Proyecto de predición de enfermedades cardíacas

Daniel Arias

9 de abril de 2025

1. Feature Engineering

1.1. Creación / modificación de variables

Durante esta etapa se realizaron dos procesos:

- **Renombramiento de las columnas**: Este proceso se realizó para obtener nombres mucho más descriptivos para unfuturo análisis de la influencia de cada una de las carácterísticas implementadas en el dataset.
- **Traducción de las variables**: Considerando que todas las variables categóricas dividen a cada columna en al menos dos sub categorías (por ejemplo, hombre o mujer) fue necesario asignar un nombre a cada una de estas categorías para posteriormente obtener las respectivas variables booleanas asociada para facilitar el entrenamiento del modelo.
- **Generar variables booleanas**: A partir de las variables categoricas ya traducidas a un formato de tipo string se las convirtió en variables booleanas mediante el método $.get_dummies()$ para facilitar más adelante el proceso de aprendizaje.

1.2. Selección de variables

Partiendo de la matriz de correlación obtenida durante la etapa de análisis exploratorio de datos 1. Se identifican como las variables que presentan una mayor correlación con la columna "num" son las columnas de "$exang$", "old_peak", "$slope$", "ca", y "$thal$" sin embargo este no es un método válido para la selección de variables para el entrenamiento del modelo puesto que al tratarse de un problema de clasificación y al presentarce varias características de tipo categórico no se pordría encontrar una correlación entre dichas características con la variable objetivo. Es por esto por lo que, se decidió emplear todos los features disponibles para el entrenamiento de los modelos a probar.

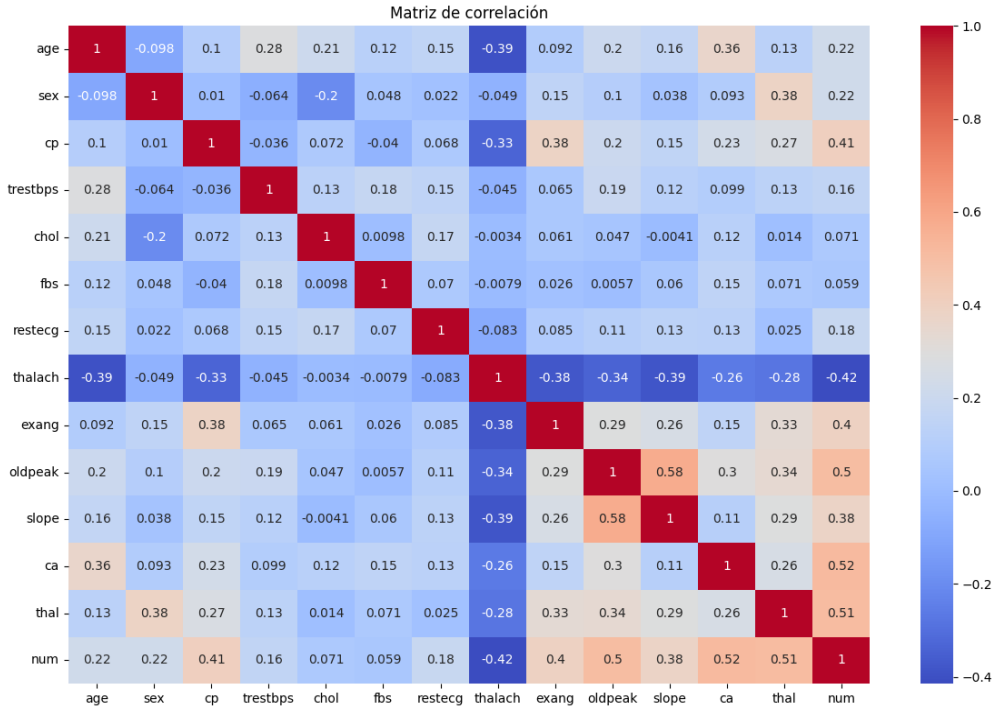

Figura 1: Matriz de correlación

2. Particionamiento de Datos

2.1. Divisiones del dataset original

Durante la etapa de partición de data únicamente se empleó la función $train_test_split$ correspondiente a la librería sklearn del módulo $model_select$. Para esto se dividió el data set original inicialmente en dos partes, una empleada para el entrenamiento y la validación (representando el 75 % de la muestra original) y otra destinada para la evaluación del modelo (representando el 25 % restante) . Una vez obtenidas ambas partes se tomo la fracción más grande y se la volvió a dividir en dos, una destinada únicamente al entrenamiento del modelo, y otra destinada para la validación del modelo manteniendo una proporción del 75 y 25 % respectivamente.

2.2. Manjeo del desbalanceo de clases

Al tratarse de datos relacionados con el área médica, es común encontrar un desbalanceo en las clases. Esto se debe a que los casos más severos de enfermedades cardíacas ocurren con menor frecuencia en compraración con escenarios en donde es paciente está sano, o incluso presenta algún cuadro de enfermedad cardíaca leve.

Debido a esto, fue necesario realizar un proceso de sobre muestreo para poder nivelar las clases con una menor frecuencia.

3. Modelado

En el proyecto de predicción de enfermedades cardíacas, se implementaron y compararon cuatro modelos de clasificación multiclase para identificar diferentes tipos de condiciones cardíacas: *RandomForestClassifier*, *XGBoostClassifier*, *AdaBoostClassifier* y *BaggingClassifier* con *DecisionTreeClassifier* como estimador base. La selección de estos modelos se basó en su capacidad para manejar problemas de clasificación multiclase, su robustez frente a datasets médicos con desbalance de clases y características mixtas, y su potencial para optimizar métricas como el F1-score macro. Además, se buscó explorar tanto técnicas de boosting como de bagging para evaluar su impacto en el rendimiento predictivo en un contexto multiclase.

El *RandomForestClassifier* se eligió por su adaptabilidad en problemas de clasificación multiclase, utilizando votación mayoritaria para combinar múltiples árboles de decisión, lo que lo hace robusto al sobreajuste y efectivo para datos desbalanceados mediante `class_weight="balanced"`. El *XGBoostClassifier* se probó debido a su alta precisión en el entrenamiento, optimizando la clasificación multiclase con `multi:softmax`. Por su parte, el *AdaBoostClassifier* se incluyó por su enfoque en mejorar la clasificación de muestras difíciles, utilizando el algoritmo SAMME para optimizar el rendimiento en problemas multiclase, lo que puede beneficiar la identificación de clases minoritarias.

Finalmente, el *BaggingClassifier* con *DecisionTreeClassifier* como estimador base se implementó para explorar una técnica de bagging que reduce la varianza de los árboles de decisión individuales, mejorando la estabilidad del modelo en un problema multiclase. Este enfoque permite combinar múltiples árboles entrenados en subconjuntos aleatorios del dataset, lo que lo hace adecuado para datasets pequeños como el de enfermedades cardíacas, donde la variabilidad entre predicciones puede ser alta. La comparación de estos modelos permitió seleccionar el más adecuado en términos de generalización y rendimiento, evaluando su capacidad para predecir con precisión las diferentes condiciones cardíacas.

4. Métricas de desempeño

Para evaluar el rendimiento de los modelos implementados en el proyecto de predicción de enfermedades cardíacas (*RandomForestClassifier*, *XGBoostClassifier*, *AdaBoostClassifier* y *BaggingClassifier* con *DecisionTreeClassifier* como estimador base), se utilizaron tres métricas principales: *recall*, *precisión* y *F1-score*, todas calculadas en su variante macro para abordar el problema de clasificación multiclase. Estas métricas se seleccionaron debido a la naturaleza del problema médico, donde es crucial equilibrar la capacidad del modelo para identificar correctamente las diferentes condiciones cardíacas (*recall*) y minimizar los falsos positivos (*precisión*), mientras que el *F1-score* proporciona una medida balanceada entre ambas. Dado el contexto clínico, se dio mayor importancia a la *precisión*, ya que un falso positivo podría llevar a tratamientos innecesarios o diagnósticos erróneos, con implicaciones significativas para los pacientes.

La *precisión* se priorizó como métrica principal, ya que mide la proporción de predicciones correctas para cada clase. Este enfoque permitió identificar qué modelo minimizaba mejor los falsos positivos en todas las categorías de enfermedades cardíacas, un aspecto crítico en aplicaciones médicas donde la confianza en las predicciones positivas es fundamental. Por ejemplo, un modelo con alta *precisión* garantiza que, cuando predice una condición cardíaca

específica, es muy probable que la predicción sea correcta, reduciendo el riesgo de intervenciones innecesarias.

El *recall* se empleó para evaluar la capacidad de los modelos para identificar correctamente todas las instancias de cada clase, un aspecto importante para no pasar por alto casos de enfermedades cardíacas graves. El *F1-score* , por su parte, se utilizó como una métrica complementaria para balancear *precisión* y *recall*, proporcionando una visión integral del rendimiento de los modelos en el contexto multiclase. La evaluación de estas métricas se realizó en el conjunto de validación, tras optimizar los hiperparámetros en el conjunto de validación, permitiendo seleccionar el modelo que mejor cumpliera con los objetivos del proyecto, priorizando la *precisión* para garantizar la fiabilidad de las predicciones en un entorno clínico.

5. Comparación de Resultados

Tabla 1: Métricas correspondientes con el modelo XGBoostClassifier

	precision	recall	f1-score	support
0	0.76	0.83	0.79	30.00
1	0.30	0.30	0.30	10.00
2	0.33	0.29	0.31	7.00
3	0.33	0.14	0.20	7.00
4	0.25	0.50	0.33	2.00
accuracy	0.57	0.57	0.57	0.57
macro avg	0.39	0.41	0.39	56.00
weighted avg	0.55	0.57	0.55	56.00

Tabla 2: Métricas correspondientes con el modelo RandomForestClassifier

	precision	recall	f1-score	support
0	0.67	0.97	0.79	30.00
1	0.00	0.00	0.00	10.00
2	0.00	0.00	0.00	7.00
3	0.00	0.00	0.00	7.00
4	0.15	1.00	0.27	2.00
accuracy	0.55	0.55	0.55	0.55
macro avg	0.17	0.39	0.21	56.00
weighted avg	0.37	0.55	0.44	56.00

Tabla 3: Métricas correspondientes con el modelo AdaBoostClassifier

	precision	recall	f1-score	support
0	0.63	0.97	0.76	30.00
1	0.00	0.00	0.00	10.00
2	0.00	0.00	0.00	7.00
3	0.00	0.00	0.00	7.00
4	0.20	1.00	0.33	2.00
accuracy	0.55	0.55	0.55	0.55
macro avg	0.17	0.39	0.22	56.00
weighted avg	0.34	0.55	0.42	56.00

Tabla 4: Métricas correspondientes con el modelo BagginClassifier

	precision	recall	f1-score	support
0	0.71	0.97	0.82	30.00
1	0.00	0.00	0.00	10.00
2	0.00	0.00	0.00	7.00
3	0.00	0.00	0.00	7.00
4	0.07	0.50	0.12	2.00
accuracy	0.54	0.54	0.54	0.54
macro avg	0.15	0.29	0.19	56.00
weighted avg	0.38	0.54	0.44	56.00

A partir de las métricas obtenidsa se obiene como mejor alternativa el modelo de XGBoost debido a que presenta una presición mayor en comparación con los otros modelos.