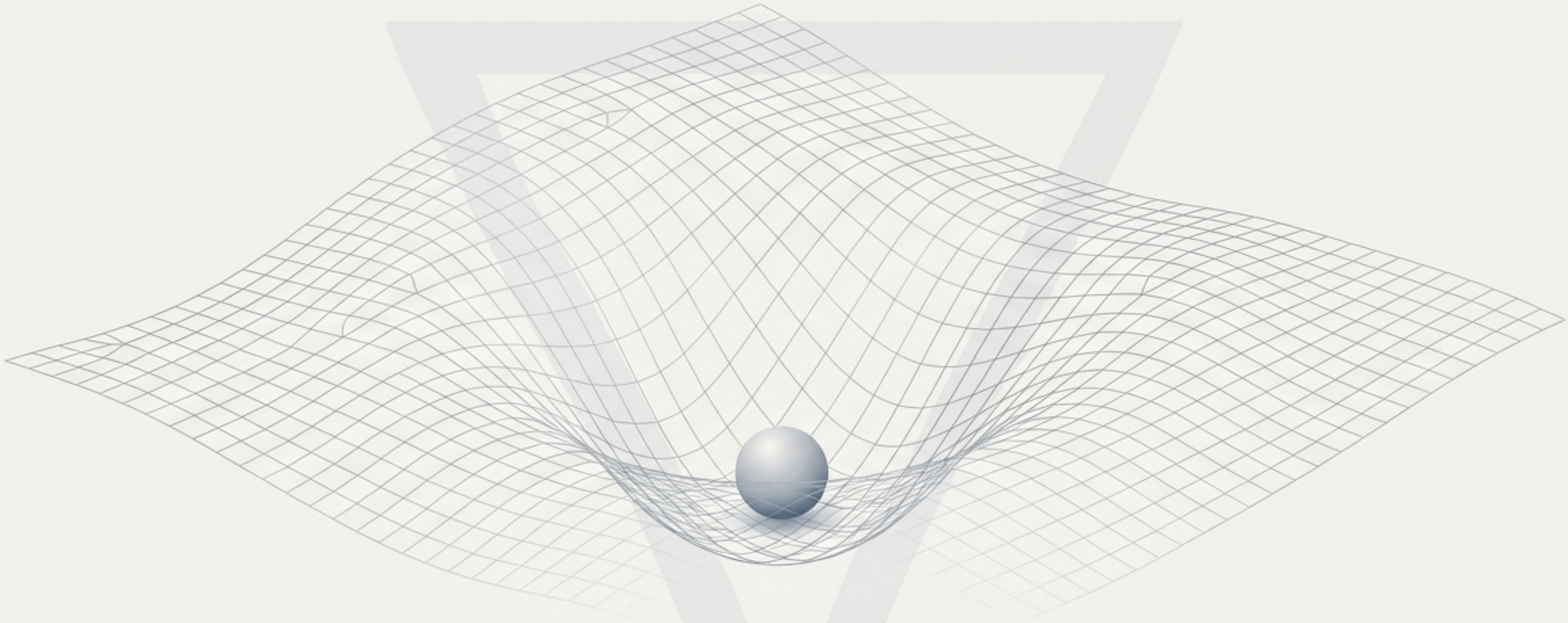
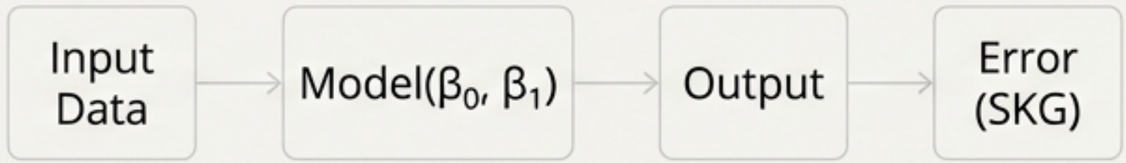


Gradijentni Spust

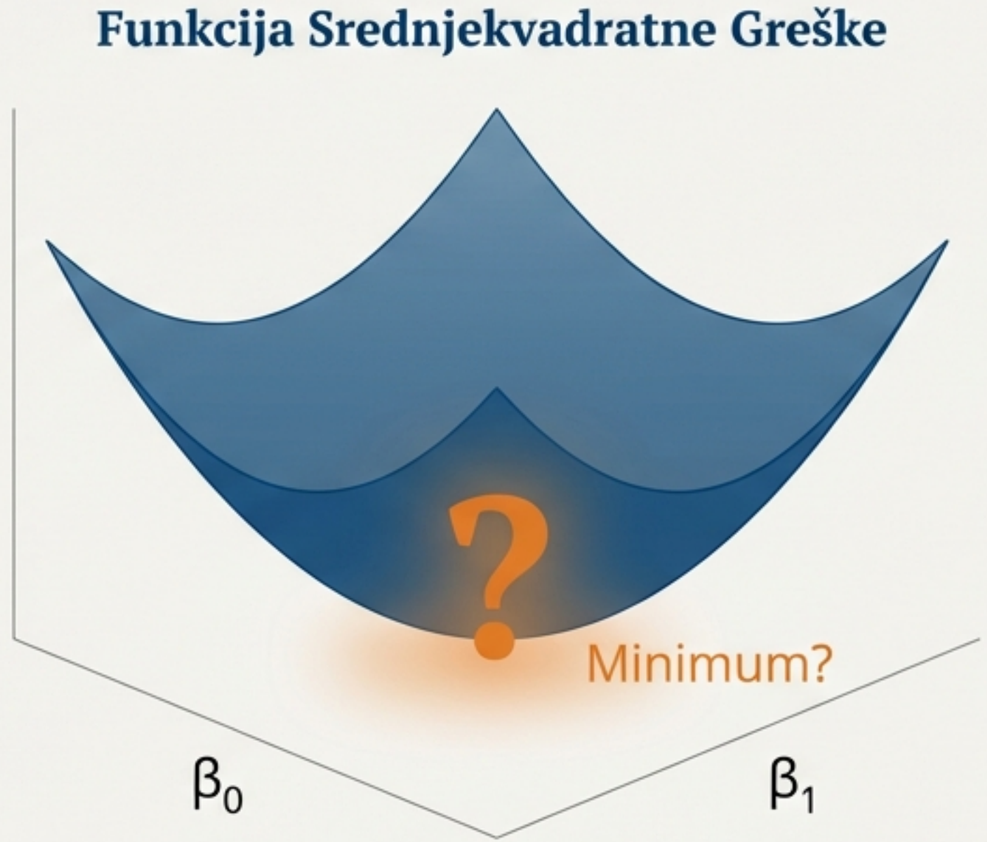
Algoritamsko putovanje do najmanje greške.

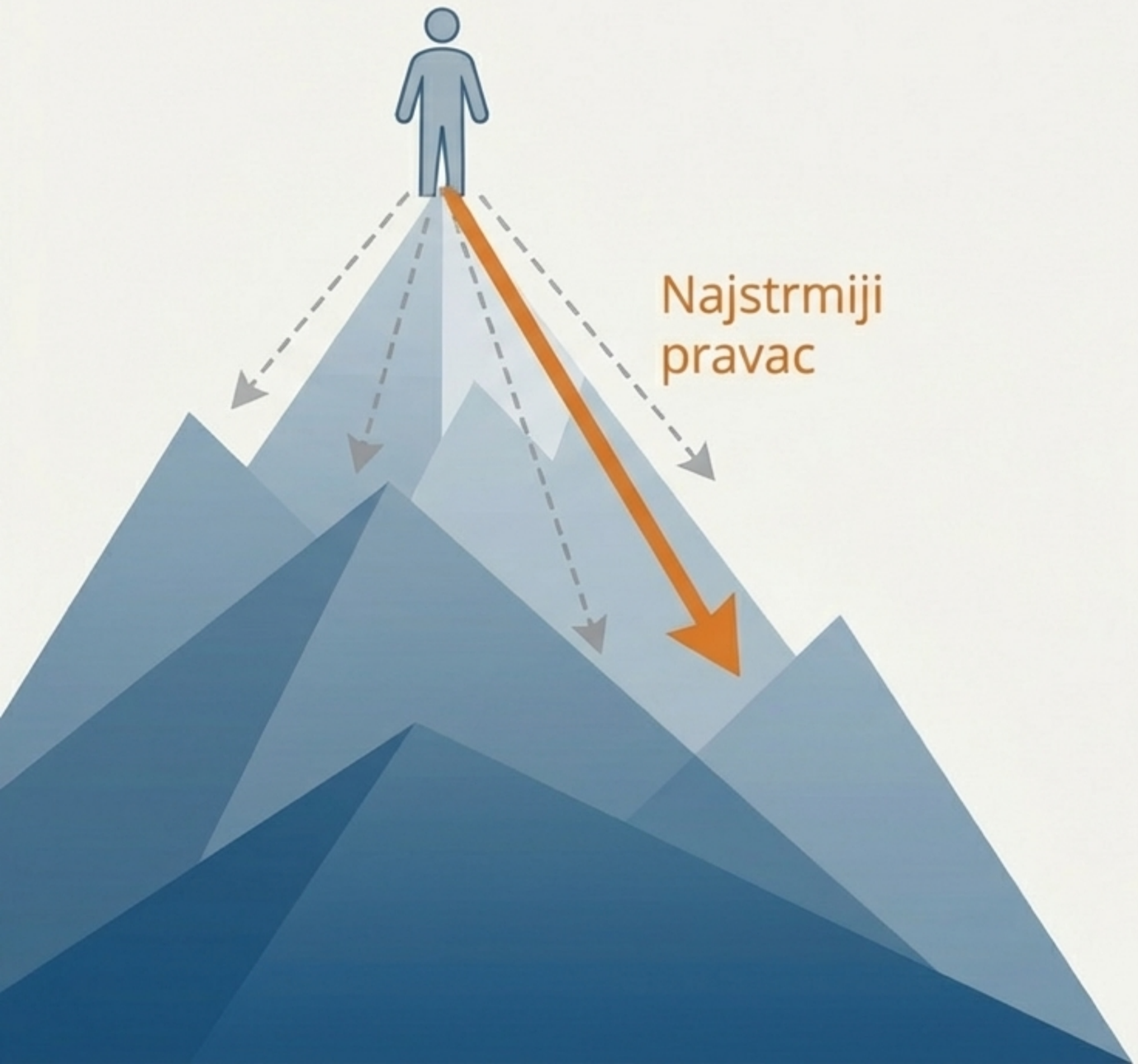
U ovoj lekciji upoznaćemo gradijentni spust, tehniku koja nam pomaže da pronađemo parametre za koje funkcija srednjekvadratne greške ima najmanju vrednost.

Cilj: Pronaći parametre koji daju najmanju grešku.



Svaki model mašinskog učenja ima funkciju greške (npr. Srednjekvadratna greška - SKG). Ona zavisi od parametara modela (β_0, β_1).
Naš zadatak je da pronađemo onu kombinaciju parametara za koju je ova greška najmanja moguća, tj. da pronađemo minimum funkcije.





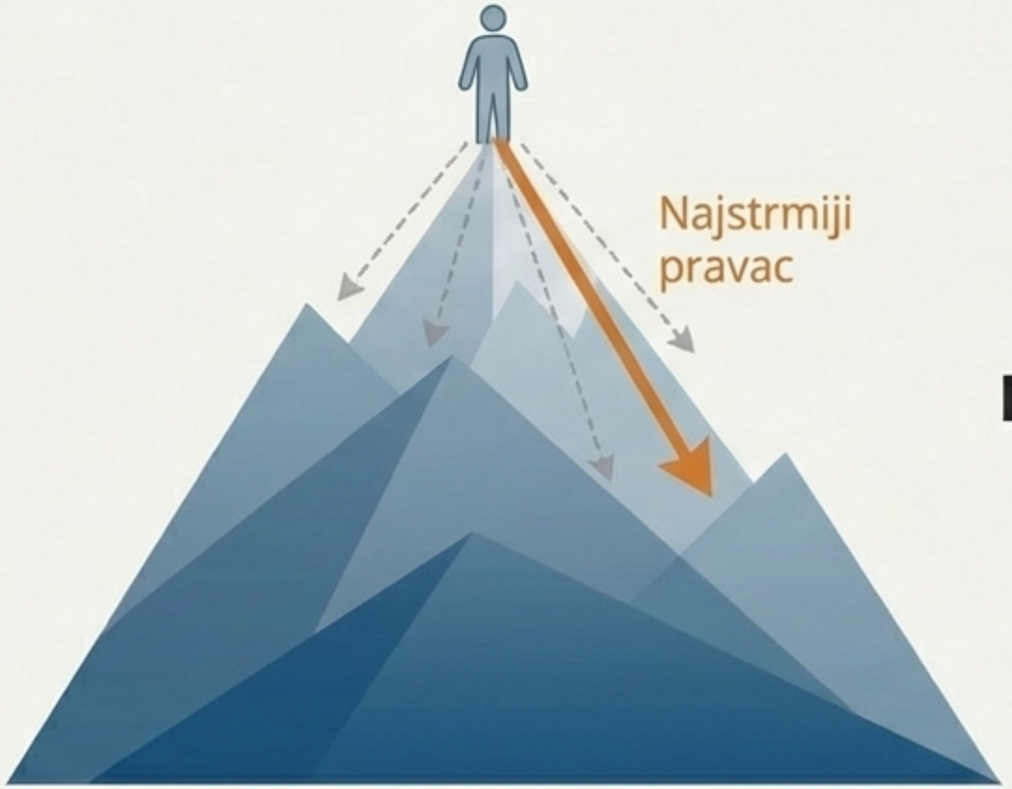
Intuicija: Spuštanje niz planinu najbržim putem.

Zamislite da se nalazite na vrhu planine i želite da se što brže spustite do podnožja. Verovatno biste sledili jednostavan proces:

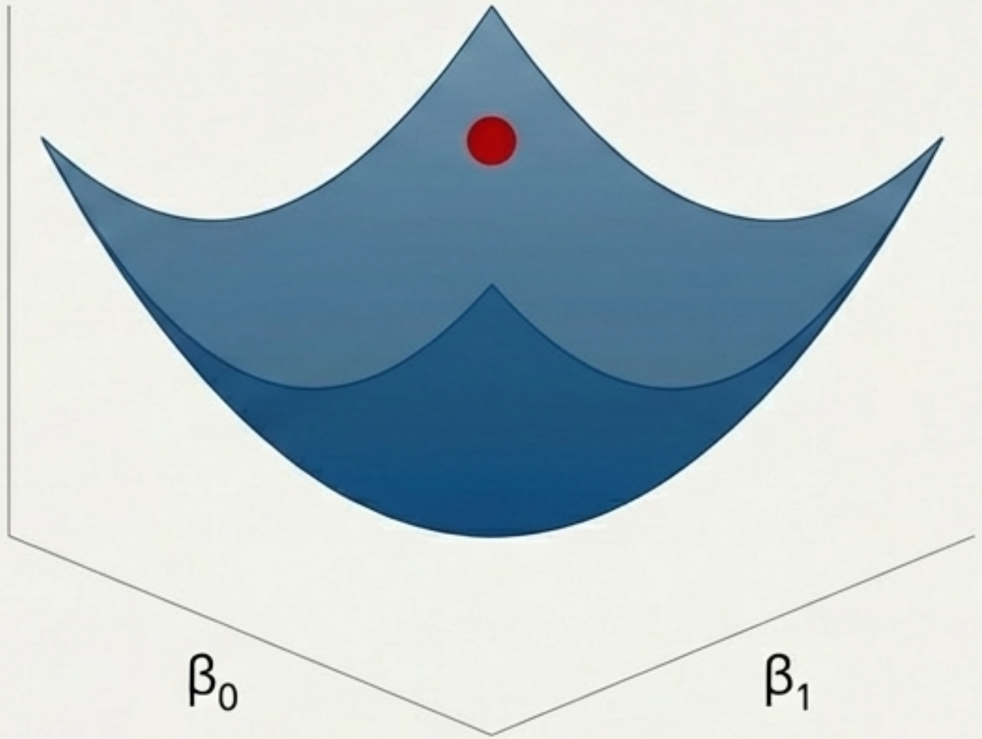
1. **Pogledate oko sebe** i pronađete najstrmiji pravac spusta.
2. **Napravite jedan korak** u tom pravcu.
3. **Ponavljate postupak** dok ne stignete do dna.

Naša planina je funkcija greške.

Baš kao što se spuštamo niz planinu, algoritam se "spušta" niz površ funkcije greške da bi pronašao minimum. Crvena tačka predstavlja nasumični početni izbor parametara. Kako matematički nalazimo "najstrmiji pravac"?



Planina

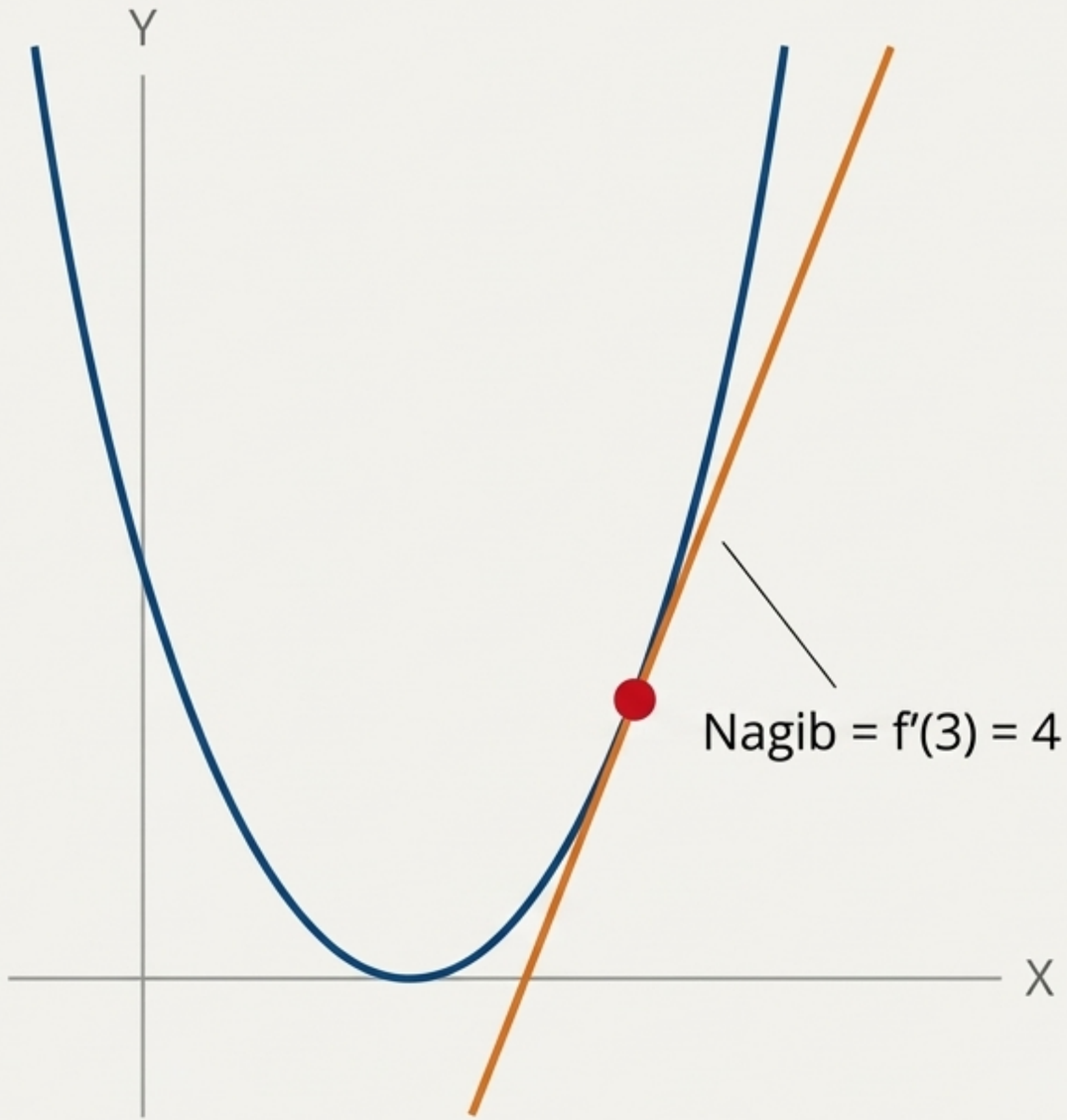


Funkcija Greške

Kompas za spust: Izvod funkcije pokazuje smer.

U jednoj dimenziji, najstrmiji pravac u bilo kojoj tački određen je tangentom na funkciju. Nagib te tangente jednak je vrednosti prvog izvoda funkcije (f'(x)) u toj tački.

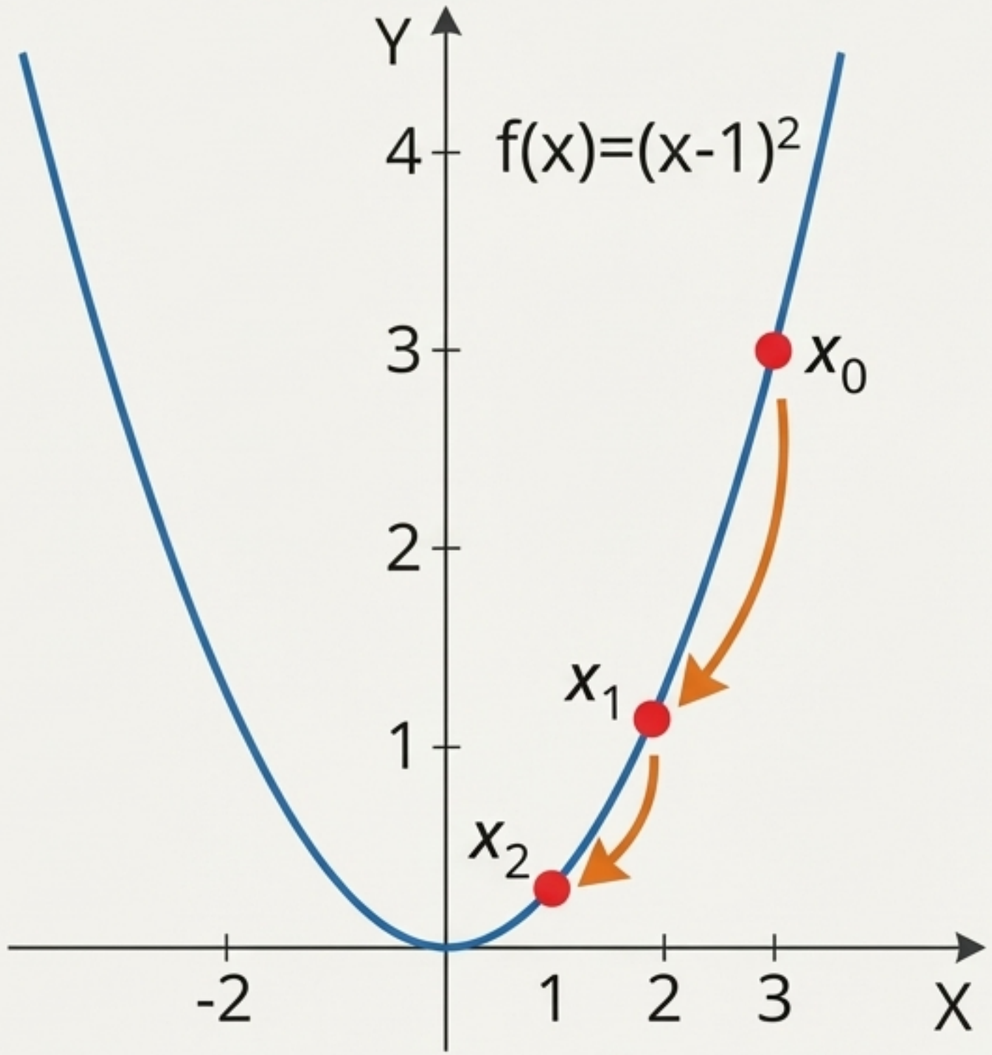
Posmatrajmo funkciju $f(x) = (x-1)^2$. Njen minimum je u x=1.



Korak po korak ka minimumu.

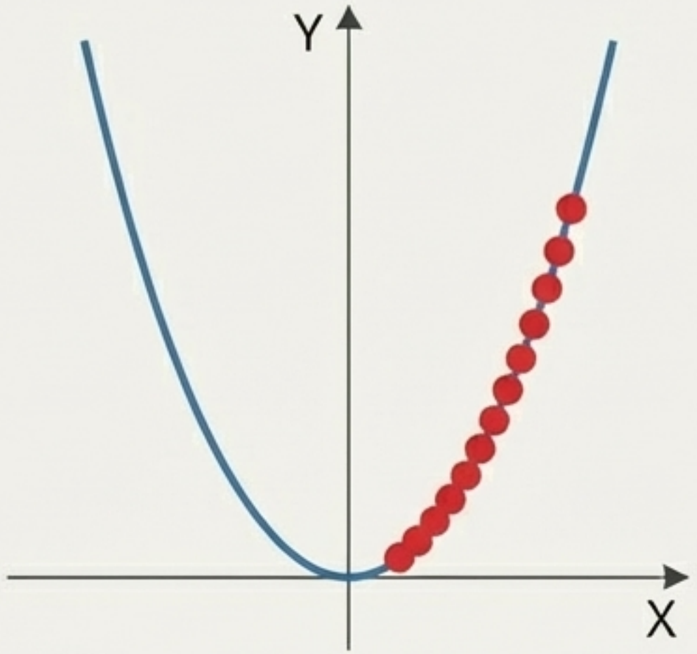
Da bismo se spuštali, krećemo se u pravcu *suprotnom* od izvoda. Svaka nova pozicija (x_{new}) se izračunava na osnovu stare pozicije (x_{old}), izvoda u toj tački i veličine koraka α.

Formula: $x_{new} = x_{old} - \alpha * f'(x_{old})$

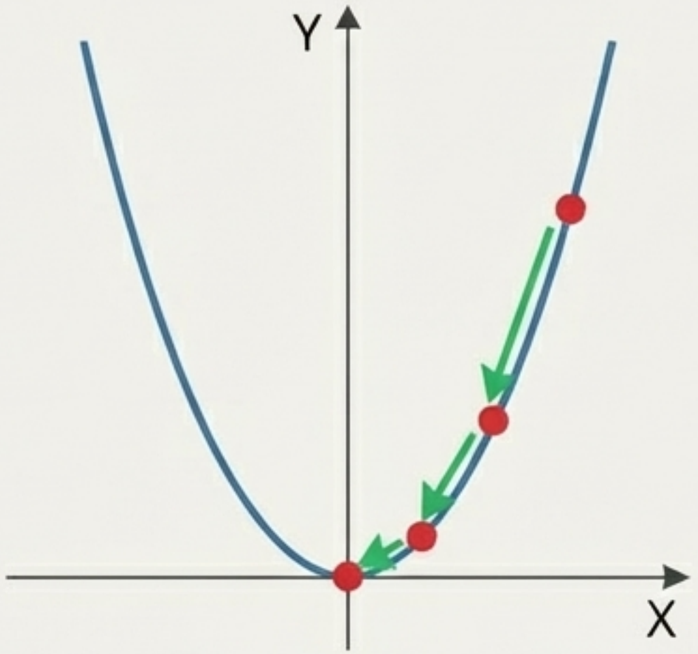


Veličina koraka (α) je ključna.

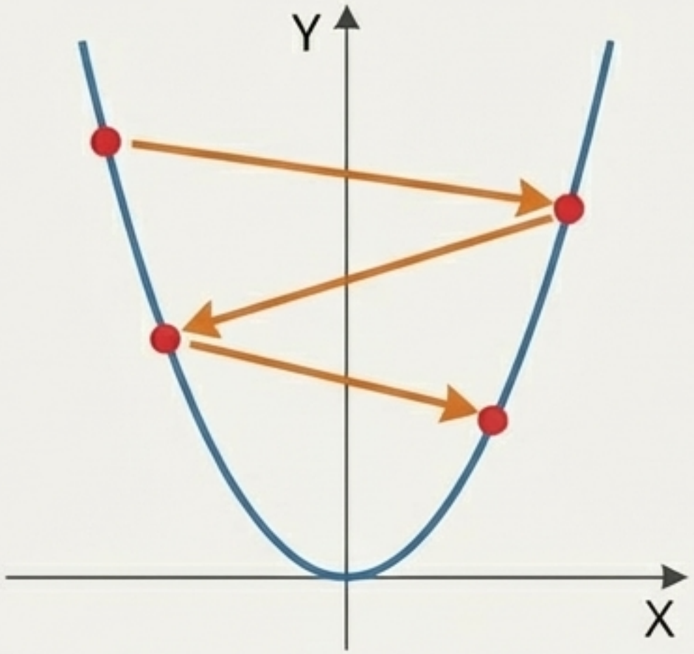
Parametar *α*, poznat kao **korak učenja** (*learning rate*), kontroliše veličinu koraka. Njegov izbor je kritičan za uspeh algoritma.



Spora konvergencija



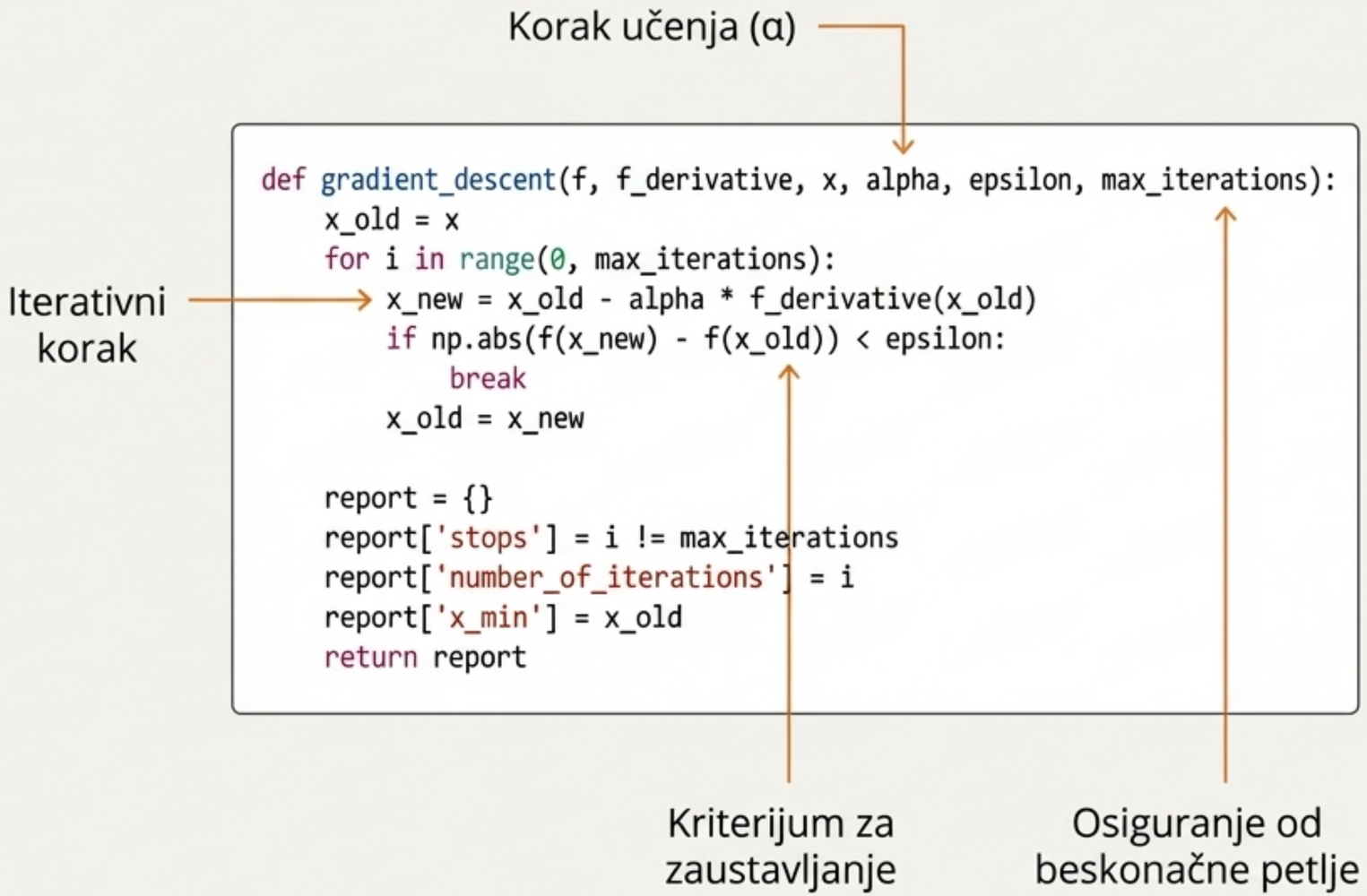
Optimalno



Divergencija / “Cikcak zamka”

Recept za Gradijentni Spust u Python-u.

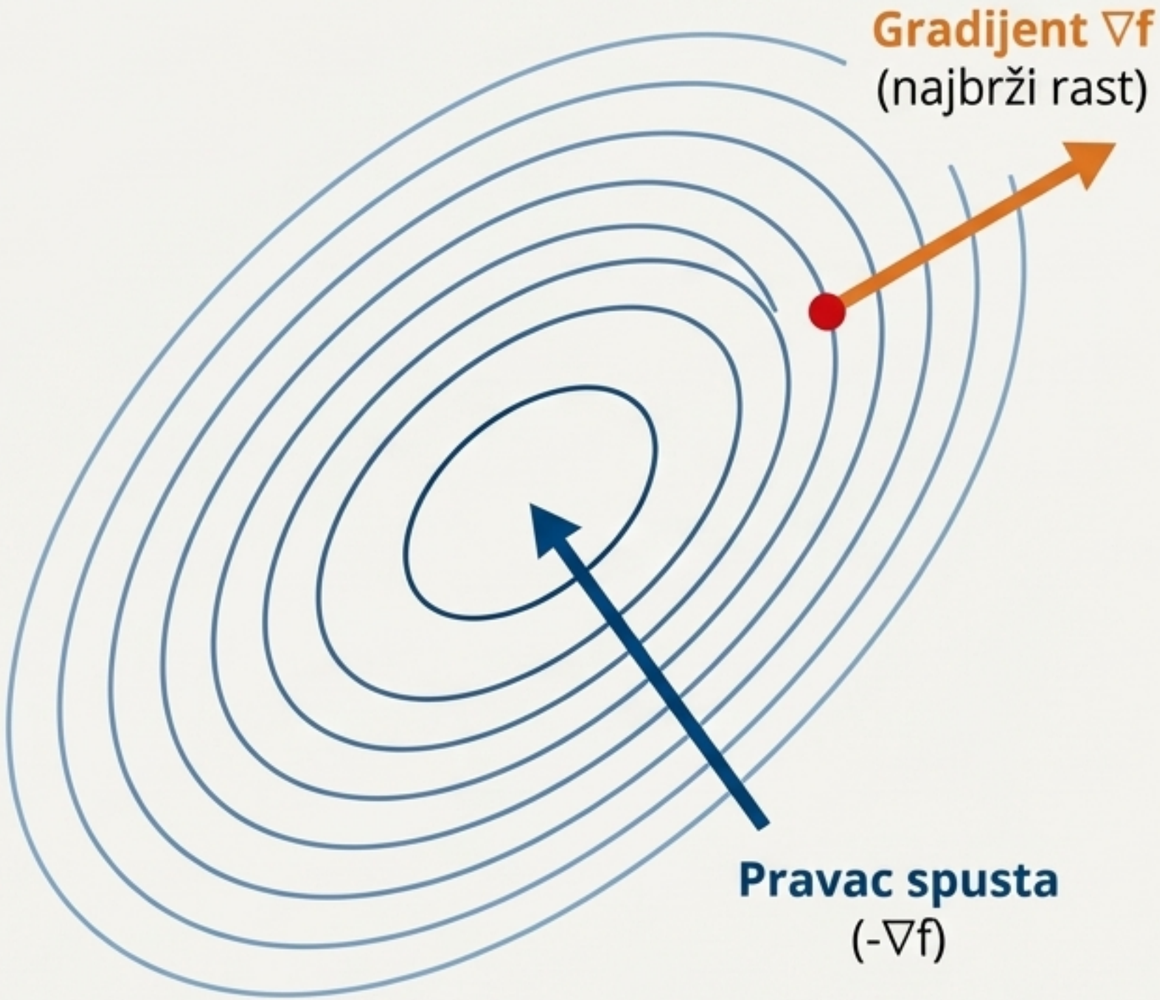
Algoritam se zaustavlja kada je promena vrednosti funkcije zanemarljivo mala (< `epsilon`) ili kada se dostigne maksimalan broj iteracija.



Proširenje mape: Od izvoda do Gradijenta (∇f).

Funkcija greške kod linearne regresije zavisi od dve promenljive: β_0 i β_1. Umesto prostog izvoda, koristimo **Gradijent** (∇f) – vektor parcijalnih izvoda.

Ključna činjenica: Gradijent uvek pokazuje u smeru najbržeg *rasta* funkcije. Zato se mi krećemo u suprotnom smeru ($-\nabla f$).



Primena: Optimizacija Srednjekvadratne Greške

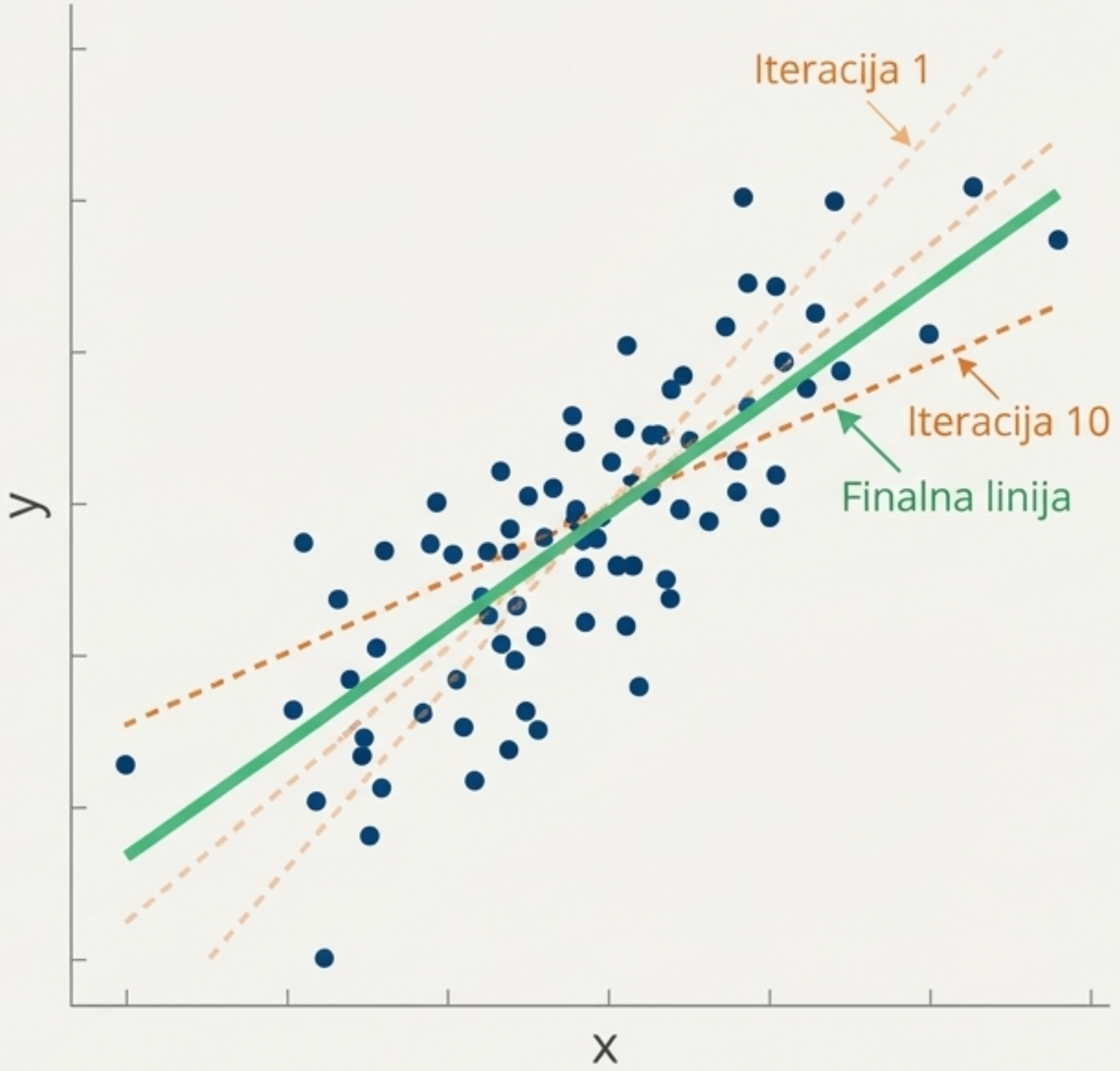
Primenom gradijentnog spusta, iterativno ažuriramo vrednosti β_0 i β_1 koristeći parcijalne izvode funkcije SKG. Ovi izvodi nam govore kako da korigujemo svaki parametar u svakom koraku.

Parcijalni izvod po β_0:

$$\frac{\partial SKG}{\partial \beta_0} = \frac{2}{N} \sum (\beta_0 + \beta_1 x_i - y_i)$$

Parcijalni izvod po β_1:

$$\frac{\partial SKG}{\partial \beta_1} = \frac{2}{N} \sum ((\beta_0 + \beta_1 x_i - y_i) * x_i$$



Rezultat: Pronađeni optimalni parametri.

Za skup podataka o nekretninama koji smo analizirali, algoritam gradijentnog spusta pronalazi vrednosti parametara koje minimizuju grešku.

β_0 (presečak)

2.056

β_1 (nagib)

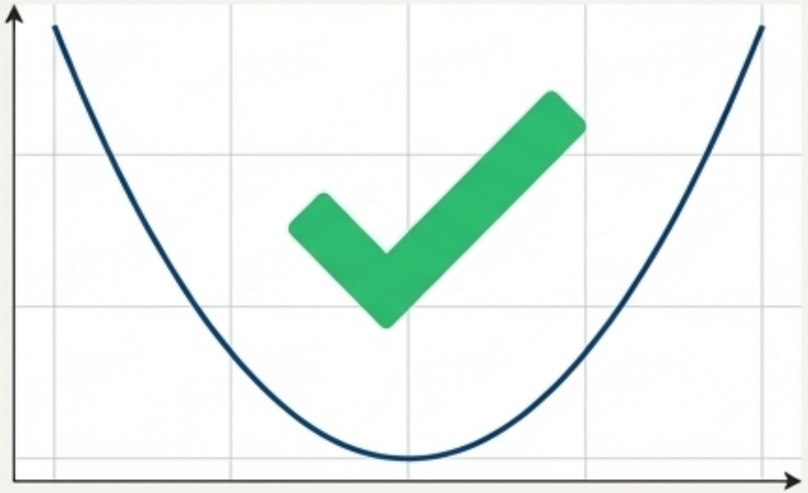
1.198

Važna napomena: Da li uvek stižemo na pravo dno?

Gradijentni spust garantuje pronalazak lokalnog minimuma, što ne mora biti i najniža tačka (globalni minimum).

Rešenje:
Srećom, funkcija srednjekvadratne greške je konveksna. Kod konveksnih funkcija, lokalni minimum je ujedno i globalni minimum. Uvek stižemo do najboljeg mogućeg rešenja.

Konveksna funkcija (SKG)



Šta smo naučili?

Gradijentni spust je **iterativni** algoritam za pronalaženje minimuma funkcije.

Koristi **gradijent (∇f)** da odredi pravac najstrmijeg spusta.

Korak učenja (α) je ključni hiperparametar koji kontroliše brzinu i stabilnost.

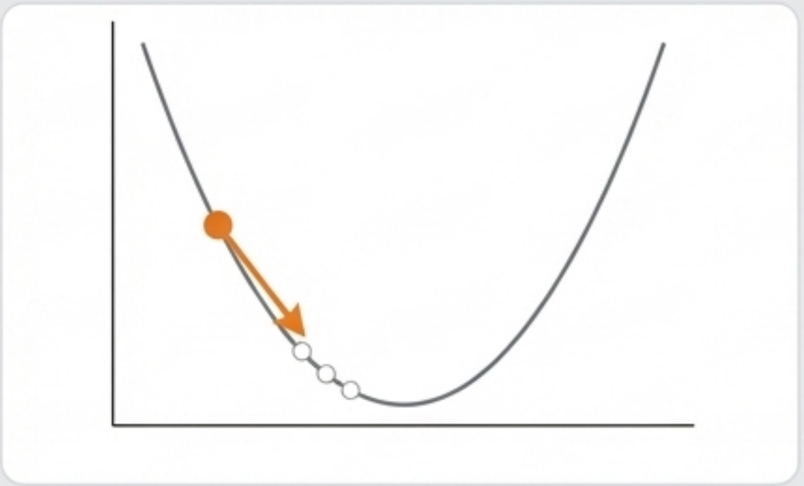
To je jedan od **najvažnijih optimizacionih algoritama** u mašinskom učenju.

Vreme je za praksu!

Najbolji način za učenje je kroz eksperimentisanje. Otvorite prateću Jupyter svesku i sami pokrenite algoritam. Promenite korak učenja i posmatrajte kako to utiče na konvergenciju i rezultat.

Jupyter Sveska

`05-2-gradient_descent.ipynb`



Otvori u Google Colab