

Licenciatura en Informática - UNLP

Trabajo Final - Matemática 4

En el siguiente [link](#) de Google Drive, se encuentran los archivos “*TP_E1aE2.do*” y “*TP_E3.py*” con los códigos que se realizaron para los ejercicios 1 y 2 (realizados en Stata) y para el ejercicio 3 (realizado en Python), respectivamente. Adicionalmente, se encuentra el archivo “*TP_E1aE2.log*” con los resultados de Stata de los ejercicios 1 y 2.

Alumnos:

- Leyva Gallarati, Emiliano (Legajo 19527/6).
- Menduiña, Juan Ignacio (Legajo 24981/9).

Un grupo de estudiantes avanzados y graduados de la Facultad de Informática ha fundado una empresa de soluciones tecnológicas. Recientemente, fueron contactados por la empresa de figuritas PANINOS, quien les encargó el desarrollo de una solución informática basada en análisis de datos.

PANINOS ha realizado un estudio de mercado que le permitió concluir que incentivar un segmento digital de tarjetas de jugadores de fútbol en formato NFT (Non-Fungible Token) ampliará su base de clientes con un gasto mínimo en publicidad. Además, este modelo incentivará la impresión de nuevas colecciones físicas de figuritas. Para dar inicio a este nuevo proyecto, PANINOS ha decidido utilizar como base a los jugadores del videojuego FIFA 21, lo que les permitirá hacer un primer piloto digital.

Con un dataset extraído de Kaggle, que incluye más de 100 atributos por jugador del FIFA 21, resulta de interés poder cuantificar los valores aceptables de algunas de esas características, tales como el valor de mercado, cláusula de rescisión, pases, velocidad, entre otros. Esto permitirá identificar jugadores nuevos que sean compatibles con los actuales y, si el piloto tiene éxito, eventualmente, crear un parche para el FIFA 21 que incluya los jugadores más relevantes.

El dataset se encuentra en el archivo “players_21.csv”, donde se registran diversas características de los jugadores, junto con su valor de mercado (“value_eur”). Con base en estos datos, el CEO les solicita desarrollar una aplicación que cumpla con los siguientes requisitos:

Parte 1: Predicción del Valor de Mercado.

Recta de regresión para predecir el valor de mercado de un jugador a partir de la característica más relevante (a la que se destinará mayor proporción del presupuesto), respaldada por:

- (i) Prueba de significancia de regresión, coeficiente de determinación (R^2) y correlación lineal (r).
- (ii) Inferencias sobre los parámetros de la recta, estimando las fluctuaciones con una confianza del 95%.
- (iii) La proporción de veces que el valor de mercado supera la incertidumbre de predicción comparada con la respuesta media del valor de mercado para una característica fija, ambas con la misma confianza y ancho mínimo.

Para seleccionar la característica más relevante (a la que se destinará mayor proporción del presupuesto) para predecir el valor de mercado de un jugador, se seleccionaron 42 variables explicativas¹ y se realizaron regresiones lineales simples con cada una de estas variables, a fin de determinar cuál estimación es la que posibilita un mayor coeficiente de determinación (R^2). Éste fue el criterio que se eligió para elección de la característica más relevante.

El resultado de este análisis iterativo resulta en que la variable que posibilita que se “maximice” el R^2 de la estimación por Mínimos Cuadrados Ordinarios (MCO) es “movement_reactions”. Cabe aclarar que, al realizar esta estimación con una sola variable explicativa, se puede tener sesgo por variables omitidas, lo cual invalidaría este análisis de predicción. Por eso, será relevante incluir otras variables más adelante. Por ahora, el modelo a estimar será el siguiente:

$$value_eur_i = \beta_0 + \beta_1 movement_reactions_i + \varepsilon_i, \quad (1)$$

donde ε será el término de error del modelo, que se supone tiene $E(\varepsilon) = 0$ y $\text{Var}(\varepsilon) = \sigma^2$.

En la tabla 1, se presenta la estimación por MCO (robusta a la presencia de heterocedasticidad) de “value_eur” en función sólo de “movement_reactions” (modelo (1)). Se puede observar que esta variable explicativa es estadísticamente significativa ($p\text{-value} = 0,000$), considerando un nivel de significancia bajo (usualmente, $\alpha = 0,01$). Se observa también que el R^2 tiene un valor de 0,2448, que es relativamente alto a la hora de explicar cuánto de la variabilidad de la variable dependiente se explica por la

¹ age, height_cm, pace, shooting, passing, dribbling, defending, physic, attacking_crossing, attacking_finishing, attacking_heading_accuracy, attacking_short_passing, attacking_volleys, skill_dribbling, skill_curve, skill_fk_accuracy, skill_long_passing, skill_ball_control, movement_acceleration, movement_sprint_speed, movement_agility, movement_reactions, movement_balance, power_shot_power, power_jumping, power_stamina, power_strength, power_long_shots, mentality_aggression, mentality_interceptions, mentality_positioning, mentality_vision, mentality_penalties, mentality_composure, defending_marking_awareness, defending_standing_tackle, defending_sliding_tackle, goalkeeping_diving, goalkeeping_handling, goalkeeping_kicking, goalkeeping_positioning, goalkeeping_reflexes.

variabilidad de la variable independiente (y es, como ya se mencionó, el máximo que surgió entre todas las regresiones). A su vez, el coeficiente de correlación lineal (r) es igual a 0,4947, indicando una relación lineal y positiva entre estas dos variables.

Cuando se observan los intervalos de confianza de 95% de los parámetros estimados de este modelo (ordenada al origen y pendiente), se tiene que estos son [-2,45e+07; -2,15e+07] y [394935,2; 445897,5], respectivamente, indicando que, si se tomaran distintas muestras aleatorias y se calculara el intervalo de confianza de 95%, el 95% de los intervalos resultantes contendrán a β_1 .

Tabla 1. Estimación por MCO del modelo (1) (en niveles).

Linear regression		Number of obs	=	18,707	
		F(1, 18705)	=	1045.86	
		Prob > F	=	0.0000	
		R-squared	=	0.2448	
		Root MSE	=	6.7e+06	
<hr/>					
		Robust			
value_eur	Coefficient	std. err.	t	P> t	
-----	-----	-----	-----	[95% conf. interval]	
movement_reactions	420416.3	12999.98	32.34	0.000	394935.2 445897.5
_cons	-2.30e+07	759341.1	-30.26	0.000	-2.45e+07 -2.15e+07
<hr/>					

Fuente: Elaboración propia.

En la tabla 2, se presenta la estimación por MCO del modelo (1), pero, ahora, en logaritmos. De esta manera, se puede pensar a los coeficientes como elasticidades (cambio porcentual de la variable dependiente ante un cambio de 1% de la variable independiente). Siendo así, un cambio de 1% en la variable independiente indica que, en promedio, la variable dependiente varía en $\beta_1\%$. En particular, en esta estimación, el parámetro estimado $\hat{\beta}_1 = 5,887$ indica que el valor de mercado (*value_eur*) aumenta, en promedio, 5,887% cuando *movement_reactions* aumenta en 1%.

Tabla 2. Estimación por MCO del modelo (1) (en logaritmos).

Linear regression		Number of obs	=	18,707	
		F(1, 18705)	=	11510.99	
		Prob > F	=	0.0000	
		R-squared	=	0.5304	
		Root MSE	=	.85487	
<hr/>					
		Robust			
ln_value_eur	Coefficient	std. err.	t	P> t	
-----	-----	-----	-----	[95% conf. interval]	
ln_movement_reactions	5.887088	.0548712	107.29	0.000	5.779536 5.994641
_cons	-10.30905	.2248	-45.86	0.000	-10.74968 -9.868422
<hr/>					

Fuente: Elaboración propia.

Por último, se puede mencionar que el porcentaje de veces que el valor de mercado se encuentra por fuera de los límites establecidos, por un lado, por el intervalo de predicción y, por otro lado, por el intervalo de estimación es 3,56% y 99,39%, respectivamente, considerando la misma confianza (95%) y ancho mínimo ($x^* = \bar{x}$). En particular, se puede mencionar lo siguiente:

- **Intervalo de Predicción:** Un 3,56% de los valores de mercado se encuentran por fuera del intervalo de predicción, lo cual indica que el modelo es relativamente preciso para predecir los valores de mercado para un dato particular de la variable explicativa. Este porcentaje es bastante bajo y sugiere que el modelo está capturando bien la variabilidad de los valores de mercado.
- **Intervalo de Confianza:** Un 99,39% de los valores de mercado se encuentran fuera del intervalo de confianza, lo cual es extremadamente alto y sugiere que el modelo puede no estar proporcionando una estimación confiable de la media del valor de mercado cuando se considera el efecto de la variable explicativa. Esto puede indicar que hay mucha variabilidad en los datos o que el modelo no está bien especificado. Puede que el modelo no esté capturando, adecuadamente, la relación entre las variables, lo que lleva a estimaciones imprecisas de los parámetros y, por ende, a intervalos de confianza inexactos. Probablemente, hay variables adicionales que no fueron incluídas en el modelo y puedan estar influyendo en el valor de mercado.

Este resultado es razonable desde el punto de vista estadístico, pero indica que podría haber problemas en la especificación del modelo o en la variabilidad de los datos.

Parte 2: Ecuación de Predicción del Valor de Mercado.

Ecuación para predecir el valor de mercado del jugador a partir de varias características.

(i) Usando el método de mínimos cuadrados. Explicar los indicadores obtenidos (como el coeficiente de determinación y la correlación) y proporcionar una breve interpretación de los resultados.

Ahora, el modelo a estimar será el siguiente:

$$Y = X\beta + \varepsilon, \quad (2)$$

donde Y es el vector ($n \times 1$) de observaciones de la variable explicada “*value_eur*”, X es la matriz ($n \times K$) de observaciones de las variables explicativas, β es el vector ($K \times 1$) de parámetros y ε es el vector ($n \times 1$) de errores del modelo, que se supone tiene $E(\varepsilon) = 0$ y $\text{Var}(\varepsilon) = \sigma^2$. En otras palabras:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}; \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k-1} \\ 1 & x_{21} & x_{22} & \dots & x_{2k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk-1} \end{bmatrix}; \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}; \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

En la tabla 3, se presenta la estimación por MCO (robusta a la presencia de heterocedasticidad) de “*value_eur*” en función de 42 variables explicativas (modelo (2)). Se puede observar que las variables explicativas “*age*”, “*height_cm*”, “*skill_ball_control*”, “*attacking_volleys*”, “*movement_reactions*”, “*movement_balance*” y “*mentality_composure*” son estadísticamente significativas al 1% (p-value= 0,000). Se observa también que el R^2 tiene un valor de 0,3467, que es mayor al obtenido en la regresión lineal simple. A su vez, el test F de significatividad conjunta indica que las variables explicativas son, en conjunto, estadísticamente significativas (Prob > F= 0,000).

Tabla 3. Estimación por MCO del modelo (2) (en niveles).

Linear regression		Number of obs		=	16,654
		F(42, 16611)		=	41.32
		Prob > F		=	0.0000
		R-squared		=	0.3467
		Root MSE		=	6.3e+06
		Robust			
		value_eur	Coefficient	std. err.	t
					P> t
					[95% conf. interval]
		age	-401562.4	18313.21	-21.93
		height_cm	68606.26	12284.32	5.58
		pace	-71818.65	162653.2	-0.44
		shooting	25176.86	141085.6	0.18
		passing	260432	156132.6	1.67
		dribbling	-160449.9	141644.7	-1.13
		defending	75991.24	160308.9	0.47
		physic	173970.8	166331	1.05
		attacking_crossing	-44254.15	32716.57	-1.35
		attacking_finishing	20798.94	63796.18	0.33
		attacking_heading_accuracy	31827.41	18243.34	1.74
		attacking_short_passing	-49785.96	56740.08	-0.88
		attacking_volleys	28552.83	9993.563	2.86
		skill_dribbling	65612.16	72131.19	0.91
		skill_curve	6145.396	9849.451	0.62
		skill_fk_accuracy	-14525.38	10690.53	-1.36
		skill_long_passing	-55690.27	24381	-2.28
		skill_ball_control	128420.6	43454.27	2.96
		movement_acceleration	74589.02	73157.05	1.02
		movement_sprint_speed	83638.45	90469.28	0.92
		movement_agility	-37451.62	15297.21	-2.45
		movement_reactions	340049.4	17361.26	19.59
		movement_balance	56522.92	10283.42	5.50
		power_shot_power	-21582.75	29641.27	-0.73
		power_jumping	-8640.068	10113.96	-0.85
		power_stamina	-46640.09	41530.44	-1.12
		power_strength	-70216.94	83301.5	-0.84
		power_long_shots	-49942.87	28762.32	-1.74
		mentality_aggression	-52193.63	33941.78	-1.54
		mentality_interceptions	-22177.65	32803.92	-0.68
		mentality_positioning	-21689.39	9304.737	-2.33
		mentality_vision	-18808.27	31867.05	-0.59
		mentality_penalties	3925.47	11823.33	0.33
		mentality_composure	97434.81	6789.912	14.35
		defending_marking_awareness	-7852.276	49452.45	-0.16
		defending_standing_tackle	4628.344	50950.66	0.09
		defending_sliding_tackle	-7201.992	20577.78	-0.35
		goalkeeping_diving	-12401.26	15889.49	-0.78
		goalkeeping_handling	-30114.84	15896.82	-1.89
		goalkeeping_kicking	2639.234	16366.88	0.16
		goalkeeping_positioning	-24784.22	16182.06	-1.53
		goalkeeping_reflexes	3689.173	15994.84	0.23
		_cons	-4.25e+07	2705425	-15.70

Fuente: Elaboración propia.

En la tabla 4, se presenta la estimación por MCO del modelo (2), pero, ahora, en logaritmos. Se puede observar que mejora mucho la significatividad estadística de todas las variables explicativas. Además, se menciona la interpretación de algunos parámetros estimados. En particular, dada la muestra y la estimación, el valor de mercado (*value_eur*):

- disminuye, en promedio, 2,36% cuando *age* aumenta en 1%, manteniendo todo lo demás constante;
- aumenta, en promedio, 0,61% cuando *height_cm* aumenta en 1%, manteniendo todo lo demás constante;
- aumenta, en promedio, 2,68% cuando *passing* aumenta en 1%, manteniendo todo lo demás constante;
- aumenta, en promedio, 5,12% cuando *dribbling* aumenta en 1%, manteniendo todo lo demás constante; y

- aumenta, en promedio, 4,98% cuando *physic* aumenta en 1%, manteniendo todo lo demás constante.

Tabla 4. Estimación por MCO del modelo (2) (en logaritmos).

Linear regression		Number of obs		=	16,654
		F(42, 16611)	=	964.67	
		Prob > F	=	0.0000	
		R-squared	=	0.8018	
		Root MSE	=	.54207	
		Robust			
	ln_value_eur	Coefficient	std. err.	t	P> t [95% conf. interval]
	ln_age	-2.361012	.0346135	-68.21	0.000 -2.428858 -2.293166
	ln_height_cm	.6131998	.2066801	2.97	0.003 .2080848 1.018315
	ln_pace	-1.851138	1.177367	-1.57	0.116 -4.158902 .4566263
	ln_shooting	.6669645	.7074923	0.94	0.346 -.7197959 2.053725
	ln_passing	2.675911	.8664785	3.09	0.002 .9775209 4.374302
	ln_dribbling	5.118618	1.686567	3.03	0.002 1.812766 8.42447
	ln_defending	-.4759095	.4125548	-1.15	0.249 -1.284561 .3327419
	ln_physic	4.981496	.409657	12.16	0.000 4.178525 5.784468
	ln_attacking_crossing	-.1806801	.1537784	-1.17	0.240 -.4821023 .120742
	ln_attacking_finishing	-.2138668	.2819648	-0.76	0.448 -.7665479 .3388143
	ln_attacking_heading_accuracy	.701418	.0680086	10.31	0.000 .5681139 .834722
	ln_attacking_short_passing	-.0804688	.3577953	-0.22	0.822 -.7817858 .6208482
	ln_attacking_volleys	.0193944	.0370641	0.52	0.601 -.0532551 .0920439
	ln_skill_dribbling	-2.004697	.7598425	-2.64	0.008 -3.49407 -.5153247
	ln_skill_curve	-.0953579	.0436937	-2.18	0.029 -.1810232 -.0097347
	ln_skill_fk_accuracy	-.0914871	.041727	-2.19	0.028 -.1732764 -.0096977
	ln_skill_long_passing	-.5986136	.1139055	-5.26	0.000 -.8218805 -.3753466
	ln_skill_ball_control	.0627132	.575088	0.11	0.913 -.1064521 1.189947
	ln_movement_acceleration	1.418654	.5330678	2.66	0.008 .3737845 2.463524
	ln_movement_sprint_speed	1.604871	.6437256	2.49	0.013 .3431004 2.866642
	ln_movement_agility	-.8066354	.1744892	-4.62	0.000 -.148653 -.4646179
	ln_movement_reactions	3.283156	.1259557	26.07	0.000 3.036269 3.530042
	ln_movement_balance	-.3226021	.0835942	-3.86	0.000 -.4864556 -.1587486
	ln_power_shot_power	.0987171	.1730951	0.57	0.568 -.2405677 .4380019
	ln_power_jumping	-.2587978	.0331741	-7.80	0.000 -.3238226 -.193773
	ln_power_stamina	-.7759571	.1067204	-7.27	0.000 -.9851405 -.5667737
	ln_power_strength	-2.186762	.1991683	-10.98	0.000 -.2577153 -.1.796371
	ln_power_long_shots	-.454553	.1419253	-3.20	0.001 -.7327418 -.1763641
	ln_mentality_aggression	-.8423128	.0744053	-11.32	0.000 -.9881552 -.6964704
	ln_mentality_interceptions	-.017247	.0768126	-0.22	0.822 -.1678079 .1333139
	ln_mentality_positioning	-.4409349	.0508895	-8.66	0.000 -.5406837 -.3411862
	ln_mentality_vision	-.7585432	.1500682	-5.05	0.000 -.1052693 -.4643936
	ln_mentality_penalties	.1285966	.0465895	2.76	0.006 .0372763 .2199169
	ln_mentality_composure	1.472157	.0442008	33.31	0.000 1.385519 1.558795
	ln_defending_marking Awareness	.2932968	.1236894	2.37	0.018 .0508523 .5357413
	ln_defending_standing_tackle	.2848335	.1148652	2.48	0.013 .0596853 .5099816
	ln_defending_sliding_tackle	.0007524	.0451511	0.02	0.987 -.0877485 .0892533
	ln_goalkeeping_diving	-.0092254	.0137007	-0.67	0.501 -.0360803 .0176295
	ln_goalkeeping_handling	-.0181163	.0137115	-1.32	0.186 -.0449922 .0087596
	ln_goalkeeping_kicking	-.0249623	.0136387	-1.83	0.067 -.0516956 .0017711
	ln_goalkeeping_positioning	-.0138784	.0138342	-1.00	0.316 -.040995 .0132381
	ln_goalkeeping_reflexes	-.0292647	.0136434	-2.14	0.032 -.0560071 -.0025222
	_cons	-24.1813	1.247051	-19.39	0.000 -.26.62566 -.21.73695

Fuente: Elaboración propia.

(ii) Usando el método de descenso por gradiente. ¿Son los valores obtenidos iguales a los conseguidos mediante la resolución del sistema de ecuaciones normales? Mostrar los resultados obtenidos junto con las últimas iteraciones del algoritmo. Indicar los valores de los parámetros utilizados (como tasa de aprendizaje y número de iteraciones).

Utilizando el método de descenso por gradiente, los valores obtenidos son cercanos a los conseguidos mediante la resolución del sistema de ecuaciones normales (partiendo de valores iniciales para los parámetros cercanos a los estimados por MCO en la tabla 4). Se utiliza una tasa de aprendizaje de $\alpha= 0,001$, un máximo de iteraciones de 1000 y una tolerancia de $\epsilon= 0,000000015$.

En la tabla 5, se presenta la estimación por el método de descenso por gradiente del modelo (2) (en logaritmos). La actualización de parámetros con este método es un proceso iterativo que, con una tasa de aprendizaje adecuada y suficientes iteraciones, debe converger a los mismos resultados que los obtenidos con el método de MCO, que es una solución cerrada. Sin embargo, MCO es una fórmula directa, mientras que el método de descenso por gradiente ajusta los parámetros poco a poco hasta alcanzar esa solución.

Tabla 5. Estimación por el método de descenso por gradiente del modelo (2).

Parámetro	Coeficiente
constante	-30,226624
ln_age	-2,951246
ln_height_cm	0,766528
ln_pace	-2,313898
ln_shooting	0,833730
ln_passing	3,344914
ln_dribbling	6,398298
ln_defending	-0,594867
ln_physic	6,226893
ln_attacking_crossing	-0,225826
ln_attacking_finishing	-0,267309
ln_attacking_heading_accuracy	0,876796
ln_attacking_short_passing	-0,100561
ln_attacking_volleys	0,024267
ln_skill_dribbling	-2,505847
ln_skill_curve	-0,119199
ln_skill_fk_accuracy	-0,114334
ln_skill_long_passing	-0,748242
ln_skill_ball_control	0,078416
ln_movement_acceleration	1,773343
ln_movement_sprint_speed	2,006113
ln_movement_agility	-1,008270
ln_movement_reactions	4,103969
ln_movement_balance	-0,403228
ln_power_shot_power	0,123421
ln_power_jumping	-0,323474
ln_power_stamina	-0,969923
ln_power_strength	-2,733429
ln_power_long_shots	-0,568166
ln_mentality_aggression	-1,052870
ln_mentality_interceptions	-0,021539
ln_mentality_positioning	-0,551144
ln_mentality_vision	-0,948154
ln_mentality_penalties	0,160769
ln_mentality_composure	1,840222
ln_defending_marking_awareness	0,366640
ln_defending_standing_tackle	0,356061
ln_defending_sliding_tackle	0,000958
ln_goalkeeping_diving	-0,011522
ln_goalkeeping_handling	-0,022632
ln_goalkeeping_kicking	-0,031188
ln_goalkeeping_positioning	-0,017334
ln_goalkeeping_reflexes	-0,036569

Fuente: Elaboración propia.

(iii) Dar una interpretación del criterio de corte utilizado en el algoritmo del gradiente. Explicar si presenta alguna falla. Si no es una buena condición de corte, ¿se puede sugerir un criterio alternativo más eficaz?

El criterio de corte utilizado en el algoritmo del gradiente se basa en el Error Cuadrático Medio (ECM), que es una medida comúnmente utilizada en regresión lineal para evaluar la calidad del modelo. Específicamente, la función de costo a minimizar será:

$$\text{Costo} = \frac{1}{2N} \sum_{i=1}^n (\ln \text{value_eur}_i - \ln \widehat{\text{value_eur}}_i)^2.$$

Se tiene en cuenta lo siguiente:

- Error: El error para cada observación se calcula como la diferencia entre el valor observado y el valor predicho por el modelo:
 $\text{error}_i = \ln \text{value_eur}_i - \ln \widehat{\text{value_eur}}_i$.
- Error cuadrático: Para penalizar los errores más grandes de manera más fuerte (debido a que se elevan al cuadrado), se toma el cuadrado de estos errores:
 $\text{error}_i^2 = (\ln \text{value_eur}_i - \ln \widehat{\text{value_eur}}_i)^2$.
- Error cuadrático medio: Para obtener una medida que no dependa del número de observaciones, se promedia el error cuadrático sumando todos los errores al cuadrado y dividiéndolos por el número total de observaciones:
 $\text{ECM} = \frac{1}{N} \sum_{i=1}^N \text{error}_i^2$.
- Función de costo: A menudo, se multiplica el ECM por $\frac{1}{2}$ para simplificar los cálculos en la derivación del algoritmo de descenso por gradiente:
 $\text{Costo} = \frac{1}{2N} \sum_{i=1}^N \text{error}_i^2$.

Cuando la diferencia absoluta entre el valor actual y el valor previo de esta función de costo sea menor a la tolerancia elegida, el algoritmo converge. En otras palabras, si la mejora en el costo es muy pequeña (menor a la tolerancia), el algoritmo asume que ha alcanzado una solución cercana al mínimo y se detiene.

Este tipo de criterio de corte es común porque:

- Controla la convergencia: Si la diferencia entre costos es muy pequeña, indica que el algoritmo ya no está haciendo mejoras significativas y, por lo tanto, se está acercando a la solución óptima.
- Evita ciclos innecesarios: Al tener un umbral de mejora mínimo, se evita que el algoritmo siga iterando indefinidamente cuando ya está cerca de la solución.

Aunque este criterio de corte es intuitivo y funcional en muchos casos, puede presentar problemas en ciertos escenarios:

- Problema de precisión: El criterio depende de la magnitud del cambio en la función de costo. Si la función de costo tiene cambios muy pequeños (por ejemplo, cuando los valores son de magnitudes muy pequeñas), el algoritmo podría detenerse demasiado pronto, incluso antes de llegar al verdadero mínimo global.

- **Sensibilidad a la escala del problema:** Si la función de costo tiene una escala grande o pequeña, la elección del valor de tolerancia puede ser arbitraria y afectar la eficacia del criterio de corte. Un valor de tolerancia que funciona bien para una función de costo puede no ser adecuado para otra con diferente escala.
- **Oscilación o estancamiento:** Si el descenso del gradiente está en una región de la función con oscilaciones o una meseta, el cambio en el costo puede ser muy pequeño durante varias iteraciones. Esto puede hacer que el algoritmo se detenga en un punto subóptimo sin alcanzar el mínimo real.

Un criterio alternativo más eficaz podría ser detener el algoritmo cuando la norma del gradiente sea lo suficientemente pequeña y no cuando la diferencia absoluta entre el valor actual y el valor previo de la función costo sea lo suficientemente pequeña. O bien, se podría utilizar un criterio combinado.

Parte 3: Comportamiento del Método de Descenso por Gradiente.

Convergencia del método de descenso por gradiente. Explicar si el método siempre converge al mínimo de la función. En caso contrario, proporcionar un contraejemplo para ilustrar este comportamiento.

Existen situaciones donde el método de descenso por gradiente no necesariamente converge al mínimo global o puede no converger en absoluto. Esto sucede, típicamente, en las siguientes situaciones:

- Mínimos locales: En funciones no convexas, el método puede converger a un mínimo local en lugar de al mínimo global. Por ejemplo, una función con varios valles puede hacer que el método se quede atrapado en el primero que encuentre.
- Tasa de aprendizaje inadecuada: Si la tasa de aprendizaje es demasiado alta, el método puede saltar sobre el mínimo y nunca acercarse lo suficiente para converger.
- Funciones no diferenciables: En puntos donde la función no es diferenciable, el método de descenso por gradiente puede fallar o no ofrecer una dirección adecuada.

Por ejemplo, se considera la siguiente la función no convexa (llamada función de Rosenbrock):

$$f(x, y) = (a - x)^2 + b(y - x^2)^2.$$

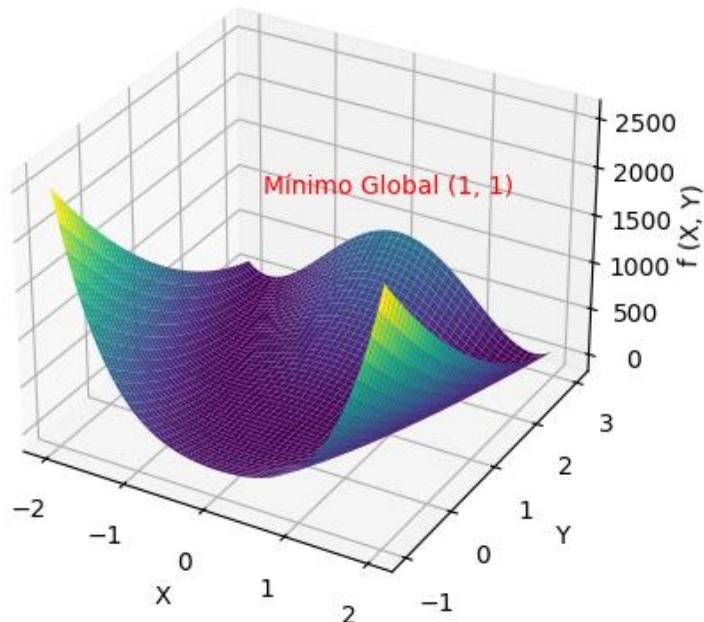
El mínimo global está dentro de un valle plano, largo, estrecho y de forma parabólica. Encontrar el valle es trivial. Sin embargo, converger al mínimo global es difícil. Esta función tiene un mínimo global en (a, a^2) , donde $f(x, y) = 0$.

Las razones por las cuales el método de descenso por gradiente puede no converger con esta función son:

- Curvatura estrecha: La forma de “valle” de la función de Rosenbrock es muy estrecha y larga, lo que puede causar que los pasos de actualización del gradiente sean ineficaces. Si el punto inicial está lejos del mínimo, los pasos de actualización pueden ser demasiado pequeños para moverse, adecuadamente, hacia el mínimo.
- Gradientes pequeños: A medida que el algoritmo se acerca al mínimo, el gradiente puede volverse muy pequeño, lo que resulta en cambios insignificantes en las actualizaciones de los parámetros. Esto puede hacer que el algoritmo se ralentice considerablemente o, incluso, se detenga antes de alcanzar el mínimo global.
- Elección de la tasa de aprendizaje: La tasa de aprendizaje es un hiperparámetro crítico. Si es demasiado alta, el algoritmo puede oscilar y no converger; si es demasiado baja, el algoritmo puede converger muy lentamente.
- Puntos de inicio: Dependiendo de la elección del punto de inicio, el método puede converger a un mínimo local, en lugar de al mínimo global. Aunque en el caso de Rosenbrock no hay mínimos locales (sólo un mínimo global), la trayectoria de descenso puede ser muy ineficiente.

Considerando $a = 1$ y $b = 100$, se tiene la siguiente gráfica:

Función de Rosenbrock (para $a=1$ y $b=100$)



En resumen, la función de Rosenbrock es un ejemplo donde el método de descenso por gradiente puede tener dificultades para converger al mínimo global. Aunque tiene un mínimo global en el punto $(1, 1)$ (cuando $a= 1$), la forma de la función puede llevar a que el algoritmo de descenso por gradiente no converja de manera eficiente.