Ciencia de datos aplicada al estudio de la Obesidad y otras enfermedades crónicas en Córdoba

Práctico N°4

Aprendizaje Supervisado

Grupo 2

➢ Edgardo Garrigo
➢ Nicolas E Ponce

Objetivo y alcance

A partir del práctico anterior, el cual fue un primer acercamiento al proceso de aprendizaje automático, trabajaremos exclusivamente en el aprendizaje supervisado. Anteriormente han trabajado prediciendo la "obesidad" pero ahora cambiaremos la variable respuesta y pondremos atención en los algoritmos y técnicas. Nos enfocaremos en el proceso de: selección de un modelo, ajuste de hiperparámetros y evaluación, regulador, métricas, similar a lo hecho previamente. En este laboratorio no se espera que se encuentre el mejor modelo con sus mejores parámetros, sino que se logre la buena práctica de realizar los pasos necesarios en un proceso de aprendizaje automático, desde la división del dataset hasta la evaluación del modelo, además de aplicar las sugerencias o cambios recomendados en la devolución del lab 3. Por ello podemos decir que el objetivo del práctico es indagar entre los diferentes modelos de aprendizaje supervisado vistos en la materia y comparar el desempeño obtenido. Para realizar el práctico vamos a utilizar el dataset generado en el lab 2 y luego, a partir de lo realizado en el práctico de introducción al aprendizaje automático, cambiar y probar otros modelos vistos en esta materia y comparar los resultados a partir de la métrica seleccionada(la/s misma/s para los diferentes modelos).

La necesidad es la siguiente:

1. Poder predecir de forma automática la presencia de HTA en toda la población. Trabajaremos con la variable "HTA" dado que tiene en cuenta no sólo la declaración de hipertensión por la persona, sino que también si consumen medicación para la misma o al tomar la tensión dió elevada.
2. Poder predecir de forma automática la presencia de HTA en mujeres (grupo 1: Martin, Fer y Memi) y en hombres (grupo 2: Nico, Fer y Edgardo).
3. Poder determinar cuales son las variables que son consideradas factores de riesgo para presentar HTA, tanto para toda la población como discriminando por sexos (cada grupo lo hace para el sexo que les haya tocado).

Para ello se debe:

- Cargar los datos, separando del dataset la etiqueta a predecir (HTA).

- Dividir el dataset en el conjunto de entrenamiento y conjunto de test
- Analizar y justificar que features se utilizarán para lograr la mejor predicción.
- Elegir un modelo de **clasificación clásico** (por cada requerimiento). El que Uds. se sientan más cómodos, pero también justificando conceptualmente la elección del mismo.
- Y por otro lado, realizar otro modelo utilizando una **técnica de ensemble** (bagging, boosting) también justificando la elección del mismo.
- Entrenar y evaluar los modelos, fijando la semilla aleatoria para hacer repetible el experimento.
- En cuanto a los hiper-parámetros:
    1. Probar primero con los **default** y elegir alguna/s métrica/s para reportar los resultados.
    2. Luego usar **grid-search y 5-fold** cross-validation para explorar muchas combinaciones posibles de valores, reportando accuracy promedio y varianza para todas las configuraciones.
- Para la mejor configuración encontrada, evaluar sobre el conjunto de entrenamiento y sobre el conjunto de evaluación, reportando:

* Accuracy

* Precision

* Recall

* F1

* Matriz de confusión

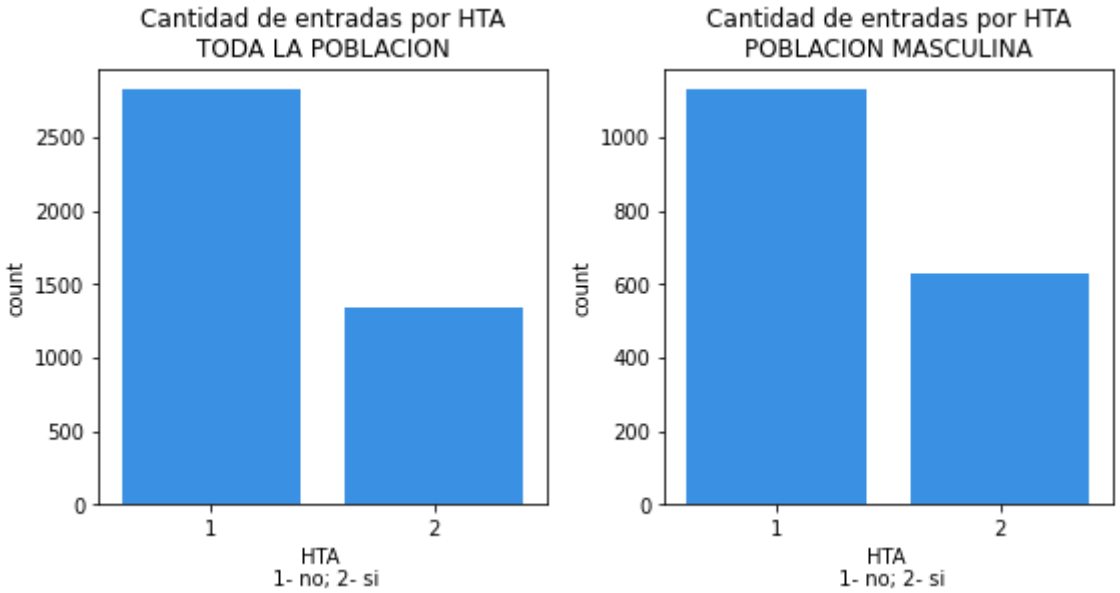
Informe:

El dataset original, generado en el práctico N°2, posee 242 features y 4288 registros, de los cuales 1814 corresponden a la población masculina y los 2474 restantes a la femenina. En este trabajo y luego de una preselección de **42** features de interés, empleamos 2 dataset para desarrollar el estudio, el dataset original con toda la población total y el dataset correspondientes a los registros de hombres.

Se separó de ambos dataset la etiqueta/*Target* a predecir HTA, que corresponde a una variable binaria que habla de la '*Presencia de hipertensión arterial según declaración de la persona y según medición' empleando el número* 1 como 'no' y el 2 como 'si'.

Respecto a la justificacion de los features preseleccionados que ayudarán significativamente a la predicción de **HTA** son podemos decir que muchos provienen del práctico anterior donde predecimos ausencia o presencia de obesidad, tal como: [sexo, edad, ecar, hta: ecar.1: icar: tecar: ten1max: ten1min: fuma: ante_ecar_pad ante_ecar_mad ante_dbt_her dbt tipo1 tipo2 cancer actfis stress IMC riesgo_ecar ten2max ten2min Naf fsodio que fque snac fsnac fritos car fcar']. Particularmente incorporamos features asociadas a comidas con sal (Naf fsodio que fque snac fsnac fritos car fcar), como también las mediciones (ten1max, ten1min, en2max y ten2min).

Ante el mensaje de *'Recuerden que las clases a clasificar están desbalanceadas' ,* lo corroboramos con un análisis previo del dataset. Como se puede ver el gráfico counterplot el atributo HTA se encuentra desbalanceados en ambos dataset.



También investigamos entre los features seleccionados, si teníamos ausencia de datos o "NaN". Encontramos que los features Naf y fsodio presentaban valores

faltantes en el dataset original. En ambos dataset, original y masculino, la falta de datos representaban menos del 3% del total de datos. Por lo que la estrategia empleada para darle solución a este problema fue borrar las entradas que tuviera “NaN” en estos features.

Ambos dataset fueron separados usando *sklearn.train_test_split*, empleando una semilla aleatoria establecida para todo el práctico (*random_state=42*) y una división Train/test = 70/30.

Seleccionar un modelo de clasificación clásico

Se seleccionó el **árbol de decisión/*Decision Tree Classifier*** ya que presenta buen desempeño para bases de datos desbalanceadas y con mucha dimensionalidad. Además es fácil de visualizar e interpretar.

Como métrica elegimos **f1** por tratarse de una base de datos desbalanceada.

A. Para el dataset original

Versión default del Arbol de Decisión (AD)

Con el AD sobre los valores de test obtuvimos las siguientes métricas. Se observa que la opción 1 (no HTA), comparado a la opción 2 (si HTA), presenta un mejor valor de f1.

	precision	recall	f1-score	support
1	0.98	0.97	0.98	856
2	0.94	0.95	0.95	393

Versión con hiperparametros optimizados del Arbol de Decisión

Empleamos cross-validation = 5 y GridSearchCV para optimizar los siguientes parámetros:

- 'criterion': ['gini', 'entropy'],
- 'splitter': ['best', 'random'],
- 'max_features': [None, 'auto', 'sqrt', 'log2'],
- 'max_depth':[None, 8, 16],
- 'min_samples_split': range(2, 10),
- 'min_samples_leaf':range(1,6)}

Obtuvimos que la mejor configuración del árbol de decisión es:

- *ccp_alpha=0.0,*
- *class_weight=None,*
- *criterion='gini',*
- *max_depth=8,*
- *max_features=None,*
- *max_leaf_nodes=None,*

- *min_impurity_decrease=0.0,*
- *min_impurity_split=None,*
- *min_samples_leaf=4,*
- *min_samples_split=9,*
- *min_weight_fraction_leaf=0.0,*
- *presort='deprecated',*
- *random_state=42,*
- *splitter='random'*

Con este AD optimizado sobre los valores de test obtuvimos las siguientes métricas. Se observa una mejora en la precisión de la opción 1 (no HTA), comparado al AD por default. Además, la opción 2 (si HTA), presenta una mejora en el recall que incrementa el valor de f1 a 0.96.

Este aumento en f1 es bueno para nosotros, ya que particularmente nos interesa, porque mejorar la predicción de las personas con HTA.

	precision	recall	f1-score	support
1	0.99	0.97	0.98	856
2	0.94	0.99	0.96	393

Además, el *Accuracy* del AD con parámetros optimizados, con Validación Cruzada presentó los siguientes 5 valores [0.968 0.928 0.956 0.96 0.9437751]. Lo que representa una media de *Accuracy* de 0.95 y una desviación estándar de 0.01.

B. Para el dataset Masculino

Versión default del Árbol de Decisión (AD)

Con el AD sobre los valores de test obtuvimos las siguientes métricas. Se observa que la opción 1 (no HTA), comparado a la opción 2 (si HTA), presenta un mayor valor de f1.

	precision	recall	f1-score	support
1	0,96	0,99	0,97	336
2	0,98	0,92	0,95	192

Versión con hiperparametros optimizados del Árbol de Decisión

Empleamos cross-validation = 5 y GridSearchCV para optimizar los mismos 6 parámetros mencionados arriba.

Obtuvimos que la mejor configuración del árbol de decisión es:

- *ccp_alpha=0.0,*
- *class_weight=None,*
- *criterion='gini',*
- *max_depth=None,*
- *max_features=None,*

- *max_leaf_nodes=None,*
- *min_impurity_decrease=0.0,*
- *min_impurity_split=None,*
- *min_samples_leaf=4,*
- *min_samples_split=2,*
- *min_weight_fraction_leaf=0.0,*
- *presort='deprecated',*
- *random_state=42,*
- *splitter='best'*

Con este AD optimizado sobre los valores de test obtuvimos las siguientes métricas. Se observa una mejora en el f1 de la opción 1 (no HTA), comparado al AD por default. Además, la opción 2 (si HTA), presenta una mejora en el recall.

	precision	recall	f1-score	support
1	0.96	0.99	0.98	336
2	0.98	0.93	0.95	192

Además, el *Accuracy* del AD con parámetros optimizados, con Validación Cruzada presentó los siguientes 5 valores [0.99056604 0.98113208 0.96226415 0.98095238 0.97142857]. Lo que representa una media de *Accuracy* de 0.977 y una desviación estándar de 0.010.

Seleccionar un modelo de clasificación utilizando una técnica de ensemble

Se seleccionó el ***boosting/GradientBoostingClassifier*** ya que nos permitirá emplear AD que darán diferentes pesos a los datos de entrenamiento. Al tratarse de AD, presenta buen desempeño para bases de datos desbalanceadas y con mucha dimensionalidad. Además, es fácil de visualizar e interpretar.

Mantenemos la métrica de **f1**, así comparamos con la performance de los AD anteriores.

C. Para el dataset original

Versión default del Boosting

Con el boosting sobre los valores de test obtuvimos las siguientes métricas. Se observa que la opción 1 (no HTA), comparado a la opción 2 (si HTA), presenta un mejor valor de f1.

Obtenemos valores idénticos al AD optimizado de arriba.

	precision	recall	f1-score	support
1	0.99	0.97	0.98	856
2	0.94	0.98	0.96	393

Versión con hiperparametros optimizados del Boosting

Empleamos cross-validation = 5 y GridSearchCV para optimizar los siguientes parámetros:

- "loss":["deviance","exponential"],
- "learning_rate": [0.01, 0.025, 0.05, 0.1],
- "max_depth":[3,5,8,16].

Obtuvimos que la mejor configuración del Boosting es:

- ccp_alpha=0.0,
- criterion='friedman_mse',
- init=None,
- learning_rate=0.05,
- loss='exponential',
- max_depth=3,
- max_features=None,
- max_leaf_nodes=None,
- min_impurity_decrease=0.0,
- min_impurity_split=None,
- min_samples_leaf=1,
- min_samples_split=2,
- min_weight_fraction_leaf=0.0,
- n_estimators=100,
- n_iter_no_change=None,
- presort='deprecated',
- random_state=42,
- subsample=1.0,
- tol=0.0001,
- validation_fraction=0.1,
- verbose=0,
- warm_start=False.

Con este Boosting optimizado sobre los valores de test obtuvimos las siguientes métricas. Se observa una mejora en la precisión de la opción 1 (no HTA), comparado al AD por default. Además, la opción 2 (si HTA), presenta una mejora en el recall.

	precision	recall	f1-score	support
1	1,00	0.97	0.98	856
2	0.94	0.99	0.96	393

Además, el *Accuracy* del Boosting con parámetros optimizados, con Validación Cruzada presentó los siguientes 5 valores [0.988 0.984 0.968 0.984 0.960]. Lo que representa una media de *Accuracy* de 0.98 y una desviación estándar de 0.01.

D. Para el dataset Masculino

Versión default del Boosting

Con el Boosting sobre los valores de test obtuvimos las siguientes métricas. Se observa que la opción 1 (no HTA), comparado a la opción 2 (si HTA), presenta un mayor valor de f1, pero ambos son valores muy altos y mejores que un AD optimizado.

	precision	recall	f1-score	support
1	0.99	0.99	0.99	336
2	0.97	0.98	0.98	192

Versión con hiperparametros optimizados del Árbol de Decisión

Empleamos cross-validation = 5 y GridSearchCV para optimizar los mismos 6 parámetros mencionados arriba.

Obtuvimos que la mejor configuración del árbol de decisión es:

- ccp_alpha=0.0,
- criterion='friedman_mse',
- init=None,
- learning_rate=0.01,
- loss='exponential',
- max_depth=3,
- max_features=None,
- max_leaf_nodes=None,
- min_impurity_decrease=0.0,
- min_impurity_split=None,
- min_samples_leaf=1,
- min_samples_split=2,
- min_weight_fraction_leaf=0.0,
- n_estimators=100,
- n_iter_no_change=None,
- presort='deprecated',
- random_state=42,
- subsample=1.0,
- tol=0.0001,
- validation_fraction=0.1,
- verbose=0,
- warm_start=False.

Con este Boosting optimizado sobre los valores de test obtuvimos las siguientes métricas. Se observa que la opción 1 (no HTA) no mejora pero tampoco que queda mucho margen para mejorar. Además, la opción 2 (si HTA), presenta una mejora en el recall.

	precision	recall	f1-score	support
1	0.99	0.99	0.99	336
2	0.97	0.99	0.98	192

Además, el *Accuracy* del Boosting con parámetros optimizados, con Validación Cruzada presentó los siguientes 5 valores [0.99, 1.0 ,0.97, 0.98, 0.98]. Lo que representa una media de *Accuracy* de 0.98 y una desviación estándar de 0.01.

La mejor configuración encontrada

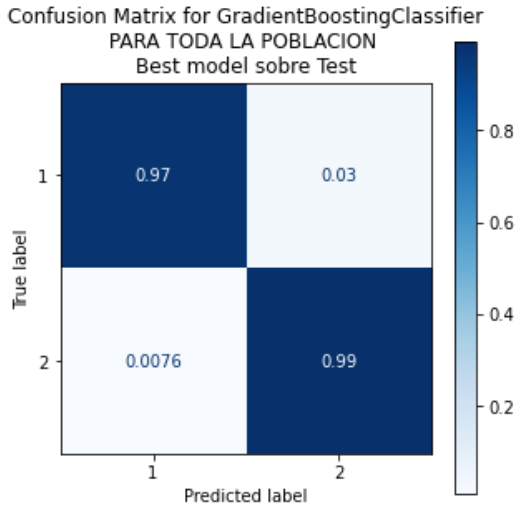
A. Para el dataset original

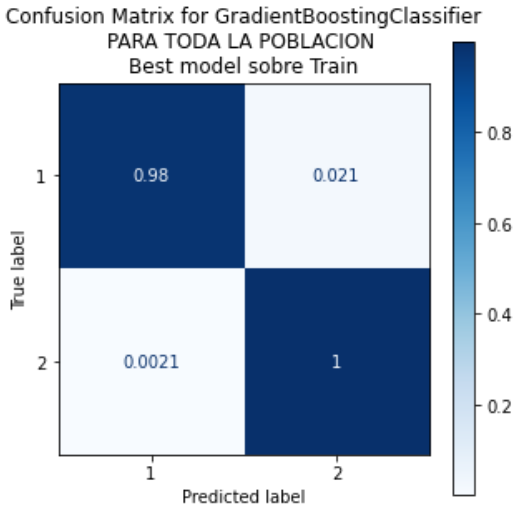
El mejor modelo corresponde al **Boosting optimizado**. Si bien presenta los mismos valores de f1 score que el AD optimizado, el Boosting mencionado tiene el mejor valor de precisión (1.00) para la opción 1, lo cual anula la posibilidad de falsos positivos a mi modelo.

Aquí muestro los valores de Train del mejor modelo.

	precision	recall	f1-score	support
1	1,00	0.98	0.99	1968
2	0.96	1,00	0.98	945

Matrices de confusión sobre los datos de train y test





Como es de esperar, con los datos de train se obtienen los valores son más altos en las métricas, ya que con ellos hizo el entrenamiento. Lo valioso de este modelo de Boosting optimizado es el gran desempeño con los datos de test.

Aquí se muestra la importancia de las variables para el modelo

	variable	importancia para el modelo
3	hta	0.48
27	ten2min	0.37
26	ten2max	0.11
33	fsodio	0.01
35	fque	0.01

Se puede observar que la variable **hta**, *Presencia de hipertensión arterial,* es la *feature* más importante para llevar a cabo la predicción por parte del modelo. Luego le siguen **ten2min** (se anota el valor mínimo de tensión arterial), **ten2max** (se anota el valor máximo de tensión arterial), **fsodio** (registro del consumo de sodio) y **fque** (cantidad de veces que se consume queso al mes). Esto nos lleva plantear que:

1. Que las variables mencionadas dejan fácilmente al descubierto la predicción de HTA.
2. Que las 37 features restantes del dataset son prescindibles.

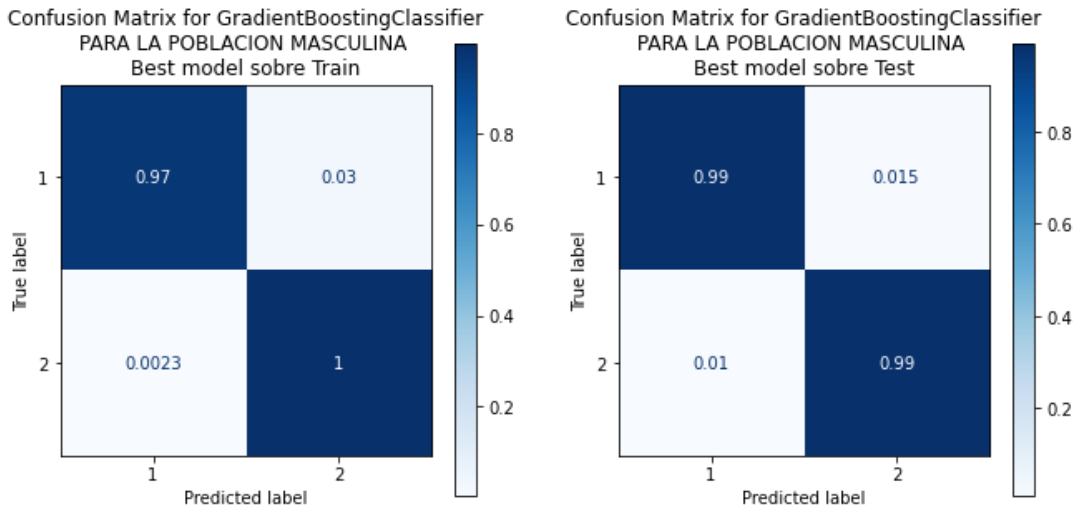
E. Para el dataset Masculino

El mejor modelo corresponde al **Boosting optimizado**. Si bien presenta los mismos valores de f1 score que el **Boosting** default, el ensemble optimizado tiene un mejor valor de recall (0.99) para la opción 2.

Aquí muestro los valores de Train del mejor modelo.

	precision	recall	f1-score	support
1	1,00	0.97	0.98	1968
2	0.95	1,00	0.97	945

Matrices de confusión sobre los datos de train y test



Nuevamente, con los datos de train se obtienen los valores más altos en las métricas. Sin embargo, con los datos de test. se obtienen muy buenos valores e incluso mejores que los resultados del train.

Aquí se muestra la importancia de las variables para el modelo

	variable	importancia para el modelo
2	hta	0.45
26	ten2min	0.39
25	ten2max	0.15
31	Naf	0.00

De manera parecida a lo que ocurre en el dataset original, las variables **hta**, **ten2min, ten2max** y **Naf** (frecuencia de Sodio) son las más importantes. Nuevamente planteamos que:

1. Que las variables mencionadas dejan fácilmente al descubierto la predicción de HTA.
2. Que las 38 features restantes del dataset son prescindibles para este modelo con este dataset.