño se definió como >43mm en el score APPLE, >40mm en el score BASE-AF2, >47mm en el score MB-LATER y >50mm en el score ALARMEc, y se estratificó en cinco categorías <40, (40,45], (45,50], (50,55] y ≥55mm en el score CAAP-AF.
- El subtipo de FA fue otra variable incluida en cinco *scores*. La FA no paroxística se ha asociado con tasas más bajas de terminación de FA y peores resultados después de la AC en comparación con la FA paroxística.
- La recurrencia temprana de FA solo se incluyó en los *scores* BASE-AF2 y MB-LATER. La recurrencia temprana se observa solo después de la AC y, aunque tiene un uso limitado en la decisión previa a la ablación, podría ser útil para predecir el resultado del ritmo después de procedimientos de ablación repetidos.
- El tamaño de la aurícula izquierda, el tipo de FA y la recurrencia temprana de FA deben considerarse como los parámetros más importantes incluidos dentro de los cinco sistemas de puntuación principales.
- La edad fue un factor de riesgo de los *scores* HATCH, CHADS₂, R₂CHA₂DS₂, CHA₂DS₂-VASc, APPLE y CAAP-AF. La insuficiencia cardíaca se utilizó en los scores HATCH, CHADS₂, R₂CHA₂DS₂, CHA₂DS₂ – VASc y CAAP-AF. La incidencia de insuficiencia cardiaca aumenta en pacientes con FA, lo que fomenta una estrategia de control del ritmo. Al contrario, los pacientes con insuficiencia cardiaca progresan a FA más fácilmente que los que no la padecen y el perfil de riesgo es compartido por insuficiencia cardiaca y FA.
- Solo el score BASE-AF2 utilizó la técnica de crioablación para AC como variable predictora, que podría haber tenido alguna influencia en el resultado de los pacientes con FA no paroxística. Los scores ALARMEc, APPLE, CAAP-AF, y MB-LATER utilizaron AC por radiofrecuencia.
- Los pacientes de estos estudios han tenido un seguimiento al menos de 12 meses. Para detectar la recurrencia de FA se utilizaron síntomas de arritmia con ECG de 12 derivaciones y monitorización de Holter continuo. Los seguimientos realizados a los pacientes en cada estudio tuvieron las siguientes duraciones:
 - HATCH: 27.4 ± 17.7 meses
 - ALARMEc: 24 meses
 - BASE-AF2: 20 meses
 - CHADS₂ / CHA₂DS₂-VASc: 16 meses
 - APPLE: ≥ 12 meses
 - CAAP-AF: 21.6 ± 1.6 meses
 - MB-LATER: 29 ± 10.1 meses
- Valoración de los resultados obtenidos:
 - Los *scores* CHADS₂, CHA₂DS₂-VASc y R₂CHADS₂, que no son predictores de riesgo relacionadas con el ritmo cardíaco, se probaron en diferentes cohortes con resultados contradictorios.
 - El *score* HATCH no tuvo un valor significativo en la predicción de la recurrencia después de AC.
 - En los estudios que describen ALARMEc y CAAP-AF, también se evaluó el valor predictivo de los *score* para los resultados repetidos de la AC. Las pruebas de validación de ALARMEc fueron contradictorias y CAAP-AF no fue validado por ninguna otra cohorte.
 - El *score* APPLE se ha validado en al menos tres cohortes (AUC de 0.634, 0.617 y 0.716, todas P < 0,05), incluida la cohorte de derivación del *score* MB-LATER. Los factores de riesgo de APPLE son índices clínicos de fácil adquisición, lo que hace que sea un buen predictor de la recurrencia de FA después de la AC.

- Para la recurrencia temprana basada en factores clínicos después del procedimiento de AC, los *scores* BASE-AF2 y MB-LATER podrían usarse.
- Se demostró que el *score* MB-LATER tiene un mejor valor predictivo para recurrencia muy tardía que otros *scores* en un estudio de cohorte prospectivo pequeño.

Evaluación de un modelo predictivo para el desarrollo de fibrilación auricular en un repositorio de historias médicas electrónicas (EMRs) [35].

Fecha de publicación del artículo: 01/12/2016

El objetivo de este estudio ha sido validar el modelo de predicción de riesgo de FA desarrollado por las Cohortes de Investigación sobre el Corazón y el Envejecimiento en Epidemiología Genómica: Fibrilación Auricular (CHARGE-AF) usando un amplio repositorio de historias médicas electrónicas (EMRs).

Para el estudio se ha utilizado una muestra de 33.494 personas de 40 años o más de edad con ninguna historia previa de FA. Todos estos pacientes tuvieron un seguimiento médico de FA desde el 31/12/2005 hasta el 31/12/2010 en las clínicas ambulatorias de medicina interna en el Centro Médico de la Universidad de Vanderbilt. Se ha utilizado un modelo de regresión de Cox. También se ha evaluado otra versión del mismo modelo, pero sin variables ecocardiográficas, con datos desde el 31/10/2013 hasta el 31/01/2014.

La salida del modelo es la incidencia de FA. Se han utilizado variables predictoras como edad, raza, altura, peso, presión arterial sistólica y diastólica, tratamiento para la hipertensión, tabaquismo, diabetes tipo 2, insuficiencia cardíaca, antecedentes de infarto de miocardio, hipertrofia ventricular izquierda e intervalo PR.

En cuanto a los resultados, la mediana de edad era de 57 años, el 57% de los pacientes era mujeres (43% hombres) y el 85.7% eran blancos y el 14.3% afroamericanos. Durante el seguimiento medio de 4.8 años, 2455 personas (7.3%) desarrollaron FA. Ambos modelos tenían un rendimiento deficiente en la cohorte de EMR, con una predicción insuficiente de FA entre las personas de bajo riesgo y una predicción excesiva de FA entre las personas de alto riesgo. El modelo CHARGE-AF tenía un índice C (Harrell's C-index) de 0.708 (95 % CI, 0.699-0.718) y el modelo simple de 0.709 (95 % CI, 0.699-0.718).

Este estudio destaca las dificultades de aplicar un modelo de riesgo derivado de estudios de cohortes prospectivos a una cohorte de EMR y sugiere que estos modelos de predicción de riesgo de FA se utilicen con precaución en el entorno de EMR.

Modelo predictivo de riesgo de fibrilación auricular incidente en la población japonesa general (The Hisayama Study) [23].

Fecha de publicación del artículo: 23/02/2021

El objetivo de este estudio ha sido desarrollar un modelo de predictivo de riesgo de FA incidente utilizando los datos longitudinales prospectivos de una población japonesa general.

Se ha utilizado una muestra de un total de 2442 residentes japoneses sin FA de 40 años o más de 1988 a 2012, siendo de 46.4 años la media de los participantes.

El desarrollo de FA se confirmó mediante un ECG estándar de 12 derivaciones en exámenes médicos repetidos y mediante registros médicos en clínicas u hospitalares. Para el modelo predictivo se ha utilizado un modelo de regresión de Cox. Se confirmaron 230 eventos de FA.

Se han tenido en cuenta variables como edad, sexo, presión arterial sistólica, circunferencia de la cintura, tasa de filtración glomerular estimada, soplo cardíaco anormal, amplitud alta de la onda R y otras arritmias.

En cuanto a los resultados, el modelo ha demostrado una buena discriminación (índice C de Harrell de 0.785) y calibración (test de Greenwood-Nam-D'Agostino, P=0.87) para FA en un plazo de 10 años.

El modelo ha demostrado un buen rendimiento en la población japonesa general y podría ser útil para determinar los requisitos para la evaluación de FA, como exámenes frecuentes de EGC en entornos clínicos y reducciones posteriores en el riesgo de complicaciones relacionadas con la FA.

Modelo predictivo para predecir la mortalidad en pacientes con fibrilación auricular (The BASIC-AF risk score) [60].

Fecha de publicación del artículo: 29/01/2021

El objetivo de este estudio ha sido desarrollar y validar un *score* de predicción de riesgo de mortalidad en pacientes con FA tras una hospitalización por razones cardíacas.

Este *score* ha derivado de una cohorte prospectiva de pacientes hospitalizados con FA concurrente. Las salidas del modelo son la mortalidad cardiovascular y por todas las causas. Se ha utilizado *random forest* para la selección de variables. Para el modelo predictor se ha usado un modelo de regresión de Cox y se ha validado internamente mediante *bootstrapping*.

En cuanto a la muestra utilizada, se incluyeron 1130 pacientes de FA. En una mediana de seguimiento de los pacientes de 2 años, 346 (30.6 %) murieron y 250 pacientes por una causa cardiovascular.

El péptido natriurético N-terminal pro-tipo B y la troponina-T de alta sensibilidad fueron las variables predictoras más importantes de mortalidad, seguidos por el volumen de la aurícula izquierda indexado, los antecedentes y el tipo de insuficiencia cardíaca, la edad, los antecedentes de diabetes mellitus y el retraso de la conducción intraventricular.

El modelo ha tenido un buen rendimiento: índice C de Harrell de 0.85 (95 % CI 0.82-0.88) en las muertes por todas las causas, e índice C de 0.84 (95 % CI 0.81-0.87) en las muertes cardiovasculares.

CHARGE-AF en una base de datos de EMRs de atención primaria de rutina en los Países Bajos: validación del riesgo de fibrilación auricular a 5 años e implicaciones para la selección de pacientes en el cribado de fibrilación auricular [25].

Fecha de publicación del artículo: 8/01/2021

El objetivo de este estudio ha sido validar el modelo de predicción de riesgo multivariable CHARGE-AF para el riesgo de FA a 5 años con datos de pacientes de atención primaria rutinaria, y evaluar el potencial del modelo para la selección automatizada y de bajo costo de pacientes con alto riesgo de FA basada esos datos.

Se ha utilizado una muestra de 111.475 pacientes libres de FA, con una edad media de 65.5 años (≥ 40) y siendo el 53% mujeres, desde el 01/01/2014 y con un seguimiento de 5 años. Se ha validado usando el índice C de Harrell para la discriminación, y gráfico de calibración y gráfico de Kaplan-Meier estratificado para calibración.

Hubo 5264 (4.7%) nuevos casos de FA durante el seguimiento de 5 años. El índice C fue de 0.74 (95% CI 0.73-0.74). El gráfico de calibración mostró una ligera subestimación del riesgo en los pacientes de bajo riesgo y una sobreestimación del riesgo de FA en aquellos con mayor riesgo previsto. La gráfica de Kaplan-Meier con categorías $< 2,5\%$, $2,5\% - 5\%$ y $> 5\%$ de riesgo predicho a 5 años fue muy precisa.

Se han utilizado como variables predictoras: CHA₂DS₂-VASc (insuficiencia o disfunción cardíaca, hipertensión, edad ≥ 75 [duplicado], diabetes, accidente cerebrovascular [duplicado] - enfermedad vascular, edad 65-74 y categoría de sexo [femenino]) y la edad sola.

Peso, altura, cambio de peso y riesgo de fibrilación auricular incidente en hombres y mujeres de mediana edad [31].

Fecha de publicación del artículo: 23/07/2020

El objetivo del estudio es descubrir las posibles asociaciones entre el peso, la altura y el cambio de peso con el riesgo de incidencia de FA en hombres y mujeres.

Se ha usado una muestra de 108.417 personas (el 51% mujeres) que participaron en un examen de salud basado en la población en el norte de Suecia a los 30, 40, 50 o 60 años de edad. En el examen de salud se midió el peso y la altura y se recopilaron datos sobre factores de riesgo cardiovascular. 40.275 participantes más se sometieron a dos reconocimientos médicos con un intervalo de 10 años. Se identificaron casos con un primer diagnóstico de FA a través del Registro Nacional de Pacientes de Suecia.

Durante un seguimiento, 5154 participantes desarrollaron FA incidente. La edad media en el momento de la inclusión fue de 46,3 años y la edad media en el momento del diagnóstico de FA fue de 66,6 años. La altura, el peso, el índice de masa corporal (IMC) y el área de superficie corporal (ASC) se asociaron positivamente con el riesgo de FA incidente tanto en hombres como en mujeres. Entre los participantes que se sometieron a dos exámenes médicos con 10 años de diferencia, 1142 personas desarrollaron FA. El cambio de peso medio fue un aumento del 4.8%. El aumento de peso o la pérdida de peso no se asoció significativamente con el riesgo de FA incidente.

Rendimiento de modelos predictores de riesgo de fibrilación auricular sobre 4 millones de personas [32].

Fecha de publicación del artículo: 14/01/2021

El objetivo de este estudio ha sido validar externamente un *score* FA de historia médica electrónica (EHR-AF) en IBM Explorys Life Sciences, un conjunto de datos multiinstitucional que contiene datos EHR estadísticamente anonimizados con más de 21 millones de personas (Explorys Dataset).

En la muestra se han incluido pacientes con riesgo total de FA, con más de dos consultas en 2 años, y pacientes sin FA prevalente. Se ha comparado EHR-AF con *scores* como CHARGE-AF, C₂HEST and CHA₂DS₂-VASc. Se ha evaluado la asociación entre los *scores* de riesgo de FA y la FA incidente a los 5 años, el accidente cerebrovascular y la insuficiencia cardíaca mediante un modelo de regresión Cox, la discriminación de FA a los 5 años mediante el índice C de Harrell y la calibración del riesgo de FA predicho con la incidencia de FA observada.

De 21.825.853 individuos en el conjunto de datos de Explorys, 4.508.180 comprendieron el análisis (edad media de 62.5 años y el 56.3 % mujeres). Los *scores* de FA se asociaron fuertemente con la FA incidente de 5 años, accidente cerebrovascular e insuficiencia cardíaca.

EHR-AF (índice C de 0.808 [95 % CI, 0.807-0.809]) demostró favorablemente la discriminación de FA comparado con CHARGE-AF (índice C de 0.806 [95 % CI, 0.805-0.807]), C₂HEST (índice C de 0.683 [95 % CI, 0.682-0.684]) y CHA₂DS₂-VASc (índice C de 0.720 [95 % CI, 0.719-0.722]). También demostró la mejor calibración para FA incidente. En los análisis de subgrupos, la discriminación de FA usando el modelo EHR-AF fue menor en personas con accidente cerebrovascular (índice C de 0.696 [95 % CI, 0.692-0.700]) e insuficiencia cardiaca (índice C de 0.621 [95 % CI, 0.617-0.625]).

EHR-AF ha demostrado un buen *accuracy* para FA incidente utilizando datos EHR fácilmente comprobados. El riesgo de FA se asocia con accidentes cerebrovasculares e insuficiencia cardíaca.

Desarrollo de un score de riesgo de fibrilación auricular en adultos con diabetes mellitus (ACCORD Study) [77].

Fecha de publicación del artículo: 20/06/2020

El objetivo del estudio ha sido desarrollar un *score* predictivo de riesgo de FA a 5 años para pacientes con diabetes mellitus (DM).

La muestra estudiada ha sido de 9.240 personas con DM (62 % hombres, edad media 62.6 años) de los cuales el 1.8 % desarrollaron FA durante una mediana de seguimiento de 4.9 años. Las personas que desarrollaron FA eran mayoritariamente hombres, de etnia blanca y con mayor obesidad y peor función renal, pero con presión arterial diastólica más baja y colesterol de lipoproteínas de baja densidad.

Se han tenido en cuenta variables predictoras como edad, sexo, raza, índice de masa corporal, insuficiencia cardíaca, presión arterial diastólica, triglicéridos, hemoglobina A1c, duración de la DM, creatinina sérica y medicación para la hipertensión.

El modelo predictor desarrollado es un modelo de regresión Cox con validación interna. Respecto a la evaluación, se ha demostrado un índice C de Harrell de 0.79 con una calibración interna excelente.

Modelos de predicción de fibrilación auricular aplicables en la comunidad: revisión sistemática y metaanálisis [25].

Fecha de publicación del artículo: 01/05/2020

El objetivo de este estudio ha sido determinar qué modelo puede ser el más adecuado para aumentar la eficiencia de los futuros esfuerzos de detección primaria de FA.

Se ha realizado una revisión sistemática de modelos multivariados derivados, validados y/o aumentados para la predicción de FA en cohortes comunitarias utilizando Pubmed, Embase y CINAHL (Cumulative Index to Nursing and Allied Health Literature) hasta el 1 de agosto de 2019.

Respecto a la muestra utilizada, se han incluido 27 estudios con un total de 2.978.659 participantes únicos entre 20 cohortes con una media de edad de 42 a 76 años. Y se han identificado 21 modelos de predicción de riesgo de FA.

Tres modelos mostraron una discriminación significativa a pesar de la alta heterogeneidad: CHARGE-AF (Cohorts for Heart and Aging Research in Genomic Epidemiology) (índice C de 0.71, 95 % CI 0.66-0.76), FHS-AF (Framingham Heart Study risk score for AF) (índice C de 0.70, 95 % CI 0.64-0.76), and CHA₂DS₂-VASc (índice C de 0.69, 95 % CI 0.64-0.74).

Solo CHARGE-AF, que comprende mediciones fácilmente obtenibles y elementos del historial médico, mostró una discriminación significativa entre las cohortes de predicción de riesgo de FA a 5 años.

Evaluación de un modelo de aprendizaje automático aplicado a datos de registros médicos electrónicos (EMRs) armonizados para la predicción de fibrilación auricular incidente [67].

Fecha de publicación del artículo: 03/01/2020

El objetivo de este estudio ha sido examinar los enfoques de aprendizaje automático aplicados a los datos de EMRs que se han armonizado con el Modelo de Datos Comunes de la Asociación de Resultados Médicos Observacionales para identificar el riesgo de FA.

Se ha usado una muestra de 2.252.219 personas atendidas en el sistema hospitalario UCHealth, que comprende 3 grandes hospitales en Colorado, desde el 01/01/2011 hasta el 01/10/2018. De los participantes el 54.4 % fueron mujeres y la edad media fue de 42.9 años (con desviación estándar de 22.3 años).

El modelo predictivo ha utilizado 200 variables predictoras de los EMRs, incluyendo diagnósticos, procedimientos, medicaciones edad y sexo. El modelo trata de clasificación de la FA incidente en intervalos designados de 6 meses, evaluado según el área bajo la curva AUC y el *F-score*. El algoritmo usado ha sido una red neuronal de una sola capa.

Respecto a los resultados, 28.036 (1.2 %) participantes desarrollaron FA incidente durante el intervalo de 6 meses designado. La evaluación del modelo determina un AUC de 0.800 y un *F-score* de 0.110. Este modelo funcionó solo ligeramente mejor que un modelo de regresión logística más básico compuesto por factores de riesgo clínicos conocidos para FA, que tenía un AUC de 0.794 y una *F-score* de 0.079.

Los modelos de aprendizaje automático para los datos de los EMRs ofrecen un método prometedor para mejorar la predicción del riesgo de FA incidente, pero se necesita más trabajo para mostrar una mejora más allá de los factores de riesgo estándar.

Anexo III:

Últimos estudios relacionados con modelos predictivos de FA en base a ECGs

AFibNet: An Implementation of Atrial Fibrillation Detection with Convolutional Neural Network [70].

En este estudio se han utilizado 9 *datasets* diferentes formados por ECGs para la detección de fibrilación auricular. Con los ECGs de estos conjuntos de datos se han dividido los subconjuntos para el entrenamiento, validación y prueba del modelo clasificador de FA.

Estos son los conjuntos de datos utilizados:

- **MIT-BIH Atrial Fibrillation [19]:** Esta base de datos tiene 23 registros de ECGs públicos tomados de pacientes con FA cada 10 horas. Las señales de ECG se muestran a 250Hz, con cuatro tipos de anotaciones de ritmo (FA [AF], flutter auricular [AFL], ritmo juncional AV [J] y todos los demás ritmos [N]).
- **The 2017 PhysioNet/CinC Challenge [13]:** Todos los registros de ECG se muestran a 250Hz mediante una sola derivación con cuatro tipos de ritmo (ritmo sinusal normal [N], FA [AF], otras arritmias [NAF] y ruidosos [Noisy]). Han seleccionado solo los registros de N y AF.
- **The China Physiological Signal Challenge 2018 [44]:** Esta base de datos se recopiló de 11 hospitales con ECGs muestreados a 500 Hz. Todos los registros de ECG de 12 derivaciones duraron de 6 a 60 segundos y se tomaron de 3178 mujeres y 3699 hombres. El estudio ha utilizado los datos de una sola derivación (derivación II) con dos ritmos (ritmo sinusal normal [N] y FA [AF]), con 981 y 1.098 registros en total, respectivamente.
- **ECG Long Term AF [54]:** Esta base de datos tiene 84 registros de ECG a largo plazo de sujetos con FA paroxística o persistente. Cada registro se digitalizó a 128 Hz y las duraciones varían pero normalmente son de 24 a 25 horas. Los 38 registros indicados como terminación de FA han sido utilizados en el estudio.
- **Paroxysmal AF [46]:** Esta base de datos consta de 50 pares de registros de ECG de media hora muestreados a 128 Hz. La base de datos consta del grupo A, que experimentó FA paroxística, y grupo N, que no tenía FA paroxística. En el estudio solo se ha probado el grupo A, con un total de 48 registros.
- **MIT-BIH Arrhythmia [20]:** La base de datos se digitalizó a 360 muestras por segundo y contiene 48 extractos de media hora de registros de ECG ambulatorios de dos canales. La base de datos se obtuvo de 47 sujetos estudiados por el Laboratorio de Arritmias de BIH entre 1975 y 1979. La base de datos tiene dos tipos: latidos y ritmos. Este estudio ha probado el tipo de ritmo AF de los registros de ECG (registros 201, 203, 210, 217, 219 y 221).
- **AF Termination Challenge [45]:** Esta base de datos se divide en un conjunto de aprendizaje y dos conjuntos de prueba. El conjunto de aprendizaje contiene 30 registros en total, con 10 registros en cada uno de los tres grupos (N, S, y T). Cada registro se muestreó a 128Hz y los segmentos se extrajeron de registros de ECG de 20 a 24 horas. Los registros de ECG se crearon para su uso en

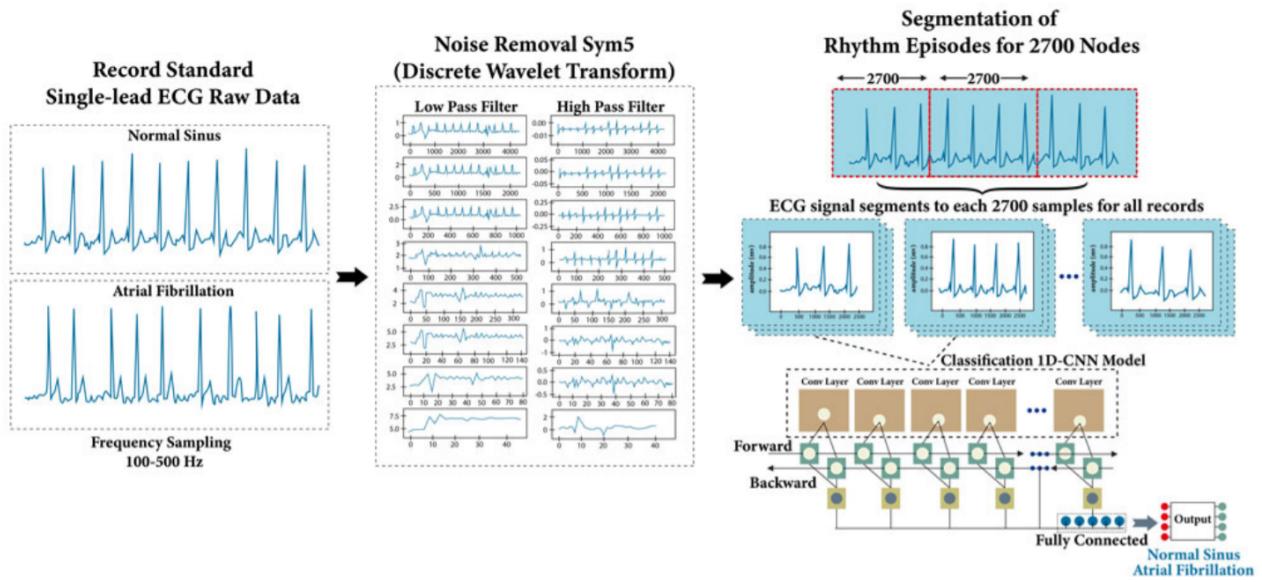
el Computers in Cardiology Challenge 2004. Entre los tres grupos, en este estudio se ha utilizado el grupo T, en el que la FA cesa inmediatamente.

- **Fantasia [28]:** Todos los registros de ECG de sujetos con ritmo sinusal se digitalizaron a 250Hz. Cada conjunto de datos incluye los datos del cinturón de respiración de 20 jóvenes (21-34 años) y 20 sujetos de edad avanzada (68-86 años). Para el conjunto de prueba, este estudio han utilizado sólo 24 registros elegidos al azar de las cohortes de jóvenes y ancianos.
- **ECG recording from Chapman University and Shaoxing People's Hospital [79]:** Esta base de datos incluye una gran cantidad de sujetos (más de 10.000) con señales de ECG de 12 derivaciones muestradas con un muestreo 500 Hz de frecuencia superior al habitual. La base de datos incluye 11 ritmos cardíacos y 56 tipos de afecciones cardiovasculares etiquetadas por médicos profesionales. Los registros de ECG se adquirieron durante 10 segundos y se recopilaron de 10.646 pacientes, incluidos 5956 hombres y 4690 mujeres. En este estudio, se han utilizado solo 1780 datos de AF y 1826 datos de ritmo sinusal normal seleccionados de la derivación II para el conjujunto de prueba.
- **ECG recording from an Indonesian Hospital:** El conjunto de datos del Hospital de Indonesia contenía 42 y 13 registros muestrados en 500Hz (N y AF) y 400Hz (AF), respectivamente. Se recopilaron todos los registros de ECG durante 10 segundos. La base de datos de ECG fue recopilada por médicos de pacientes que utilizan dispositivos de ECG ambulatorios (febrero a junio de 2020).

Se han utilizado tres conjuntos de datos (*MIT-BIH Atrial Fibrillation [19]*, *PhysioNet/CinC Challenge 2017 [13]* y las bases de datos *China Physiological Signal Challenge 2018 [44]*) para el entrenamiento y validación. A continuación, el modelo se ha probado utilizando los conjuntos de datos restantes.

En este estudio, se ha creado un red neuronal convolucional (CNN) para la detección de fibrilación auricular basada en 13 capas convolucionales y 5 capas *max-pooling* de una arquitectura 1D-CNN (ver Figura 7.2).

A la creación de la CNN le precede un proceso de eliminación de ruido de los registros de ECGs, y una segmentación de 2700 muestras de las señales de los ECGs. Una condición de FA puede contraerse hasta 600 latidos por minuto (bpm), creando así una alta irregularidad de los intervalos R-R y una repentina desaparición de los ondas P que se producen. Por lo tanto, son necesarios al menos tres o cuatro latidos para representar los episodios de FA. La segmentación de ECG de 2700 nodos contenía al menos dos intervalos R-R o tres latidos con diferentes muestreos de frecuencia (250Hz, 300Hz y 500Hz) en todos los registros. Además, con un muestreo de frecuencia mínima de 128Hz para el conjunto de prueba, la segmentación de 2700 nodos podría presentar más de dos intervalos R-R. Por lo tanto, se han seleccionado 2700 nodos para la segmentación de ECG como los mejores episodios del ECG.



7.2. Figura: Diseño de la CNN del estudio *AFibNet*

La arquitectura 1D-CNNs tiene dos tipos de capas distintos, seguidas por la capa CNN y la capa completamente conectada (FC). El aprendizaje de características se procesa en la capa CNN mediante el proceso de convolución y submuestreo (*pooling*). La función específica de las dos capas es reducir la complejidad y dimensión de la función de ECG. En este estudio, se ha generado uno por uno cada episodio (2700 nodos) como una característica junto con el registro de la señal de ECG. La amplitud está representada por cada nodo desde el nodo 1 al 2700. Las capas de CNN procesan y aprenden a extraer características (aprendizaje de características) de los datos 1D sin procesar, que se utilizan en la tarea de clasificación realizada por las capas FC. Como resultado, tanto la extracción de características como las operaciones de clasificación se integran en un proceso que se puede simplificar para mejorar el rendimiento de la clasificación.

En el proceso de convolución, se generan varias características únicas. Por ejemplo, en la capa de convolución 1, se alimentó con 2700 nodos de la señal de ECG. Esta capa tiene 64 núcleos con un tamaño de 3×1 y un salto de 1. Este proceso de saltos se lleva a cabo a lo largo de la señal del ECG para que pueda producir 3 características: al principio, en la mitad y al final del episodio de la señal. Luego, el resultado del proceso de convolución continúa hasta la capa de *max-pooling*. La capa de agrupación (*pooling*) tiene como objetivo resumir las características resultantes del proceso de convolución para que no solo reduzca la carga de cálculo, sino que también pueda fortalecer el modelo frente a variaciones en la señal de entrada.

La arquitectura de la red neuronal convolucional se muestra en la Figura 7.3.

Layer	Input Nodes	Filter Number	Kernel Size/Pool Size	Output Nodes	Feature Interpretation
Input	2700,1	-		-	ECG amplitude for one episode
Convolution 1	2700,1	64	3x1, stride 1	2698 x 64	64 feature map
Convolution 2	2698 x 64	64	3x1, stride 1	2696 x 64	64 feature map
Max-pooling 1	2696 x 64	-	2x1, stride 2	1348 x 64	Feature reduction (1348 nodes for one episode)
Convolution 3	1348 x 64	128	3x1, stride 1	1346 x 128	128 feature map
Convolution 4	1346 x 128	128	3x1, stride 1	1344 x 128	128 feature map
Max-pooling 2	1344 x 128	-	2x1, stride 2	672 x 128	Feature reduction (672 nodes for one episode)
Convolution 5	672 x 128	256	3x1, stride 1	670 x 256	256 feature map
Convolution 6	670 x 256	256	3x1, stride 1	668 x 256	256 feature map
Convolution 7	668 x 256	256	3x1, stride 1	666 x 256	256 feature map
Max-pooling 3	666 x 256	-	2x1, stride 2	333 x 256	Feature reduction (672 nodes for one episode)
Convolution 8	333 x 256	512	3x1, stride 1	331 x 512	512 feature map
Convolution 9	331 x 512	512	3x1, stride 1	329 x 512	512 feature map
Convolution 10	329 x 512	512	3x1, stride 1	327 x 512	512 feature map
Max-pooling 4	327 x 512	-	2x1, stride 2	163 x 512	Feature reduction (163 nodes for one episode)
Convolution 11	163 x 512	512	3x1, stride 1	161 x 512	512 feature map
Convolution 12	161 x 512	512	3x1, stride 1	159 x 512	512 feature map
Convolution 13	159 x 512	512	3x1, stride 1	157 x 512	512 feature map
Max-pooling 5	157 x 512	-	2x1, stride 2	78 x 512	Feature reduction (78 nodes for one episode)
Flatten	39936	-	-	-	Dot product between 78 nodes and 512 feature map
Dense	-	-	-	1000	Weight params
Dense	-	-	-	1000	Weight params
Output	-	-	-	1	Class

7.3. Figura: Arquitectura de la CNN del estudio *AFibNet*

En general, el modelo ha obtenido un buen rendimiento con una *accuracy* de hasta el 99,80 %. La *accuracy*, sensibilidad, especificidad, *F-score* y la precisión para las dos clases (ritmo sinusal normal [N] y fibrilación auricular [AF]) en una validación *10-fold Cross Validation* fueron de 99,8 %, 99,8 %, 99,8 %, 99,77 % y 99,74 %, respectivamente.

Classification of 12-lead ECGs: the PhysioNet/ Computing in Cardiology Challenge 2020 [2].

Es la competición de PhysioNet de 2020 para la clasificación de ECGs de 12 derivaciones. Se obtuvieron un total de 66.361 registros de ECG de 12 derivaciones de sistemas de seis hospitales de cuatro países en tres continentes. Se publicaron 43.101 registros públicamente con un enfoque en 27 diagnósticos, entre los que se encuentra la fibrilación auricular. Un total de 217 equipos enviaron 1395 algoritmos durante el desafío, que representa una diversidad de enfoques para identificar estas 27 anomalías cardíacas.

Se ha usado la información de las siguientes bases de datos:

- **CPSC Database and CPSC-Extra Database [44]:** Esta fuente incluye tres bases de datos: un conjunto de datos de entrenamiento público (CPSC), un conjunto de datos no utilizado (CPSC-Extra) y un conjunto de datos de prueba (el conjunto oculto de la CPSC) de la CPSC2018. Los datos de la CPSC y la CPSC-Extra fueron compartidos como conjuntos de entrenamiento. El conjunto de CPSC oculto se dividió en validación y prueba
- **INCART Database [76]:** El conjunto de datos se compartió como un conjunto de entrenamiento.
- **PTB and PTB-XL Database [6]:** Esta fuente incluye dos bases de datos públicas: la base de datos de ECG de diagnóstico PTB y la base de datos PTB-XL, un gran conjunto de datos de ECG disponible. Se compartieron como conjuntos de entrenamiento.
- **The Georgia 12-lead ECG Challenge (G12EC) Database:** Esta es una nueva base de datos, que representa una gran población del sureste de los Estados Unidos, y se divide en los conjuntos

de entrenamiento, validación y prueba. El conjunto de validación y prueba lo formaba el conjunto G12EC oculto.

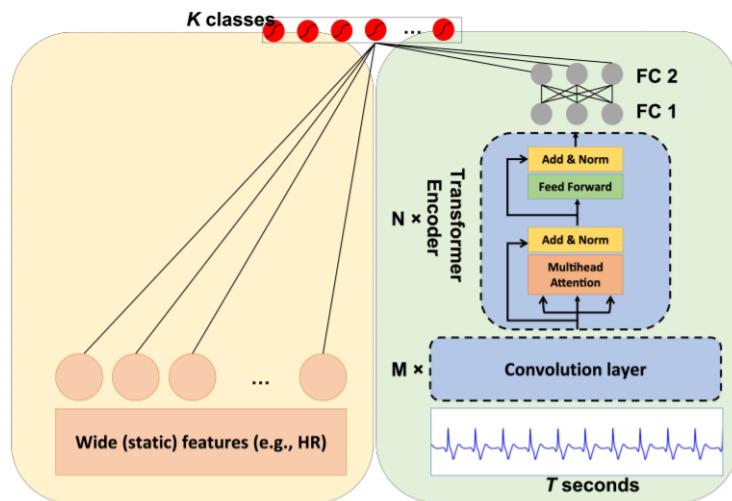
- **Undisclosed Database:** Este conjunto de datos nunca se publicó y se utilizó como un conjunto de prueba para la competición.

Cada registro de ECG anotado contenía datos de señales de ECGs de 12 derivaciones con una muestra de frecuencia que varía de 257Hz a 1kHz. También se incluía información demográfica como la edad y el sexo, y el o los diagnósticos, es decir, las etiquetas de los datos de la competición.

A continuación se ha recopilado información de los dos equipos cuyas implementaciones han obtenido las dos mejores puntuaciones de la competición:

prna team [50].

En este trabajo, se presenta una red de transformación amplia (*wide*) y profunda (*deep*) para la clasificación multietiqueta de 27 diagnósticos a partir de ECGs, entre los que se incluye la FA. Antes de nada, se aumentó o redujo cada registro a 500Hz y se extrajeron 15 segundos de cada uno de ellos. La red combina características estáticas de ECGs seleccionadas a mano (*wide features*), junto con características extraídas directamente de la forma de onda de ECG sin procesar a través de una red neuronal (*deep features*). Para codificar características *deep*, primero se realiza una serie de operaciones de convolución para aprender un *embedding* de la forma de onda ECG sin procesar. Los *embeddings* aprendidos se alimentan luego a una arquitectura de transformación (*Transformer Encoder*), más rápida de entrenar que las arquitecturas neuronales recurrentes típicas que requieren computación secuencial. Un conjunto final de capas completamente conectadas (FC) combina las características *wide* y *deep* para producir clasificaciones multietiqueta de los resultados de ECG (ver Figura 7.4).



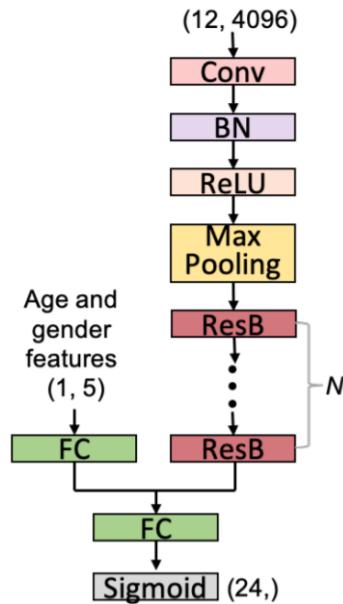
7.4. Figura: Arquitectura de la red neuronal del estudio del equipo *prna*

Para la extracción de las características *wide*, inicialmente se extrajeron más de 300 características de la derivación II del ECG, incluidas características de variabilidad de la frecuencia cardíaca (tiempo, dominio de frecuencia y no lineal) y características morfológicas. Se entrenó un modelo *random forest* inicial para investigar la importancia de las características y se seleccionaron las 20 características principales identificadas. También se agregaron la edad y el sexo para obtener un total de 22 características *wide*. En conjunto, estas características se enviaron como entrada a la red *wide* y se concatenaron con las salidas aprendidas de la parte *deep* del modelo.

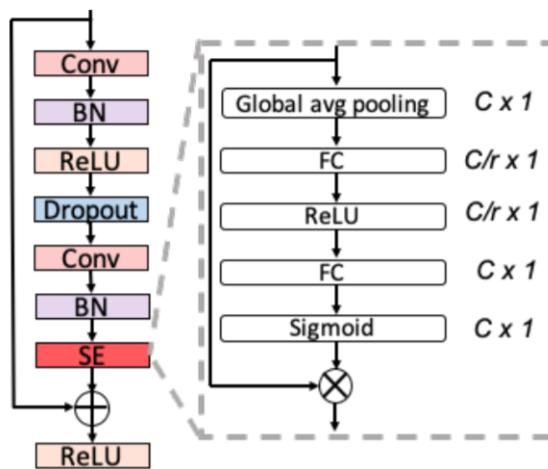
El segundo componente del modelo (*deep*) consta de los siguientes módulos: una red de *embedding* que extrae información directamente del segmento de forma de onda del ECG, una pila de *Transformer Encoders*, y un encabezado de clasificación de etiquetas múltiples.

Between_a_ROC_and_a_heart_place team [78].

En este trabajo, se diseñó una *ResNet* mejorada para asignar los registros de ECG de 12 derivaciones a las 24 clases de diagnóstico. Como preprocesado, todos los ECG se volvieron a muestrear a la frecuencia mínima de 257Hz y cada ECG se estableció en 4096 puntos, es decir, un poco menos de aproximadamente 16 segundos. La *ResNet* mejorada consta de una capa convolucional seguida de 8 bloques residuales (*ResB*), cada uno de los cuales contiene dos capas convolucionales y un bloque de compresión y excitación (SE) (ver Figuras 7.5 y 7.6). Debido al hecho de que algunas muestras tienen múltiples clases de 27 diagnósticos clínicos, en lugar de usar la función softmax como en el problema de clasificación tradicional, se asumió que cada clase era independiente y se usó la función sigmoidea para cada neurona de salida para hacer frente a este problema multietiquetado.



7.5. Figura: Arquitectura de la red neuronal del estudio del equipo *Between_a_ROC_and_a_heart_place*



7.6. Figura: Bloque residual (*ResB*) y bloque SE de la red neuronal del equipo *Between_a_ROC_and_a_heart_place*

La primera capa (convolucional) y las dos unidades *ResB* iniciales tienen 64 filtros de convolución. El número de filtros aumenta en un factor de dos por cada segunda unidad *ResB*. Las dimensiones de las características se reducen a la mitad después de la capa *max-pooling* y las tercera, quinta y séptima *ResB*.

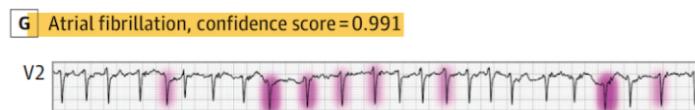
El *ResNet* mejorado tiene cuatro modificaciones del *ResNet* original. Primero, se modificó la capa final completamente conectada (FC) para incorporar la edad y el sexo del paciente. Estas dos características se pasaron a través de otra capa de FC con 10 neuronas antes de su inclusión en la capa final. En segundo lugar, se utilizó un tamaño de núcleo de 15 relativamente grande en el primer núcleo convolucional y un tamaño de núcleo de 7 en los últimos núcleos convolucionales. Trabajos anteriores han demostrado que los tamaños de kernel grandes son más útiles para que las redes aprendan características significativas. En tercer lugar, como se muestra en la Figura 6.6, se agregó una capa de *dropout* con una tasa de abandono de 0.2 entre dos núcleos convolucionales en cada *ResB* para reducir la probabilidad de sobreajuste. Finalmente, se agregó un bloque SE en cada *ResB*. El bloque SE sirve para modelar interdependencias de canales, y en este caso, se incorporó para modelar la relación espacial entre los canales de ECG. El bloque SE utiliza un perceptrón multicapa (MLP) con una capa oculta para calcular la importancia de los canales. El parámetro $r = 16$ en la Figura 7.6 denota el factor de reducción, que controla la capacidad del MLP.

JAMA Cardiology — Performance of a Convolutional Neural Network and Explainability Technique for 12-Lead Electrocardiogram Interpretation [27].

En este estudio, se ha entrenado una CNN para predecir la presencia de 38 clases de diagnóstico en 5 categorías a partir de datos de ECGs de 12 derivaciones. Entre los diagnósticos se encuentran la fibrilación auricular y el ritmo sinusal normal. El modelo CNN tiene una arquitectura *ResNet*. Se ha utilizado una técnica de explicabilidad de CNN llamada LIME (Linear Interpretable Model-Agnostic Explanations) para visualizar los segmentos de ECG que contribuyen a los diagnósticos de la CNN.

La CNN tuvo un AUC de al menos 0.910 y ha demostrado *F1-scores* comparables y una mayor sensibilidad que los cardiólogos, excepto en la fibrilación auricular (*F1-score* de la CNN de 0.847 frente al *F1-score* de los cardiólogos de 0.881), y otras pocas arritmias.

Comprender lo que las CNN aprenden basándose en los datos de los ECGs puede proporcionar información clínica valiosa. Por lo tanto, se ha aplicado una técnica de explicabilidad llamada LIME para resaltar qué segmentos del ECG contribuyen a los predicho diagnósticos según lo aprendido por la CNN de los datos de entrenamiento. Es decir, para mitigar la limitación de las *black boxes* de las CNN, la técnica de explicabilidad LIME ha identificado los segmentos del ECG que la CNN ha aprendido como importantes para cada diagnóstico. En la figura 7.7 se muestran en color más intenso los segmentos de mayor importancia para la CNN en las predicciones de fibrilación auricular teniendo en cuenta la derivación V2.



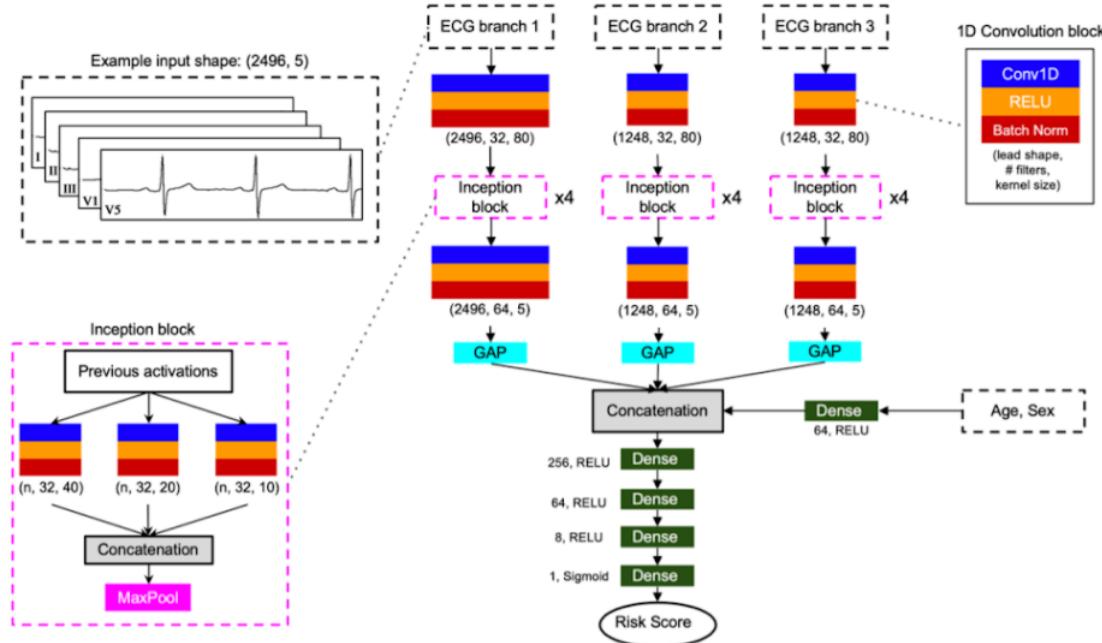
7.7. Figura: Segmentos de la derivación V2 de un ECG resaltados por LIME como más importantes para la clasificación de fibrilación auricular

Circulation: Deep Neural Networks Can Predict New Onset Atrial Fibrillation From the 12-Lead ECG and Help Identify Those at Risk of Atrial Fibrillation–Related Stroke [57].

En este estudio, se ha diseñado una red neuronal convolucional profunda (DNN) utilizando solo trazos de ECG digitales como entrada en 3 ramas temporalmente coherentes. Los datos se han reestructurado en señales de 0 a 5s para las derivaciones I, II, V1 y V5 en la primera rama, señales de 5 a 7.5s para las derivaciones V1, V2, V3, II y V5 en la segunda rama, y señales de 7.5 a 10s para las derivaciones II, V1, V4, V5 y V6 en la tercera rama (ver Figura 7.8).

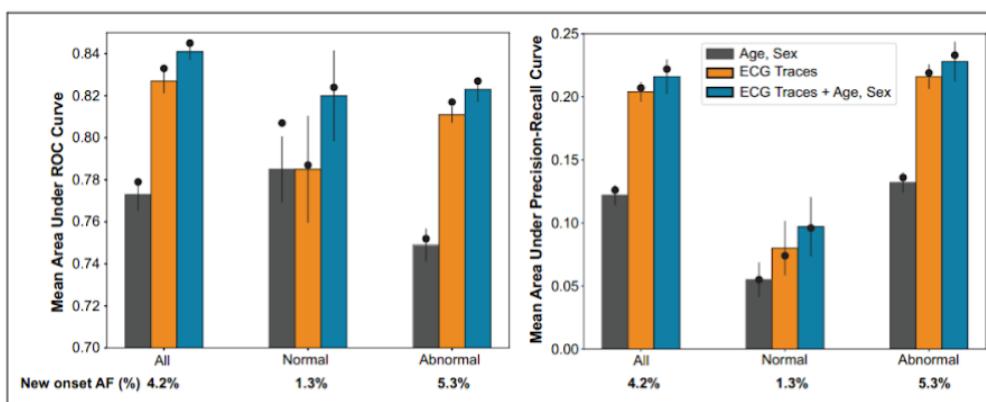
Se han implementado tres modelos:

- Modelo de *Extreme gradient boosting (XGBoost)* usando solo la edad y el sexo como entradas.
- DNN usando trazos de ECG como entradas (DNN-ECG)
- DNN usando trazos de ECG y la edad y el sexo como entradas (DNN-ECG-AS)



7.8. Figura: Arquitectura de la DNN del estudio de *Circulation*

El AUROC (área bajo la curva ROC) y AUPRC (área bajo la curva *precision-recall*) de los modelos DNN para la predicción de FA de nueva aparición en 1 año en el conjunto de validación han sido de 0.83 (CI del 95 %, 0.83–0.84) y 0.21 (CI del 95 %, 0.20–0.22), respectivamente, para DNN-ECG; y 0.85 (CI del 95 %, 0.84–0.85) y 0.22 (CI del 95 %, 0.21–0.24), respectivamente, para DNN-ECG-AS. Este rendimiento representa una mejora significativa en comparación con el modelo XGBoost que utiliza solo la edad y el sexo (AUROC, 0.78 [CI del 95 %, 0.77–0.79]; AUPRC, 0.13 [CI del 95 %, 0.12–0.14]) (ver Figura 7.9). En el conjunto de validación, había datos suficientes para calcular las puntuaciones de CHARGE-AF para el 65 % de los pacientes. En este subconjunto, el DNN-ECG-AS ha mostrado un rendimiento superior (AUROC, 0.84 [CI del 95 %, 0.83–0.85]; AUPRC, 0.20 [CI del 95 %, 0.19–0.22]) en comparación con la puntuación CHARGE-AF (AUROC, 0.79 [IC del 95 %, 0.78–0.80]; AUPRC, 0.12 [IC del 95 %, 0.11–0.13]).



7.9. Figura: Rendimiento de los modelos del estudio de *Circulation* en cuanto a las métricas AUROC y AUPRC

Convolutional Neural Network and Rule-Based Algorithms for Classifying 12-lead ECGs [64].

Es uno de los trabajos de la competición de clasificación de 12-lead ECGs de PhysioNet en 2020 [2], concretamente el del equipo *TeamUIO* que quedó en el 20º puesto.

El objetivo de este estudio fue clasificar 27 anomalías cardíacas basándose en un conjunto de datos de 43.101 registros de ECG. Se desarrolló un modelo híbrido que combina un algoritmo basado en reglas con diferentes arquitecturas de aprendizaje profundo.

Se compararon dos redes neuronales convolucionales diferentes, una red neuronal totalmente convolucional (FCN) y una red codificadora (Encoder), una combinación de ambas, y con la adición de otra red neuronal (MLP) que utiliza la edad y el sexo como entrada. Finalmente, dos de estas combinaciones se combinaron con un modelo basado en reglas que utiliza características de ECG derivadas.

Como punto de partida para clasificar las señales de ECG, se emplearon los tipos FCN y Encoder de modelos CNN. Todas las capas de salida de cada modelo utilizaron una función de activación sigmoidea.

Para hacer uso de los datos de edad y género proporcionados, se agregó un modelo de red neuronal (MLP) más simple con 2 entradas, una capa oculta de 50 unidades y 2 salidas en la capa final. Este nuevo modelo se combinó con los modelos FCN y Encoder mediante la concatenación de la última capa de las CNN.

Los dos modelos de CNN (FCN y Encoder) también se combinaron como modelos paralelos, concatenados en la penúltima capa. Este modelo fue probado con y sin una capa densa paralela.

Durante el entrenamiento, se usó la puntuación del área bajo la curva (AUC) en el conjunto de validación para determinar si la tasa de aprendizaje (*learning rate*) debería disminuir o permanecer. La tasa de aprendizaje se estableció inicialmente en 0.001 para todos los modelos y se redujo en un factor de 10 (*reduce on plateau method*) para cada *epoch* en la que la puntuación AUC no mejoró. La interrupción temprana (*early stopping*) se activó cuando la puntuación AUC en los datos de validación no mejoró durante dos épocas sucesivas.

Para todos los modelos, tanto en desarrollo como en implementación, se usó el optimizador Adam, un tamaño de lote (*batch size*) de 30 y entropía cruzada binaria (*binary crossentropy*) como función de pérdida. Se utilizó un generador de lotes para alimentar el modelo con datos durante el entrenamiento, programado para cambiar el orden de los datos para cada *epoch*.

Bibliografía

- [1] Personalised medicine. background, conference reports, publications and links related to personalised medicine.
- [2] EA Perez Alday, A Gu, A J Shah, C Robichaux, AK Ian Wong, C Liu, F Liu, A Bahrami Rad, A Elola, S Seyedi, Q Li, A Sharma, GD Clifford, and MA Reyna. Classification of 12-lead ecgs: the physionet/computing in cardiology challenge 2020. *Physiol Meas.*, 41:124003, Jan 2021.
- [3] TW Barrett, WH Self, BS Wasserman, CD McNaughton, and D Darbar. Evaluating the hatch score for predicting progression to sustained atrial fibrillation in ed patients with new atrial fibrillation. *Am J Emerg Med.*, 31:792–7, May 2013.
- [4] A Berkowitsch, M Kuniss, H Greiss, M Wójcik, S Zaltsberg, S Lehinant, S Erkapic, D Pajitnev, HF Pitschner, CW Hamm, and T Neumann. Impact of impaired renal function and metabolic syndrome on the recurrence of atrial fibrillation after catheter ablation: a long term follow-up. *Pacing Clin Electrophysiol.*, 35:532–43, May 2012.
- [5] S Blum, P Meyre, S Aeschbacher, S Berger, C Auberson, and M Briel. Incidence and predictors of atrial fibrillation progression: A systematic review and meta-analysis. *Heart Rhythm.*, 16:507–9, 2019.
- [6] R Bousseljot, D Kreiseler, and A Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. *Biomedizinische Technik, Band 40, Ergänzungsband.*, 1:317, 1995.
- [7] U Canpolat, K Aytemir, H Yorgun, L Şahiner, EB Kaya, and A Oto. A proposal for a new scoring system in the prediction of catheter ablation outcomes: promising results from the turkish cryoablation registry. *Int J Cardiol.*, 169:201–6, Nov 2013.
- [8] H Cao, O Zhou, Y Wu, Q Li, OD Røe, Y Chen, Z Wu, and D Wang. Preoperative serum soluble receptor activator of nuclear factor-kappab ligand and osteoprotegerin predict postoperative atrial fibrillation in patients undergoing cardiac valve surgery. *Ann Thorac Surg.*, 96:800–6, Sep 2013.
- [9] Z Che, J St Sauver, H Liu, and Y Liu. Deep learning solutions for classifying patients on opioid use. *AMIA Annu Symp Proc.*, 2017:525–534, Apr 2018.
- [10] D Chen, N Afzal, S Sohn, EB Habermann, JM Naessens, DW Larson, and H Liu. Postoperative bleeding risk prediction for patients undergoing colorectal surgery. *Surgery.*, 164:1209–1216, Dec 2018.
- [11] D Chen, G Goyal, RS Go, SA Parikh, and CG Ngufor. Improved interpretability of machine learning model using unsupervised clustering: Predicting time to first treatment in chronic lymphocytic leukemia. *JCO Clin Cancer Inform.*, 3:1–11, May 2019.
- [12] D Chen, S Liu, P Kingsbury, S Sohn, CB Storlie, EB Habermann, JM Naessens, DW Larson, and H Liu. Deep learning and alternative learning strategies for retrospective real-world clinical data. *NPJ Digit Med.*, 2, May 2019.
- [13] G Clifford, L Chengyu, M Benjamin, L Li-wei, S Ikarlo, J Alistair, and M Roger. Af classification from a short single lead ecg recording - the physionet computing in cardiology challenge 2017. 2017.
- [14] JE Dalen and JS Alpert. Silent atrial fibrillation and cryptogenic strokes. *Am J Med.*, 130:264–267, Mar 2017.

- [15] CB de Vos, R Pisters, R Nieuwlaat, MH Prins, EG Tielemans, RJ Coelen, AC van den Heijkant, MA Allessie, and HJ Crijns. Progression from paroxysmal to persistent atrial fibrillation clinical correlates and prognosis. *J Am Coll Cardiol.*, 55:725–731, 2010.
- [16] CB de Vos, R Pisters, R Nieuwlaat, MH Prins, RG Tielemans, RJ Coelen, AC van den Heijkant, MA Allessie, and HJ Crijns. Progression from paroxysmal to persistent atrial fibrillation clinical correlates and prognosis. *J Am Coll Cardiol.*, 55:725–31, Feb 2010.
- [17] H Deng, Y Bai, A Shantsila, L Fauchier, TS Potpara, and GYH Lip. Clinical scores for outcomes of rhythm control or arrhythmia progression in patients with atrial fibrillation: a systematic review. *Clin Res Cardiol.*, 71:813–823, Oct 2017.
- [18] H Deng, Y Bai, A Shantsila, L Fauchier, TS Potpara, and GYH Lip. Clinical scores for outcomes of rhythm control or arrhythmia progression in patients with atrial fibrillation: a systematic review. *Clin Res Cardiol.*, 106:817, 2017.
- [19] GB Moody GB, RG Mark, and . A new method for detecting atrial fibrillation using r-r intervals. *Computers in Cardiology.*, 10:227–230, 1983.
- [20] GB Moody GB, RG Mark, and . The impact of the mit-bih arrhythmia database. *IEEE Eng in Med and Biol.*, 20:45–50, May-June 2001.
- [21] J Gu, X Liu, H Tan, L Zhou, W Jiang, Y Wang, Y Liu, and J Gu. Impact of chronic obstructive pulmonary disease on procedural outcomes and quality of life in patients with atrial fibrillation undergoing catheter ablation. *J Cardiovasc Electrophysiol.*, 24:148–54, Feb 2013.
- [22] JJ Gómez-Doblas, MA López-Garrido, I Esteve-Ruiz, and G Barón-Esquivias. Epidemiología de la fibrilación auricular. *Rev Esp Cardiol.*, 16, 2016.
- [23] J Hata, T Nagata, S Sakata, E Oishi, Y Furuta, Y Hirakawa, T Honda, D Yoshida, T Kitazono, and T Ninomiya. Risk prediction model for incident atrial fibrillation in a general Japanese population - the hisayama study. *Circ J.*, Feb 2021.
- [24] S Higa, A Maesato, S Ishigaki, K Suenari, Y Chen, and S Chen. Diabetes and endocrine disorders (hyperthyroidism/hypothyroidism) as risk factors for atrial fibrillation (review). *Cardiac Electrophysiology Clinics.*, 13:63–75, 2021.
- [25] JCL Himmelreich, WAM Lucassen, RE Harskamp, C Aussems, HCPM van Weert, and MMJ Nielen. Charge-af in a national routine primary care electronic health records database in the netherlands: validation for 5-year risk of atrial fibrillation and implications for patient selection in atrial fibrillation screening. *Open Heart.*, 8, Jan 2021.
- [26] G Hindricks, T Potpara, N Dagres, E Arbelo, JJ Bax, C Blomström-Lundqvist, G Boriani, M Castella, GA Dan, PE Dilaveris, L Fauchier, G Filippatos, JM Kalman, M La Meir, DA Lane, JP Lebeau, M Lettino, GYH Lip, FJ Pinto, GN Thomas, M Valgimigli, IC Van Gelder, BP Van Putte, and CL Watkins. Esc scientific document group. 2020 esc guidelines for the diagnosis and management of atrial fibrillation developed in collaboration with the european association for cardio-thoracic surgery (eacts). *Eur Heart J.*, 42:373–498, Feb 2021.
- [27] JW Hughes, JE Olgin, R Avram, SA Abreau, T Sittler, K Radia, H Hsia, T Walters, B Lee, JE Gonzalez, and GH Tison. Performance of a convolutional neural network and explainability technique for 12-lead electrocardiogram interpretation. *JAMA Cardiol.*, 6:1285–1295, Nov 2021.
- [28] N Iyengar, C-K Peng, R Morin, AL Goldberger, and LA Lipsitz. Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics. *Am J Physiol.*, 271:1078–1084, 1996.
- [29] V Jacobs, HT May, TL Bair, BG Crandall, M Cutler, JD Day, JP Weiss, JS Osborn, JB Muhlestein, JL Anderson, C Mallender, and TL Bunch. The impact of risk score (chads2 versus cha2ds2-vasc) on long-term outcomes after atrial fibrillation ablation. *Heart Rhythm.*, 12:681–6, Apr 2015.
- [30] CT January, LS Wann, H Calkins, LY Chen, JE Cigarroa, JC Jr Cleveland, PT Ellinor, MD Ezekowitz, ME Field, KL Furie, PA Heidenreich, KT Murray, JB Shea, CM Tracy, and CW Yancy. 2019 aha/acc/hrs focused update of the 2014 aha/acc/hrs guideline for the management of patients with

- atrial fibrillation: A report of the american college of cardiology/american heart association task force on clinical practice guidelines and the heart rhythm society. *J Am Coll Cardiol.*, 74:104–132, Jul 2019.
- [31] C Johansson, MM Lind, M Eriksson, and L Johansson. Weight, height, weight change, and risk of incident atrial fibrillation in middle-aged men and women. *J Arrhythm.*, 36:974–981, Jul 2020.
- [32] S Khurshid, U Kartoun, JM Ashburner, L Trinquart, A Philippakis, AV Khera, PT Ellinor, K Ng, and SA Lubitz. Performance of atrial fibrillation risk prediction models in over 4 million individuals. *Circ Arrhythm Electrophysiol.*, 14, Jan 2021.
- [33] S Kochhäuser, DG Dechering, K Trought, P Hache, T Haig-Carter, Y Khaykin, Z Wulffhart, A Pantano, B Tsang, L Eckardt, and A Verma. Predictors for progression of atrial fibrillation in patients awaiting atrial fibrillation ablation. *Can J Cardiol.*, 32:1348–1354, Nov 2016.
- [34] TA Koleck, SA Mitha, A Biviano, BA Caceres, EJ Corwin, and I Goldenthal. Exploring depressive symptoms and anxiety among patients with atrial fibrillation and/or flutter at the time of cardioversion or ablation. *J Cardiovasc Nurs.*, 2020.
- [35] MJ Kolek, AJ Graves, M Xu, A Bian, PL Teixeira, MB Shoemaker, B Parvez, H Xu, SR Heckbert, PT Ellinor, EJ Benjamin, A Alonso, JC Denny, KG Moons, AK Shintani, FE Jr Harrell, Dm Roden, and D Darbar. Evaluation of a prediction model for the development of atrial fibrillation in a repository of electronic medical records. *JAMA Cardiol.*, 1:1007–1013, Dec 2016.
- [36] J Kornej, G Hindricks, J Kosiuk, A Arya, P Sommer, D Husser, S Rolf, S Richter, C Piorkowski, T Gaspar, GY Lip, and A Bollmann. Renal dysfunction, stroke risk scores (chads2, cha2ds2-vasc, and r2chads2), and the risk of thromboembolic events after catheter ablation of atrial fibrillation: the leipzig heart center af ablation registry. *Circ Arrhythm Electrophysiol.*, 6:868–74, Oct 2013.
- [37] J Kornej, G Hindricks, J Kosiuk, A Arya, P Sommer, D Husser, S Rolf, S Richter, Y Huo, C Piorkowski, and A Bollmann. Comparison of chads2, r2chads2, and cha2ds2-vasc scores for the prediction of rhythm outcomes after catheter ablation of atrial fibrillation: the leipzig heart center af ablation registry. *Circ Arrhythm Electrophysiol.*, 7:281–7, Apr 2014.
- [38] J Kornej, G Hindricks, MB Shoemaker, D Husser, A Arya, P Sommer, S Rolf, P Saavedra, A Kanagamudram, S Patrick Whalen, J Montgomery, CR Ellis, D Darbar, and A Bollmann. The apple score: a novel and simple score for the prediction of rhythm outcomes after catheter ablation of atrial fibrillation. *Clin Res Cardiol.*, 104:871–6, Oct 2015.
- [39] J Kornej, G Hindricks, A Arya, P Sommer, D Husser, and A Bollmann. The apple score - a novel score for the prediction of rhythm outcomes after repeat catheter ablation of atrial fibrillation. *PLoS One.*, 12, Jan 2017.
- [40] DH Lau, S Nattel, JM Kalman, and P Sanders. Modifiable risk factors and atrial fibrillation. *Circulation.*, 136:583–596, Nov 2017.
- [41] KP Letsas, M Efremidis, G Giannopoulos, S Deftereos, L Lioni, P Korantzopoulos, K Vlachos, S Xydonas, C Kossyvakis, and A Sideris. Chads2 and cha2ds2-vasc scores as predictors of left atrial ablation outcomes for paroxysmal atrial fibrillation. *Europace.*, 16:202–7, Feb 2014.
- [42] A Li, Y Chen, W Wang, L Su, and Z Ling. Association of clinical predictors with recurrence of atrial fibrillation after catheter ablation. *Ann Noninvasive Electrocardiol.*, 25, Nov 2020.
- [43] T Lin, X Du, R Bai, YW Chen, RH Yu, DY Long, RB Tang, CH Sang, SN Li, CS Ma, and JZ Dong. Long-term results of single-procedure catheter ablation for atrial fibrillation in pre- and post-menopausal women. *J Geriatr Cardiol.*, 11:120–5, Jun 2014.
- [44] FF Liu, CY Liu, LN Zhao, XY Zhang, XL Wu, XY Xu, YL Lia, CY Ma, SS Wei, ZQ He, JQ Li, and NY Kwee. An open access database for evaluating the algorithms of ecg rhythm and morphology abnormal detection. *Journal of Medical Imaging and Health Informatics.*, 8:1368–1373, 2018.
- [45] GB Moody. Spontaneous termination of atrial fibrillation: A challenge from physionet and computers in cardiology 2004. *Computers in Cardiology.*, 31:101–104, 2004.

- [46] GB Moody, AL Goldberger, S McClenen, and SP Swiryn. Predicting the onset of paroxysmal atrial fibrillation: The computers in cardiology challenge 2001. *Computers in Cardiology.*, 28:113–116, 2001.
- [47] A Mostayed, J Luo, X Shu, and W Wee. Classification of 12-lead ecg signals with bi-directional lstm network. Nov 2018.
- [48] N Mujović, M Marinković, N Marković, A Shantsila, GY Lip, and TS Potpara. Prediction of very late arrhythmia recurrence after radiofrequency catheter ablation of atrial fibrillation: The mb-later clinical score. *Sci Rep.*, 7, Jan 2017.
- [49] Y Naruse, H Tada, M Satoh, M Yanagihara, H Tsuneoka, Y Hirata, Y Ito, K Kuroki, T Machino, H Yamasaki, M Igarashi, Y Sekiguchi, A Sato, and K Aonuma. Concomitant obstructive sleep apnea increases the recurrence of atrial fibrillation following radiofrequency catheter ablation of atrial fibrillation: clinical impact of continuous positive airway pressure therapy. *Heart Rhythm Off J Heart Rhythm Soc.*, 10:331–337, 2013.
- [50] A Natarajan, Y Chang, S Mariani, A Rahman, G Boverman, S Vij, and J Rubin. A wide and deep transformer neural network for 12-lead ecg classification. *2020 Computing in Cardiology.*, pages 1–4, 2020.
- [51] DL Packer, DB Mark, RA Robb, KH Monahan, TD Bahnsen, JE Poole, PA Noseworthy, YD Rosenberg, N Jeffries, LB Mitchell, GC Flaker, E Pokushalov, A Romanov, TJ Bunch, G Noelker, A Ardashev, A Revishvili, DJ Wilber, R Cappato, KH Kuck, G Hindricks, DW Davies, PR Kowey, GV Naccarelli, JA Reiffel, JP Piccini, AP Silverstein, HR Al-Khalidi, and KL Lee. Cabana investigators. effect of catheter ablation vs antiarrhythmic drug therapy on mortality, stroke, bleeding, and cardiac arrest among patients with atrial fibrillation: The cabana randomized clinical trial. *JAMA.*, 321:1261–1274, Apr 2019.
- [52] J Park, TH Kim, JS Lee, JK Park, JS Uhm, B Joung, MH Lee, and HN Pak. Prolonged pr interval predicts clinical recurrence of atrial fibrillation after catheter ablation. *J Am Heart Assoc.*, 3, 2014.
- [53] RK Pathak, ME Middeldorp, DH Lau, AB Mehta, R Mahajan, and D Twomey. Aggressive risk factor reduction study for atrial fibrillation and implications for the outcome of ablation: the arrest-of cohort study. *J Am Coll Cardiol.*, 64:2222–31, Dec 2014.
- [54] S Petrucci, AV Sahakian, and S Swiryn. Abrupt changes in fibrillatory wave characteristics at the termination of paroxysmal atrial fibrillation in humans. *Europace.*, 9:466–470, 2007.
- [55] TS Potpara, GR Stankovic, BD Beleslin, MM Polovina, JM Marinkovic, MC Ostojic, and GYH Lip. A 12-year follow-up study of patients with newly diagnosed lone atrial fibrillation: implications of arrhythmia progression on prognosis: the belgrade atrial fibrillation study. *Chest.*, 141:339–347, Feb 2012.
- [56] M Proietti and G Boriani. Obesity paradox in atrial fibrillation: Implications for outcomes and relationship with oral anticoagulant drugs. *Am J Cardiovasc Drugs.*, 20:125–137, 2020.
- [57] S Raghunath, AE Ulloa Cerna, L Jing, DP vanMaanen, J Stough, DN Hartzel, JB Leader, HL Kirchner, MC Stumpe, A Hafez, A Nemani, T Carbonati, KW Johnson, K Young, CW Good, JM Pfeifer, AA Patel, BP Delisle, A Alsaid, D Beer, CM Haggerty, and BK Fornwalt. Prediction of mortality from 12-lead electrocardiogram voltage data using a deep neural network. *Nat Med.*, 26:886–891, Jun 2020.
- [58] SY Roh, J Shim, KN Lee, J Ahn, DH Kim, DI Lee, JI Choi, and YH Kim. Gender-related difference in clinical outcome of the patient with atrial fibrillation after radiofrequency catheter ablation. *Korean Circ J.*, 48:605–618, Jul 2018.
- [59] A Ruigomez, S Johansson, MA Wallander, and LA Garcia Rodriguez. Predictors and prognosis of paroxysmal atrial fibrillation in general practice in the uk. *BMC Cardiovasc Disord.*, 5:20, 2005.
- [60] A Samaras, A Kartas, E Akritos, G Fotos, G Dividis, D Vasdeki, E Vrana, G Rampidis, H Karvounis, G Giannakoulas, and A Tzikas. A novel prognostic tool to predict mortality in patients with atrial fibrillation: The basic-af risk score. *Hellenic J Cardiol.*, Jan 2021.

- [61] NJ Sethi, J Feinberg, EE Nielsen, S Safi, C Gluud, and JC Jakobsen. The effects of rhythm control strategies versus rate control strategies for atrial fibrillation and atrial flutter: A systematic review with meta-analysis and trial sequential analysis. *PLoS One.*, 12, Oct 2017.
- [62] A Sepehri Shamloo, N Dagres, A Müssigbrodt, A Stauber, A Kircher, S Richter, S Dinov, B Bertagnolli, L Husser-Bollmann, D Bollmann, G Hindricks, and A Arya. Atrial fibrillation and cognitive impairment: New insights and future directions. *Heart Lung Circ.*, 29:69–85, Jan 2020.
- [63] I Silva, G Moody, DJ Scott, LA Celi, and RG Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. *Comput Cardiol (2010)*., 39:245–248, 2012.
- [64] BJ Singstad, C Tronstad, and . Convolutional neural network and rule-based algorithms for classifying 12-lead ecgs. *2020 Computing in Cardiology Conference.*, Dec 2020.
- [65] K Suenari, TF Chao, CJ Liu, Y Kihara, TJ Chen, and SA Chen. Usefulness of hatch score in the prediction of new-onset atrial fibrillation for asians. *Medicine (Baltimore)*., 96, Jan 2017.
- [66] RB Tang, JZ Dong, DY Long, RH Yu, M Ning, CX Jiang, CH Sang, XH Liu, and CS Ma. Efficacy of catheter ablation of atrial fibrillation beyond hatch score. *Chin Med J (Engl)*., 125:3425–9, Oct 2012.
- [67] P Tiwari, KL Colborn, DE Smith, F Xing, D Ghosh, and MA Rosenberg. Assessment of a machine learning model applied to harmonized electronic health record data for the prediction of incident atrial fibrillation. *JAMA Netw Open*., 3, Jan 2020.
- [68] SA Trines, G Stabile, E Arbelo, N Dagres, J Brugada, J Kautzner, E Pokushalov, AP Maggioni, C Laroche, M Anselmino, R Beinart, V Traykov, and C Blomström-Lundqvist. Influence of risk factors in the esc-ehra eorp atrial fibrillation ablation long-term registry. *Pacing Clin Electrophysiol.*, 42:1365–1373, Oct 2019.
- [69] D Tsiaichris, G Giannopoulos, S Deftereos, C Kossyvakis, C Tsiofis, C Siasos, E Oikonomou, K Gatzoulis, D Tousoulis, and C Stefanadis. Biomarkers determining prognosis of atrial fibrillation ablation. *Curr Med Chem.*, 26:925–937, 2019.
- [70] B Tutuko, S Nurmaini, AE Tondas, MN Rachmatullah, A Darmawahyuni, R Esafri, F Firdaus, and AI Sapitri. Afibnet: an implementation of atrial fibrillation detection with convolutional neural network. *BMC Med Inform Decis Mak.*, 21:216, Jul 2021.
- [71] RA Winkle, JW Jarman, RH Mead, G Engel, MH Kong, W Fleming, and RA Patrawala. Predicting atrial fibrillation ablation outcome: The caap-af score. *Heart Rhythm.*, 13:2119–2125, Nov 2016.
- [72] CX Wong, T Sullivan, MT Sun, R Mahajan, RK Pathak, and M Middeldorp. Obesity and the risk of incident, post-operative, and post-ablation atrial fibrillation: a meta-analysis of 626,603 individuals in 51 studies. *JACC Clin Electrophysiol.*, 1:139–52, 2015.
- [73] JT Wu, JZ Dong, CH Sang, RB Tang, and CS Ma. Prolonged pr interval and risk of recurrence of atrial fibrillation after catheter ablation. *Int Heart J.*, 55:126–130, 2014.
- [74] S Wu, S Liu, S Sohn, S Moon ans CI Wi, Y Juhn, and H Liu. Modeling asynchronous event sequences with rnns. *J Biomed Inform.*, 83:167–177, Jul 2018.
- [75] M Wójcik, A Berkowitsch, H Greiss, S Zaltsberg, D Pajitnev, N Deubner, CW Hamm, HF Pitschner, M Kuniss, and T Neumann. Repeated catheter ablation of atrial fibrillation: how to predict outcome? *Circ J.*, 77:2271–9, Jan 2013.
- [76] E Yakushenko. St petersburg incart 12-lead arrhythmia database. May 2008.
- [77] P Yang, Y Zhao, and ND Wong. Development of a risk score for atrial fibrillation in adults with diabetes mellitus (from the accord study). *Am J Cardiol.*, 125:1638–1643, Jun 2020.
- [78] Z Zhao, H Fang, SD Relton, R Yan, Y Liu, Z Li, J Qin, and DC Wong. Adaptive lead weighted resnet trained with different duration signals for classifying 12-lead ecgs. *2020 Computing in Cardiology.*, pages 1–4, 2020.

- [79] J Zheng, J Zhang, S Danioko, H Yao, H Guo, and C Rakovski. A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Sci Data.*, 7:48, Feb 2020.
- [80] J Zhuang, Y Lu, K Tang, W Peng, and Y Xu. Influence of body mass index on recurrence and quality of life in atrial fibrillation patients after catheter ablation: a meta-analysis and systematic review. *Clin Cardiol.*, 36:269–275, 2013.