

# Clasificación de niveles de Super Mario Maker 2 por su dificultad

Patricio Eugenio Cantú Treviño

Noviembre 2025

## 1. Introducción

Creatividad ilimitada es una de las mejores descripciones que le puedes dar al juego Super Mario Maker 2, ya que en este juego se le permite a los jugadores crear sus propios niveles utilizando las mecánicas de la franquicia de Super Mario, y compartirlos por internet por el resto del mundo.

Cada nivel publicado le es asignado una de 4 dificultades: Fácil, Normal, Experto y Súper Experto. No es completamente claro que es lo que define la dificultad de cada nivel, puede tener cierta relación con el ratio de éxito promedio que se tienen los jugadores al jugar el nivel, pero esta no es una división exacta.

En este artículo, buscaremos encontrar la mejor definición posible para la clasificación de los niveles de dificultad. La información que salga de este análisis podrá ayudar a hacer ingeniería inversa sobre los métodos que usa el juego para estas definiciones.

## 2. Descripción de los datos

Se trabajara con un conjunto de datos [4] que contiene información de niveles de Super Mario Maker 2 de hasta el año 2022. El conjunto de datos [4] contiene mas de 26 millones de muestras, pero para este análisis se usara una muestra aleatoria de 10,000 muestras.

En el cuadro 1 se ven las variables que se usarán para el análisis y su descripción, las variables no numéricas o categóricas (a excepción de `difficulty`) fueron eliminadas.

También añadí la variable calculada `unique_clear_rate`, la cual se define como  $\min\{\frac{\text{clears}}{\text{unique.players.and.versus}}, 1\}$ . Añadí esta variable calculada con la creencia de que el porcentaje de compleción del nivel por jugador sería un buen estimador para el nivel de dificultad.

Cuadro 1: Descripciones de las variables del conjunto de datos.

variable	descripción
difficulty	(Categórica) El nivel de dificultad del nivel
uploaded	(Entero) Cuando fue publicado el nivel
created	(Entero) Cuando fue creado el nivel
game_version	(Entero) La versión del juego en la que se creó el nivel
world_record	(Milisegundo) El tiempo récord de compleción
upload_time	(Milisegundo) El tiempo de compleción del nivel del creador
upload_attempts	(Entero) Intentos del creador para publicar el nivel
num_comments	(Entero) Cantidad de comentarios en el nivel
timer	(Segundo) Tiempo límite para completar el nivel
autoscroll_speed	(Entero) Nivel de movimiento automático de la cámara
clears	(Entero) Cantidad de intentos exitosos
attempts	(Entero) Cantidad de intentos totales
clear_rate	(Decimal) Ratio de éxito en las compleciones
plays	(Entero) Cantidad de sesiones de juego del nivel
versus_matches	(Entero) Cantidad de partidas competitivas
coop_matches	(Entero) Cantidad de partidas cooperativas
likes	(Entero) Cantidad de me gusta
boos	(Entero) Cantidad de no me gusta
unique_players_and_versus	(Entero) Cantidad de jugadores que han jugado el nivel

Cuadro 2: Estadísticos descriptivos de las variables.

	minimo	Q1	mediana	Q3	maximo	media	devs_std
uploaded	1621048585.000000	1621060323.250000	1621072574.000000	1621085646.750000	162108964.000000	1621073536.052400	15445.848509
created	1039796700.000000	1620836505.000000	1621075800.000000	1621103220.000000	2868488580.000000	1618842678.642000	40335462.936095
difficulty	0.000000	0.000000	1.000000	1.000000	3.000000	0.878800	0.862430
game_version	0.000000	5.000000	5.000000	5.000000	5.000000	4.917800	0.482977
world_record	-1.000000	9056.250000	19070.000000	39415.750000	1835882.000000	33711.396300	53235.787054
upload_time	433.000000	19295.000000	37502.500000	69960.500000	499590.000000	52789.483700	49968.051316
upload_attempts	1.000000	1.000000	1.000000	3.000000	217.000000	3.894600	9.416770
num_comments	0.000000	0.000000	0.000000	1.000000	872.000000	1.061500	13.508959
timer	10.000000	300.000000	300.000000	300.000000	500.000000	294.961000	113.816053
autoscroll_speed	0.000000	0.000000	0.000000	0.000000	2.000000	0.011000	0.138134
clears	0.000000	3.000000	7.000000	19.000000	13480.000000	23.151400	224.180811
attempts	0.000000	14.000000	35.000000	80.000000	166769.000000	131.536800	2034.905108
clear_rate	0.000000	11.538462	29.561460	54.545455	100.000000	36.136169	29.164521
plays	1.000000	6.000000	14.000000	31.000000	14135.000000	35.647000	279.984010
versus_matches	0.000000	0.000000	0.000000	0.000000	5148.000000	1.750700	70.389901
coop_matches	0.000000	0.000000	0.000000	1.000000	297.000000	0.995300	7.180239
likes	0.000000	0.000000	1.000000	3.000000	3126.000000	4.551400	55.901848
boos	0.000000	0.000000	1.000000	1.000000	641.000000	1.411500	10.454449
unique_players_and_versus	1.000000	6.000000	14.000000	30.000000	10938.000000	32.685700	231.207906
unique_clear_rate	0.000000	0.363636	0.666667	0.976744	1.000000	0.628823	0.316359

### 3. Antecedentes

Furia and Mocci realizaron un análisis de regresión para predecir la dificultad de los niveles de Super Mario Maker 2 [2], ellos se enfocaron en predecir el *clear\_rate*, y el uso del procesamiento de lenguaje natural.

Mientras que Drachen et al. realizaron agrupamiento a jugadores de World of Warcraft segun sus comportamientos [1].

En este análisis me enfoco en predecir la categoría de dificultad a la que cada nivel pertenece, según como los jugadores han interactuado con el. Utilizo tecnicas que ya se han usado con conjuntos de datos similares, pero con un enfoque diferente hacia la clasificación.

### 4. Metodología

#### 4.1. Determinación número de grupos

Me decanté por el método de silueta para seleccionar la cantidad de  $k$  grupos debido su objetividad mas sencilla de comprender. El método de silueta simplemente elige la cantidad de grupos que maximice el coeficiente de silueta, el cual se define como:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Donde  $a(i)$  es la distancia promedio del punto  $i$  a los demás puntos de su propio grupo, y  $b(i)$  es la distancia promedio del punto  $i$  a los puntos del grupo vecino mas cercano.

#### 4.2. Agrupamiento

Drachen et al. obtuvieron resultados interpretables al usar el análisis de arquetipos en sus datos del comportamiento de jugadores en un videojuego [1]. Fue debido a que yo también estoy utilizando datos generados por usuarios en un videojuego que yo me decante por usar el análisis de arquetipos.

##### 4.2.1. Análisis de arquetipos

El análisis de arquetipos forma grupos con los valores mas extremos de los mismos. El análisis de arquetipos busca minimizar la formula: [3]

$$\min \|X - CZ\|$$

Donde  $X \in \mathbb{R}^{N \times d}$  es la matriz con los datos,  $Z \in \mathbb{R}^{k \times d}$  es la matriz con los arquetipos,  $C \in \mathbb{R}^{N \times k}$  es la matriz de coeficientes de  $X$  y  $Z$ ,  $N$  es la cantidad de muestras,  $d$  es la cantidad de dimensiones de las muestras y  $k$  es la cantidad de arquetipos.

### 4.3. Selección de características

#### 4.3.1. Filtrado

Elegí primero filtrar los valores utilizando `SelectPercentile` con la función  $F$  de clasificación. Este método evalúa cada variable independientemente mediante ANOVA univariado, calculando el estadístico  $F$  que mide la relación entre la varianza entre grupos y la varianza dentro de grupos.

Seleccione el percentil 50, reteniendo las características con mayor poder discriminatorio entre niveles de dificultad.

#### 4.3.2. Selección secuencial

Elegí la selección secuencial de características hace adelante como método de envoltura, debido a que genera buenos resultados, de manera mucho mas eficiente.

La selección secuencial hacia adelante, en vez de probar todas las combinaciones posibles, va añadiendo de uno en uno las variables que le dan mas precisión al modelo. Para esta envoltura utilicé el estimador de regresión logística.

Utilicé  $f1$  para la calificación, ya que esta especialmente hecha para la clasificación, y se obtiene mediante:

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

Donde  $TP$  es la cantidad de verdaderos positivos,  $FP$  la de falsos positivos, y  $FN$  la de falsos negativos.

### 4.4. Clasificación

Wheat et al. clasificaron la dificultad percibida por los jugadores en niveles de plataformas [5]. Y obtuvieron sus mejores resultados de clasificación con el bosque aleatorio, así que decidí utilizar los modelos de clasificación de esa misma familia (arbol de decisión, bosque aleatorio, y potenciación de gradiente) para este análisis.

#### 4.4.1. Árbol de decisión

Los árboles de decisión clasifican datos dividiendo el espacio de atributos  $X$  en regiones  $R_m$ . Cada región se asigna a una clase según la proporción de observaciones:

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

La predicción elige la clase más frecuente en la región:

$$k(m) = \arg \max_k p_{mk}$$

Para evaluar la calidad de las divisiones se usan medidas de impureza como el índice Gini:

$$\text{Gini}(R_m) = \sum_{k=1}^K p_{mk}(1 - p_{mk})$$

#### 4.4.2. Bosque aleatorio

Bosque aleatorio es un método de ensamble que construye múltiples árboles de decisión durante el entrenamiento. Cada árbol se entrena con una muestra del conjunto de datos y considera solo un subconjunto aleatorio de  $m$  características en cada división.

La predicción final se obtiene mediante votación mayoritaria:

$$\hat{y} = \text{moda}\{h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_B(\mathbf{x})\}$$

Donde  $h_b(\mathbf{x})$  es la predicción del árbol  $b$  y  $B$  es el número total de árboles. Este enfoque reduce la varianza del modelo y mejora la generalización al promediar las predicciones de múltiples árboles

#### 4.4.3. Potenciación del gradiente

La potenciación del gradiente construye un ensamble de manera secuencial, donde cada nuevo árbol corrige los errores de los anteriores. El modelo final es una suma ponderada:

$$F_M(\mathbf{x}) = \sum_{m=1}^M \nu \cdot h_m(\mathbf{x})$$

Donde  $h_m(\mathbf{x})$  es un árbol débil entrenado para aproximar el gradiente negativo de la función de pérdida, y  $\nu$  es la tasa de aprendizaje que controla la contribución de cada árbol.

## 5. Resultados

### 5.1. Agrupamiento

Se realizó el análisis de arquetipos, y con el método de silueta se llegó a que la cantidad óptima de grupos es 2, así como se muestra en los resultados de la figura 1

Utilizando el análisis de arquetipos, obtenemos que según el método de silueta, 2 es la cantidad de grupos ideal, así como se ve en la figura 1.

La figura 3 muestra la posición de los arquetipos generados. Estos arquetipos fueron utilizados para generar las columnas `dist_archetype_0` y `dist_archetype_1`, las cuales contienen la distancia escalada de la muestra a cada arquetipo.

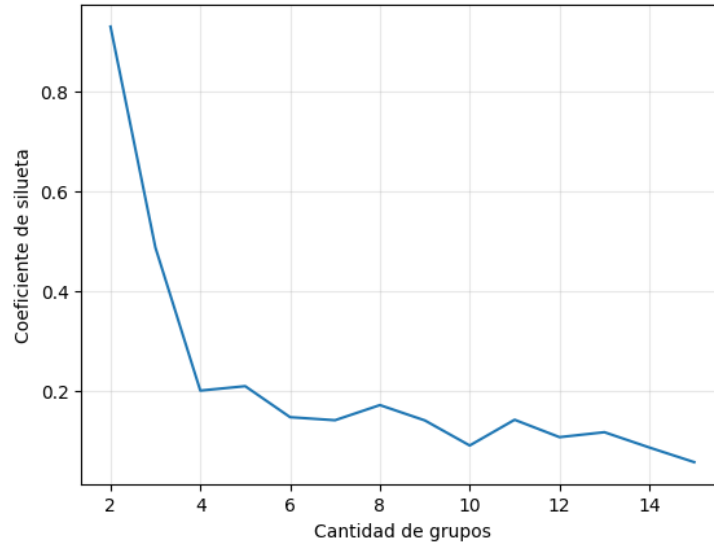


Figura 1: Resultados del método de silueta.

Cuadro 3: Posición de los arquetipos generados.

columnas	arquetipo 0	arquetipo 1
uploaded	1621073311.996442	1621064499.760710
created	1621069388.385310	1618211859.649827
game_version	5.000000	4.937985
world_record	73379.350369	33493.542222
upload_time	84450.556087	56586.190645
upload_attempts	2.492569	4.998671
num_comments	59.369557	0.341264
timer	399.504612	292.147930
autoscroll_speed	0.995046	0.000075
clears	898.714854	11.950100
attempts	1508.373043	74.388268
clear_rate	62.944048	34.944737
plays	1233.543289	16.485513
versus_matches	1812.727027	0.245327
coop_matches	186.847355	0.139380
likes	462.044644	1.181371
boos	225.540396	0.917777
unique_players_and_versus	2927.241496	12.214271
unique_clear_rate	0.521755	0.635226

## 5.2. Selección de características

La filtración inicial de características con `SelectPercentile` resulto en la selección de 10 variables: `world_record`, `upload_time`, `upload_attempts`, `timer`, `clears`, `attempts`, `clear_rate`, `boos`, `unique_clear_rate` y `dist_archetype_1`.

La envoltura posteriormente realizó arrojo los resultados del cuadro 4, aunque técnicamente la combinación de `upload_attempts`, `timer`, y `clear_rate` fue la que generó la mayor precisión, la combinación de `upload_attempts` y `clear_rate` tiene una diferencia del menos del 1 % (82.15 % vs 81.81 %).

Debido a que la diferencia de precisión entre las dos opciones es muy pequeña, me decante por la segunda opción debido a que al utilizar unicamente 2 variables sera mas fácil realizar los análisis posteriores.

Cuadro 4: Los resultados de la selección secuencial de características.

	avg_score	feature_names
3	0.821842	upload_attempts, timer, clear_rate
9	0.821597	world_record, upload_time, upload_attempts, timer, clears, attempts, clear_rate, boos, unique_clear_rate
8	0.821495	world_record, upload_time, upload_attempts, timer, clears, attempts, clear_rate, boos
4	0.821313	upload_attempts, timer, clear_rate, boos
5	0.821139	world_record, upload_attempts, timer, clear_rate, boos
6	0.820701	world_record, upload_time, upload_attempts, timer, clear_rate, boos
7	0.820509	world_record, upload_time, upload_attempts, timer, clears, clear_rate, boos
10	0.819776	world_record, upload_time, upload_attempts, timer, clears, attempts, clear_rate, boos, unique_clear_rate, dist_archetype_1
2	0.818724	upload_attempts, clear_rate
1	0.809867	clear_rate

## 5.3. Clasificación

Con las variables `upload_attempts` y `clear_rate` se generaron los modelos con distintos valores de profundidad máxima. Los modelos de clasificación utilizados (árbol de decisión, bosque aleatorio, y potenciación de gradiente) obtuvieron una precisión del 81.46 %, 82.2 %, y 82.1 % respectivamente.

En la figura 2 se muestra precisión que obtuvieron los modelos utilizados con diferentes niveles de profundidad máximos. En este caso el bosque aleatorio con una máxima profundidad de 6 fue el que generó la mejor precisión.

Las regiones de decisión de los modelos (cada uno con su profundidad máxima optima) se pueden ver en la figura 3. Y las matrices discriminantes de la figura 4 nos define mejor los puntos fuertes del modelo a la hora de clasificar..

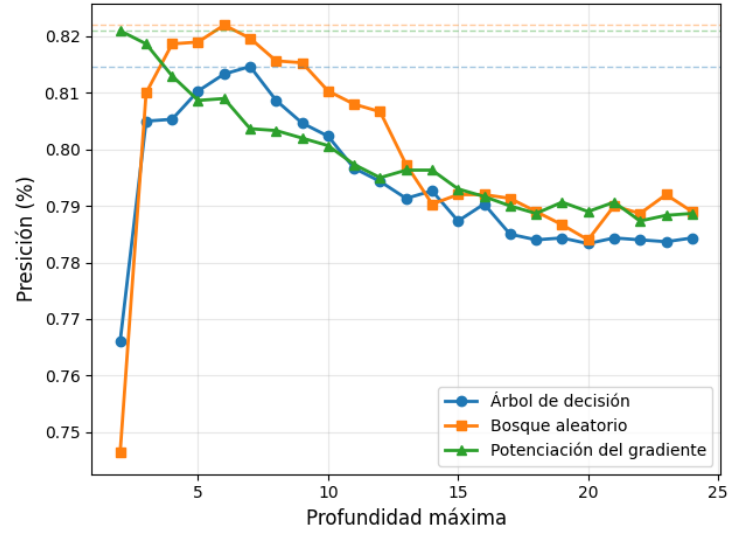


Figura 2: Precisión obtenida por los modelos de clasificación según su nivel máximo de profundidad.

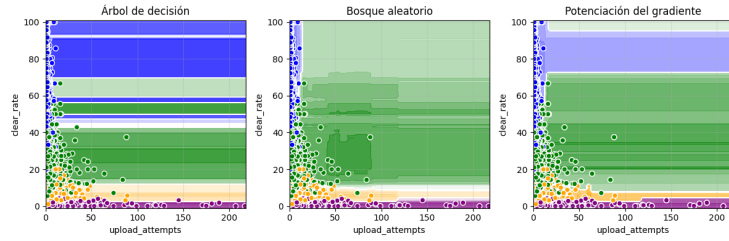


Figura 3: Dispersión de los datos de los modelos de clasificación usados.

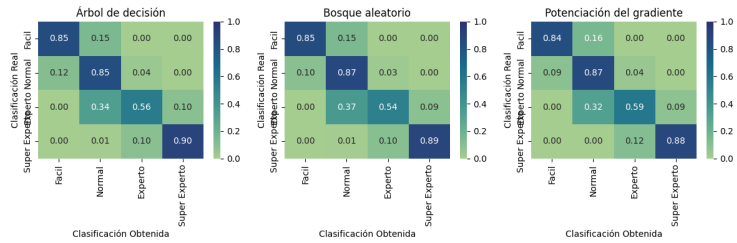


Figura 4: Matrices de discriminación con los resultados de los modelos de clasificación usados.



## 6. Discusión

### 6.1. Agrupamiento

La literatura de los datos decía que la cantidad optima de grupos debería de ser 4, debido a que esa es la cantidad de dificultades que queremos clasificar. Pero el método de silueta arrojó que la cantidad optima es de 2 grupos, y por una diferencia substancial.

Ademas, las variables generadas con la agrupación no pasaron la selección de características, por lo que no tuvo un efecto en el resultado de la clasificación.

### 6.2. Clasificación

Cosas importantes que me sorprendieron de los resultados de clasificación, son que los 3 modelos obtuvieron precisiones bastantes similares, separadas por menos de un 1%, y que la precisión de la potenciación del gradiente bajaba mientras mayor era la profundidad máxima.

## 7. Conclusiones

En este análisis se encontraron muy buenas aproximaciones para la clasificación de la dificultad de los niveles de Super Mario Maker 2. Aunque todos los modelos tuvieron precisiones similares, el bosque aleatorio fue el modelo con la mayor precisión.

Es importante notar que la mayor complicación para el modelo fue clasificar los niveles de dificultad experto, ya que estos comparten características con los niveles normales.

Aunque no se encontró una medida exacta y objetiva de que da el nivel de dificultad en este análisis, si logramos definir las variables con mayor correlación a esta clasificación.

## Referencias

- [1] A. Drachen, C. Thureau, R. Sifa, and C. Bauckhage. A comparison of methods for player clustering via behavioral telemetry. *arXiv preprint arXiv:1407.3950*, 2014.
- [2] C. A. Furia and A. Mocci. What makes a level hard in super mario maker 2? In *2025 IEEE Conference on Games (CoG)*, page 1–8. IEEE, Aug. 2025. doi: 10.1109/cog64752.2025.11114357. URL <http://dx.doi.org/10.1109/cog64752.2025.11114357>.
- [3] E. Gomedé. Deciphering Data Extremes: A Practitioner’s Guide to Archetypal Analysis, 4 2024.

- [4] TheGreatRambler. mm2\_level. [https://huggingface.co/datasets/TheGreatRambler/mm2\\_level](https://huggingface.co/datasets/TheGreatRambler/mm2_level), 2022.
- [5] D. Wheat, M. Masek, C. P. Lam, and P. Hingston. Modeling perceived difficulty in game levels. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACSW '16, page 1–8. ACM, Feb. 2016. doi: 10.1145/2843043.2843478. URL <http://dx.doi.org/10.1145/2843043.2843478>.