Práctica 1: Aprendizaje Supervisado

Comparativa de SVM, Árboles y Ensembles sobre el dataset de Obesidad

Jordi Blasco Lozano
DNI: 74527208D
Universidad de Alicante - Escuela Politécnica Superior
Aprendizaje Avanzado - Curso 2025/2026
Email: jbl42@alu.ua.es

Resumen

En esta práctica se evalúan algoritmos de aprendizaje supervisado sobre el mismo problema de clasificación multiclase de la Práctica 0 (niveles de obesidad). Se mantiene exactamente el mismo dataset y el mismo tratamiento de datos para garantizar comparabilidad: eliminación IQR en Weight/Height, transformación Box-Cox en Age, filtrado IQR posterior, codificación binaria + one-hot, split estratificado 80/20 y filtro Z-score en train. Sobre esta base se comparan SVM, árboles de decisión, Random Forest/Extra Trees y Gradient Boosting/AdaBoost mediante validación cruzada 5-fold, métricas weighted y análisis de hiperparámetros. El mejor rendimiento global se obtiene con Gradient Boosting optimizado, alcanzando ***Accuracy = 0.9716*** *y* ***F1-weighted = 0.9718*** *en test.*

1. Introducción

Se reutiliza el dataset *Estimation of Obesity Levels Based on Eating Habits and Physical Condition* (UCI) para comparar directamente con la Práctica 0. El objetivo aquí no es reexplicar el preprocesado, sino analizar de forma compacta el comportamiento de varios modelos supervisados y sus hiperparámetros sobre la misma base experimental.

2. Configuración Experimental

2.1. Pipeline de datos heredado de Práctica 0

Se aplicó el mismo flujo de limpieza y codificación para mantener paridad experimental. El control de paridad obtenido se resume en la Tabla 1.

Tabla 1: Check de paridad del preprocesamiento respecto a Práctica 0

raw_rows	rows_after_weight_height	rows_after_age	lambda_age_boxcox	split_train	split_test	rows_removed_train_zscore
2111	2109	2107	-1.6415	1685	422	0

Interpretación breve: los conteos y la transformación de Age coinciden con la práctica anterior ($2111 \rightarrow 2109 \rightarrow 2107$, $\lambda \approx -1{,}6415$, split 1685/422, sin eliminaciones adicionales por Z-score en train).

2.2. EDA resumido

Se mantuvieron las mismas visualizaciones clave (distribuciones, boxplots, correlación, scatter, balance de clases, PCA), con texto reducido. Como recordatorio visual:

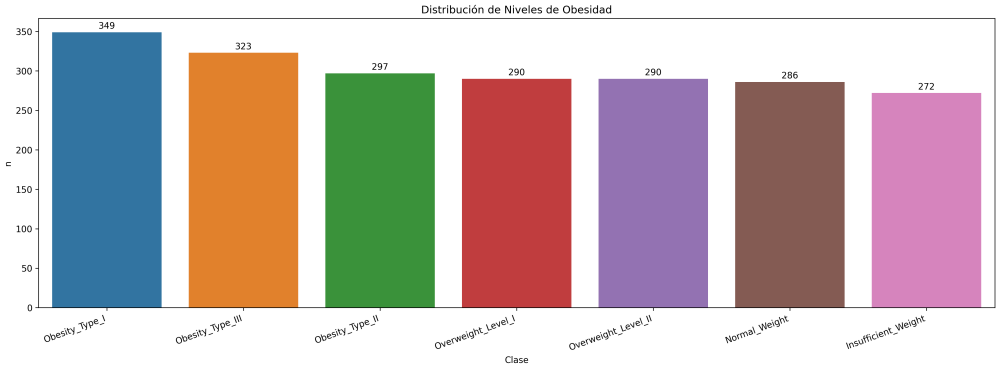

Figura 1: Distribución de clases tras el tratamiento de datos (mismo pipeline de P0).

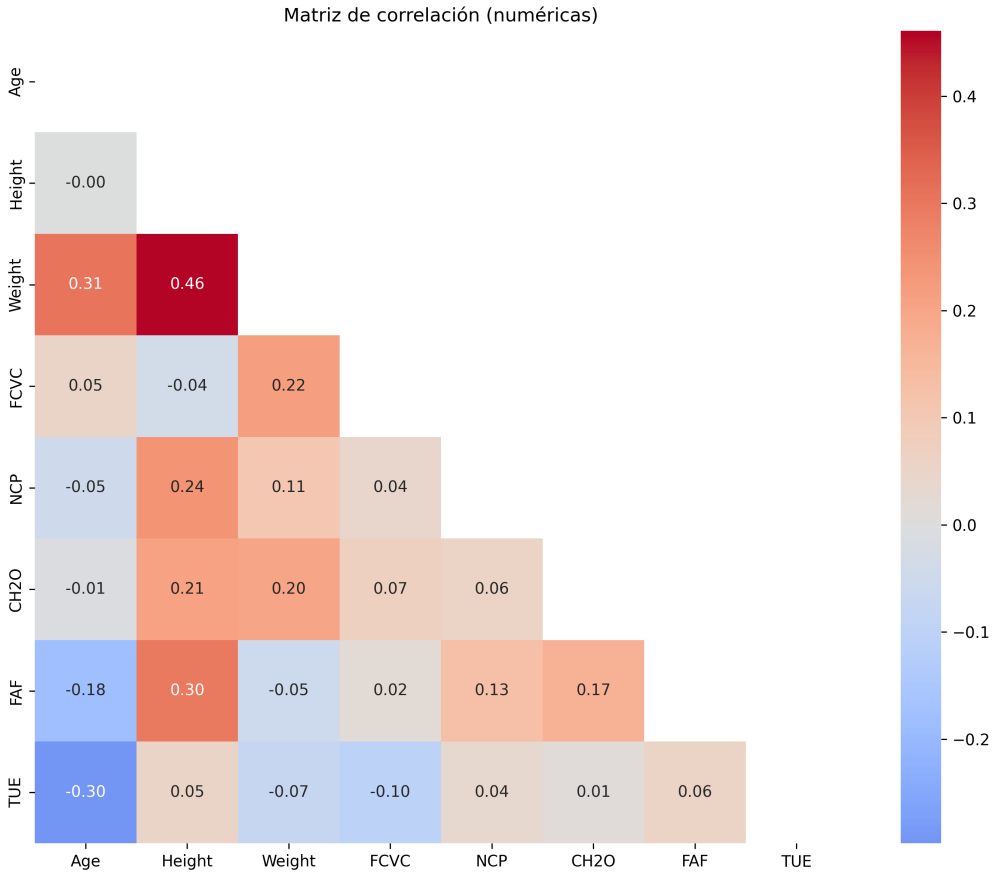

Figura 2: Matriz de correlación de variables numéricas.

3. Resultados

3.1. Parte 1: Support Vector Machines

Se evaluaron kernels por defecto, optimización de RBF y polinomial, comparación de implementaciones lineales y análisis de vectores de soporte.

Tabla 2: SVM con kernels por defecto

kernel	cv_accuracy_mean	cv_accuracy_std	test_accuracy	precision_w	recall_w	f1_w	n_sv	pct_training_sv	fit_time_s
Linear	0.9418	0.0249	0.9384	0.9391	0.9384	0.9381	565	33.5312	0.1032
RBF	0.8528	0.0232	0.8720	0.8807	0.8720	0.8747	1089	64.6291	0.1122
Poly (d=3)	0.7454	0.0205	0.7441	0.7612	0.7441	0.7389	1202	71.3353	0.1239
Sigmoid	0.6409	0.0320	0.6517	0.6495	0.6517	0.6501	1217	72.2255	0.0845

Tabla 3: Grid representativo de SVM RBF (C, γ)

C	gamma	cv_accuracy	test_accuracy	n_sv	pct_training_sv	fit_time_s	cv_fit_time_s
100.0000	0.010000	0.9401	0.9431	629	37.3294	0.0612	0.0838
100.0000	0.001000	0.9223	0.9408	879	52.1662	0.0711	0.0894
100.0000	auto	0.9086	0.9218	799	47.4184	0.0858	0.1174
100.0000	scale	0.9086	0.9218	799	47.4184	0.0801	0.1315
10.0000	scale	0.9080	0.9100	850	50.4451	0.0840	0.0985
10.0000	auto	0.9080	0.9100	850	50.4451	0.0798	0.0991
10.0000	0.100000	0.8896	0.8981	943	55.9644	0.1118	0.1381
1.0000	scale	0.8528	0.8720	1089	64.6291	0.0945	0.1186
100.0000	1	0.7858	0.7915	1260	74.7774	0.2357	0.3315
0.1000	0.001000	0.1656	0.1659	1685	100.0000	0.2906	0.2182

Tabla 4: Top-12 configuraciones de SVM polinomial

degree	C	gamma	cv_accuracy	test_accuracy	n_sv
3	10.0000	0.1	0.8718	0.8886	789
3	1.0000	0.1	0.8641	0.8768	897
3	10.0000	auto	0.8605	0.8791	916
3	10.0000	scale	0.8605	0.8791	916
2	10.0000	0.1	0.8588	0.8460	784
2	10.0000	auto	0.8475	0.8555	900
2	10.0000	scale	0.8475	0.8555	900
4	1.0000	0.1	0.8392	0.8436	960
4	10.0000	0.1	0.8356	0.8460	856
2	1.0000	0.1	0.8309	0.8389	946
4	10.0000	auto	0.8131	0.8175	1048
4	10.0000	scale	0.8131	0.8175	1048

Tabla 5: Comparación de implementaciones lineales

implementation	cv_accuracy_mean	cv_accuracy_std	test_accuracy	n_sv	fit_time_s
SVC(linear)	0.9418	0.0249	0.9384	565.0000	0.0678
LinearSVC	0.7632	0.0350	0.7867	nan	4.2676
SGDClassifier	0.7062	0.0313	0.6943	nan	0.0434

Tabla 6: Análisis de vectores de soporte por configuración

configuracion	n_sv	pct_training_set
RBF (C=100, gamma=1)	1260	74.7774
Poly (d=3, C=1)	1202	71.3353
RBF (C=1, gamma=scale)	1089	64.6291
RBF (C=10, gamma=0.1)	943	55.9644
Linear (C=1)	565	33.5312

Tabla 7: Resumen final Parte 1 (mejor configuración por familia)

modelo	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	n_sv	pct_training_sv
SVM Linear (mejor C)	0.9614	0.0090	0.9716	0.9715	271	16.0831
SVM RBF (mejor config)	0.9401	0.0182	0.9431	0.9432	629	37.3294
SVM Poly (mejor config)	0.8718	0.0239	0.8886	0.8885	789	46.8249
SVM Sigmoid (mejor config)	0.6409	0.0320	0.6517	0.6501	1217	72.2255

Hallazgos: el mejor SVM fue **lineal** (test 0,9716, F1 0,9715) y además con menor porcentaje de SV (16.08 %). En RBF, la mejor zona fue C alto y γ bajo-moderado.

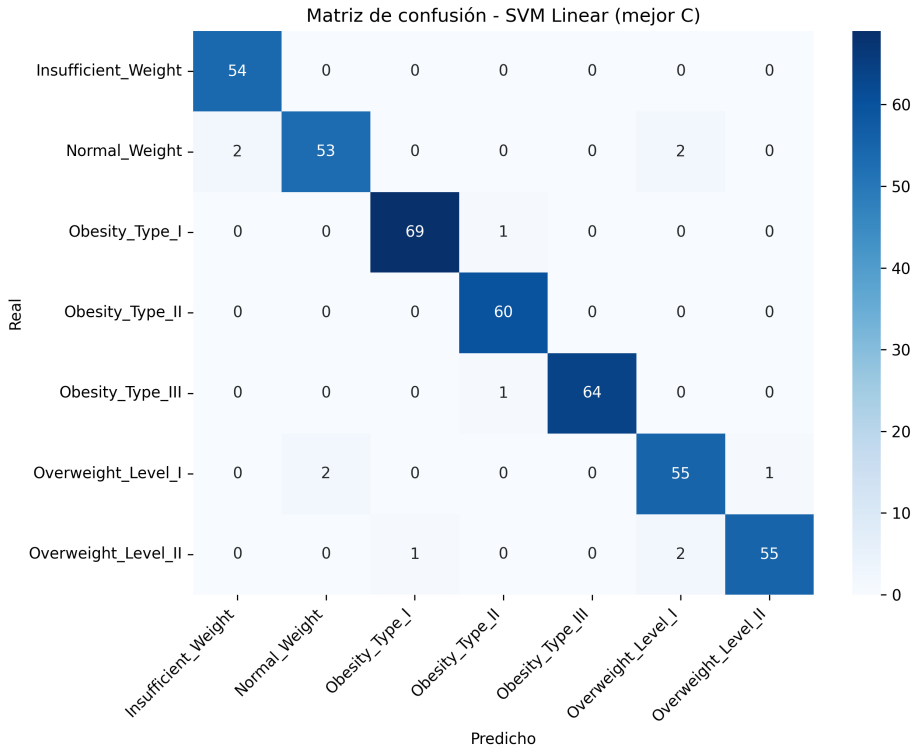

Figura 3: Matriz de confusión del mejor modelo SVM de la parte 1.

3.2. Parte 2: Árboles de Decisión

Se analizó árbol baseline, profundidad, criterio, poda previa (grid), poda posterior (*ccp_alpha*), importancia y visualización.

Tabla 8: Árbol sin restricciones (baseline)

configuracion	train_accuracy	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	depth	n_leaves	fit_time_s
Sin restricciones	1.0000	0.9240	0.0110	0.9289	0.9297	12	101	0.0102

Tabla 9: Barrido de profundidad máxima

max_depth	train_accuracy	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	depth_real	n_leaves
10	0.9988	0.9252	0.0174	0.9289	0.9297	10	98
None	1.0000	0.9240	0.0110	0.9289	0.9297	12	101
15	1.0000	0.9228	0.0129	0.9289	0.9297	12	101
7	0.9454	0.8855	0.0258	0.9005	0.9007	7	57
5	0.8528	0.8231	0.0154	0.8199	0.8185	5	25
3	0.6433	0.6291	0.0062	0.6564	0.6447	3	8
2	0.5543	0.5531	0.0055	0.5450	0.4287	2	4
1	0.2908	0.2908	0.0027	0.2915	0.1429	1	2

Tabla 10: Comparación de criterios de división

criterion	cv_accuracy_mean	cv_accuracy_std	test_accuracy	n_leaves
entropy	0.9418	0.0115	0.9597	79
log_loss	0.9418	0.0115	0.9597	79
gini	0.9240	0.0110	0.9289	101

Tabla 11: Top-12 configuraciones de poda previa (GridSearch)

max_depth	min_samples_split	min_samples_leaf	max_leaf_nodes	cv_accuracy_mean	cv_accuracy_std	rank_test_score
NaN	2	2	NaN	0.9288	0.0126	1
NaN	5	1	NaN	0.9276	0.0139	2
NaN	5	2	NaN	0.9270	0.0175	3
10	2	2	NaN	0.9252	0.0160	4
10	2	1	NaN	0.9252	0.0174	5
10	5	1	NaN	0.9252	0.0124	5
10	5	2	NaN	0.9240	0.0196	7
NaN	2	1	NaN	0.9240	0.0110	7
NaN	2	2	50	0.9228	0.0171	9
10	2	2	50	0.9223	0.0173	10
NaN	10	1	NaN	0.9205	0.0183	11
NaN	5	2	50	0.9205	0.0156	12

Tabla 12: Valores representativos de poda posterior (*ccp_alpha*)

ccp_alpha	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	depth	n_leaves
0.0000	0.9240	0.0110	0.9289	0.9297	12	101
0.0010	0.9282	0.0115	0.9265	0.9274	10	70
0.0011	0.9252	0.0128	0.9265	0.9274	10	67
0.0019	0.9223	0.0218	0.9313	0.9313	9	43
0.0048	0.8955	0.0262	0.9194	0.9188	8	25
0.0912	0.5027	0.0641	0.4313	0.3215	2	3

Tabla 13: Top-10 importancias de features (árbol)

rank	feature	importance_gini
1	Weight	0.5066
2	Height	0.2470
3	Gender	0.1603
4	Age	0.0248
5	FAVC	0.0238
6	CH2O	0.0101
7	CAEC_Sometimes	0.0076
8	TUE	0.0047
9	NCP	0.0038
10	CAEC_Frequently	0.0023

Tabla 14: Resumen final Parte 2

modelo	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	depth	n_leaves
Sin restricciones	0.9240	0.0110	0.9289	0.9297	12	101
Mejor poda posterior	0.9282	0.0115	0.9265	0.9274	10	70
Mejor poda previa	0.9288	0.0126	0.9218	0.9224	11	90

Hallazgos: aunque la mejor CV aparece en poda previa/posterior, el baseline sin restricciones mantiene test competitivo (0.9289). En criterio, `entropy/log_loss` superan a `gini` en esta práctica.

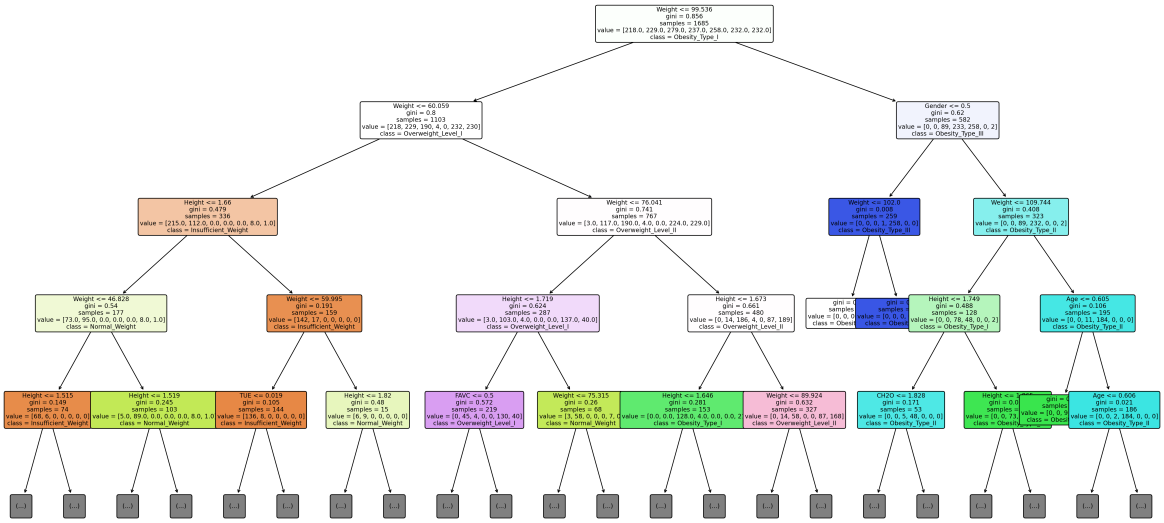

Figura 4: Visualización del árbol (profundidad limitada para legibilidad).

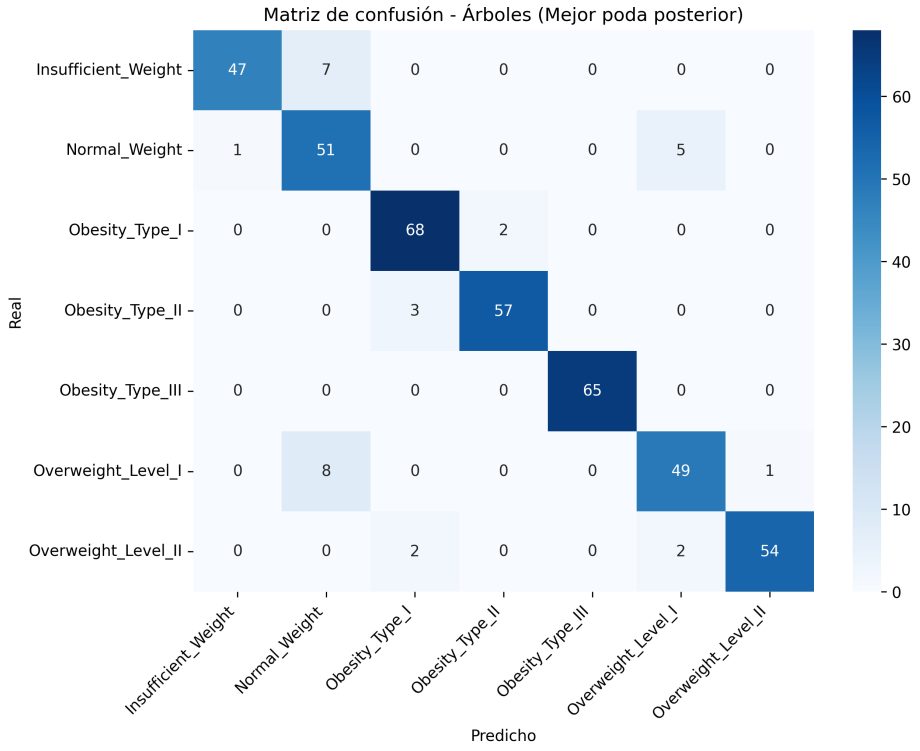

Figura 5: Matriz de confusión del mejor árbol en la parte 2.

3.3. Parte 3: Random Forest

Se evaluó RF por defecto, barridos de *n_estimators* y *max_features*, grid search, importancias (Gini+Permutation) y Extra Trees.

Tabla 15: Random Forest por defecto

configuracion	cv_accuracy_mean	cv_accuracy_std	oob_score	test_accuracy	f1_w	fit_time_s
RF default (n=100)	0.9389	0.0189	0.9407	0.9479	0.9490	0.4784

Tabla 16: Efecto del número de estimadores

n_estimators	oob_score	test_accuracy	fit_time_s
10	0.8427	0.9076	0.0638
25	0.9086	0.9360	0.1006
50	0.9276	0.9479	0.1654
100	0.9407	0.9479	0.2056
200	0.9442	0.9479	0.3891
500	0.9454	0.9526	0.9214

Tabla 17: Efecto de `max_features`

max_features	oob_score	test_accuracy
0.5	0.9591	0.9550
None	0.9543	0.9526
sqrt	0.9407	0.9479
log2	0.9407	0.9479

Tabla 18: Top-12 configuraciones de RF (GridSearch)

n_estimators	max_features	max_depth	min_samples_leaf	cv_accuracy_mean	cv_accuracy_std	rank_test_score
200	NaN	NaN	1	0.9519	0.0168	1
200	NaN	20	1	0.9519	0.0168	1
100	NaN	20	1	0.9507	0.0170	3
100	NaN	NaN	1	0.9507	0.0170	3
200	NaN	10	1	0.9496	0.0168	5
200	NaN	NaN	2	0.9484	0.0182	6
200	NaN	20	2	0.9484	0.0182	6
100	NaN	10	1	0.9478	0.0158	8
100	NaN	NaN	2	0.9472	0.0193	9
100	NaN	20	2	0.9472	0.0193	9
200	NaN	10	2	0.9472	0.0191	11
100	NaN	10	2	0.9460	0.0191	12

Tabla 19: Top-10 importancias de features en RF

rank	feature	gini_importance	perm_importance_mean	perm_importance_std
1	Weight	0.4585	0.7384	0.0187
2	Height	0.2463	0.3116	0.0154
3	Gender	0.1583	0.1720	0.0139
4	Age	0.0388	0.0581	0.0076
5	FAVC	0.0235	0.0057	0.0022
6	FAF	0.0128	-0.0021	0.0027
7	FCVC	0.0099	-0.0005	0.0026
8	CH2O	0.0089	0.0028	0.0028
9	TUE	0.0086	0.0021	0.0029
10	NCP	0.0086	0.0045	0.0027

Tabla 20: Resumen final Parte 3

modelo	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	oob_score	fit_time_s
Random Forest mejor config	0.9519	0.0168	0.9550	0.9553	0.9573	0.4140
Random Forest default	0.9389	0.0189	0.9479	0.9490	0.9407	0.4784
Decision Tree (baseline)	0.9240	0.0110	0.9289	0.9297	nan	0.0102
Extra Trees	0.9163	0.0205	0.9265	0.9277	nan	0.1174

Hallazgos: la mejor configuración RF alcanza 0,9550 en test y mejora al RF por defecto. El ranking de importancia confirma el peso dominante de `Weight` y `Height`.

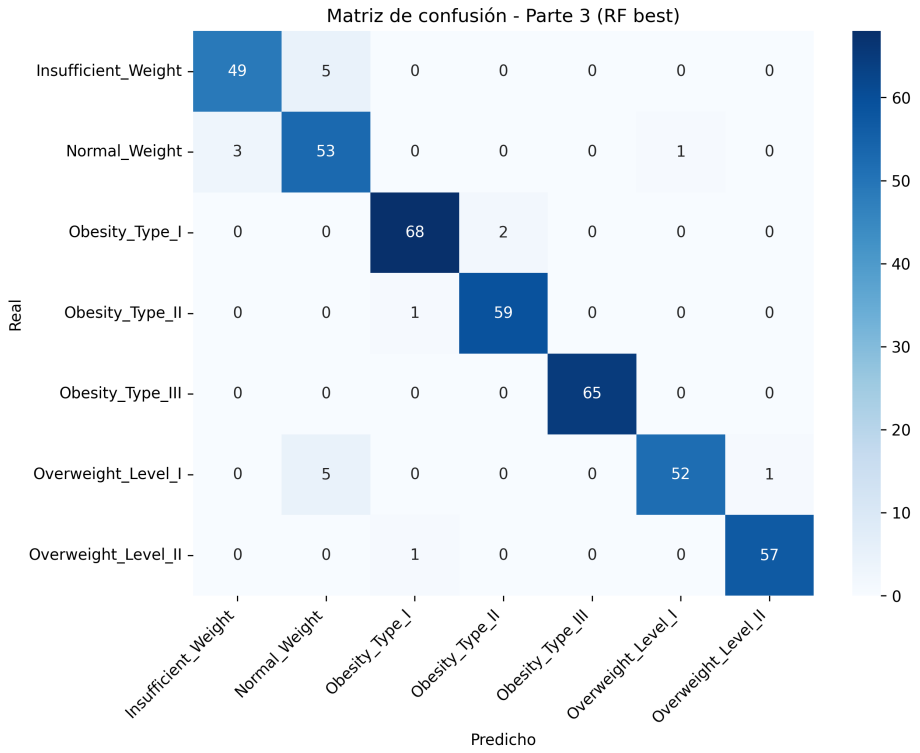

Figura 6: Matriz de confusión del mejor modelo de la parte 3.

3.4. Parte 4: Gradient Boosting y AdaBoost

Se analizó GB por defecto, barridos $(n\_estimators, learning\_rate)$, profundidad, grid search y AdaBoost con stump.

Tabla 21: Gradient Boosting por defecto

modelo	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	fit_time_s
Gradient Boosting default	0.9561	0.0181	0.9668	0.9674	2.6888

Tabla 22: Efecto conjunto de *n_estimators* y *learning_rate*

n_estimators	learning_rate	cv_accuracy_mean	cv_accuracy_std	test_accuracy
200	0.1000	0.9608	0.0142	0.9716
50	0.5000	0.9567	0.0121	0.9550
100	0.1000	0.9561	0.0181	0.9668
50	0.1000	0.9460	0.0184	0.9431
100	0.0500	0.9430	0.0208	0.9455
200	0.0100	0.9068	0.0187	0.9005

Tabla 23: Efecto de `max_depth` en GB

max_depth	train_accuracy	cv_accuracy_mean	cv_accuracy_std	test_accuracy
5	1.0000	0.9567	0.0144	0.9597
3	1.0000	0.9561	0.0181	0.9668
2	0.9905	0.9389	0.0104	0.9408
1	0.8789	0.8433	0.0083	0.8412

Tabla 24: Top-12 configuraciones de GB (GridSearch)

n_estimators	learning_rate	max_depth	subsample	cv_accuracy_mean	cv_accuracy_std	rank_test_score
200	0.100000	3	1.000000	0.9608	0.0142	1
200	0.500000	3	1.000000	0.9585	0.0101	2
200	0.100000	3	0.800000	0.9585	0.0111	2
200	0.050000	3	0.800000	0.9585	0.0130	2
200	0.050000	3	1.000000	0.9579	0.0148	5
100	0.500000	3	0.800000	0.9573	0.0118	6
100	0.500000	3	1.000000	0.9573	0.0089	6
200	0.500000	3	0.800000	0.9567	0.0132	8
100	0.100000	3	1.000000	0.9561	0.0181	9
100	0.100000	3	0.800000	0.9549	0.0087	10
100	0.500000	2	0.800000	0.9543	0.0127	11
100	0.500000	2	1.000000	0.9543	0.0137	11

Tabla 25: Barrido de AdaBoost

configuracion	n_estimators	learning_rate	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	fit_time_s
AdaBoost (n=100, lr=0.5)	100	0.5000	0.3899	0.0257	0.4265	0.3233	0.2989
AdaBoost (n=50, lr=1.0)	50	1.0000	0.3442	0.0099	0.3483	0.2519	0.1480
AdaBoost (n=100, lr=1.0)	100	1.0000	0.3193	0.0345	0.3483	0.2519	0.2959
AdaBoost (n=200, lr=0.1)	200	0.1000	0.2843	0.0083	0.2844	0.1825	0.5932

Tabla 26: Resumen final Parte 4

modelo	cv_accuracy_mean	cv_accuracy_std	test_accuracy	f1_w	fit_time_s
Gradient Boosting mejor config	0.9608	0.0142	0.9716	0.9718	5.4175
Gradient Boosting default	0.9561	0.0181	0.9668	0.9674	2.6888
Random Forest (mejor config)	0.9519	0.0168	0.9550	0.9553	0.4140
AdaBoost (mejor config)	0.3899	0.0257	0.4265	0.3233	0.2989

Hallazgos: Gradient Boosting optimizado fue el mejor de esta parte (test 0,9716, F1 0,9718). En este dataset, AdaBoost con stump quedó muy por debajo del resto.

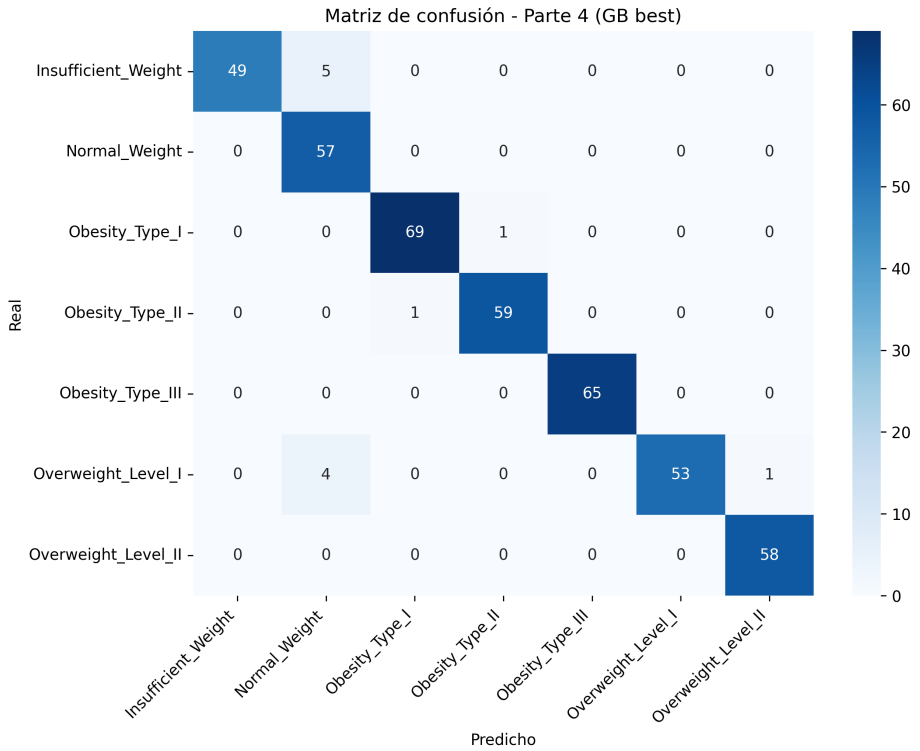

Figura 7: Matriz de confusión del mejor modelo de la parte 4.

4. Comparativa Global

La comparación final de mejores modelos por familia se muestra en la Tabla 27. El mejor modelo global fue **Gradient Boosting**.

Tabla 27: Comparativa final global de modelos

modelo	cv_accuracy_mean	cv_accuracy_std	test_accuracy	precision_w	recall_w	f1_w	fit_time_s
Gradient Boosting	0.9608	0.0142	0.9716	0.9745	0.9716	0.9718	5.5777
Random Forest	0.9519	0.0168	0.9550	0.9568	0.9550	0.9553	0.4411
SVM (mejor config)	0.9401	0.0182	0.9431	0.9440	0.9431	0.9432	0.0659
Decision Tree	0.9282	0.0115	0.9265	0.9308	0.9265	0.9274	0.0103
AdaBoost	0.3899	0.0257	0.4265	0.2759	0.4265	0.3233	0.3135

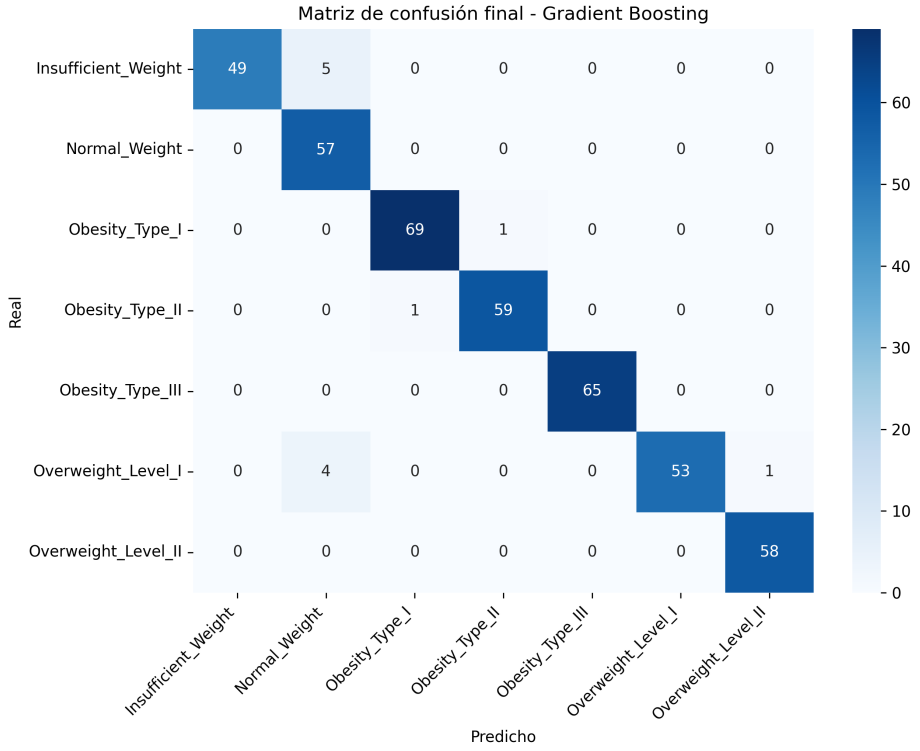

Figura 8: Matriz de confusión del mejor modelo global (Gradient Boosting).

5. Conclusiones

- Mantener exactamente el pipeline de P0 permitió comparar modelos en igualdad de condiciones.
- En SVM, el kernel lineal fue superior al resto en este problema tras optimización de C.
- En árboles, la poda mejoró la estabilidad en CV, pero el baseline siguió siendo competitivo en test.
- Random Forest mejoró claramente a un árbol individual y confirmó la relevancia de `Weight/Height`.
- El mejor resultado global fue Gradient Boosting optimizado (Accuracy 0.9716, F1-weighted 0.9718).