

# Jednostruka Linearna Regresija

## Tim:

- Marko Vukotić SW-71/2018
- Miloš Popović SW-24/2018

## Zadatak:

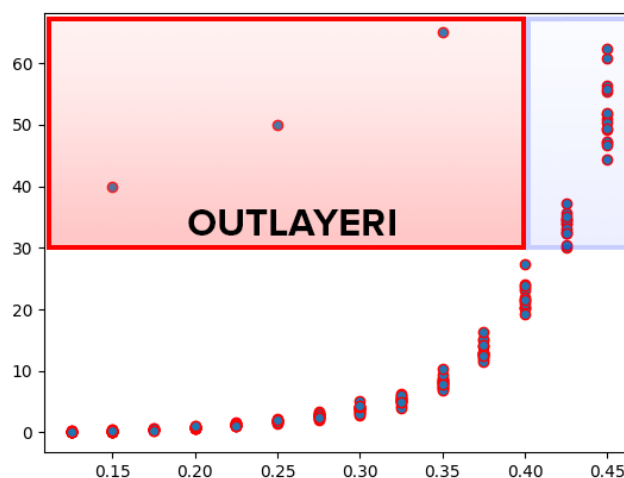
Skup dobijenih podataka sadrži zavisnost otpornosti broda po jedinice težine tokom pomeraja (Y) u odnosu na njegovu veličinu (dimenzije) i brzinu kretanja broda (X). Potrebno je naći regresioni model koji se najbolje uklapa u skup podataka.

## Analiza podataka:

Podaci su dati u formatu:

X	Y
0.3	4.46
0.35	7.16
0.125	0.2
0.2	0.93

Nakon analize podataka uradili smo vizuelizaciju trening skupa. Vizuelizacija je izgledala ovako:



1) Vizuelizacija train.csv

Kako se vidi na datoj fotografiji postoje očigledni *Outlieri* (tačke visokog uticaja). Da bi smo poboljšali ponašanje naseg sistema uklonili smo na trening skupu tačke visokog uticaja. U kodu smo napisali metodu koja obeležava kritične tačke koje se nalaze ~0.36-0.40 na X osi i značajno više na Y osi (~30+) od ostalih tačaka u tom regionu X ose.

## Metoda

Metoda sa kojom smo radili je *Gradijentni Spust* (Gradient Descent). Važne formule:

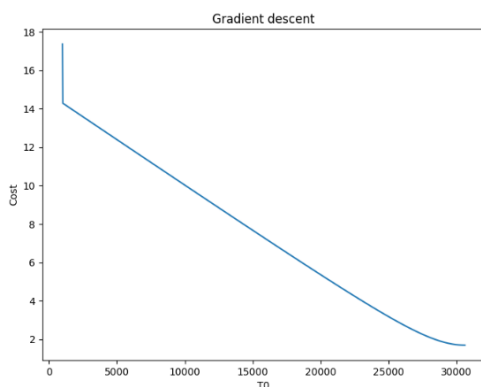
$$J(\theta) = \frac{1}{2N} \sum_{i=1}^N (\theta x^{(i)} - y^{(i)})^2 \quad \bullet \text{ Funkcija Greške (Cost)}$$

*Gradijenti:*

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{N} \sum_{i=1}^N (\theta_1 x^{(i)} + \theta_0 - y^{(i)}) \cdot 1 \quad \bullet \text{ Prvi izvod od } tete0 \text{ (Bias)}$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{N} \sum_{i=1}^N (\theta_1 x^{(i)} + \theta_0 - y^{(i)}) \cdot x^{(i)} \quad \bullet \text{ Prvi izvod od } tete1 \text{ (Weight)}$$

Gradijentni spust je iterativni algoritam koji se koristi da minimizuje funkciju tako što traži optimalne parametre.



Maksimalan broj iteracija: 30000

Learning rate: 0.1

## Funkcije:

Testirali smo algoritam na nekim od funkcija.

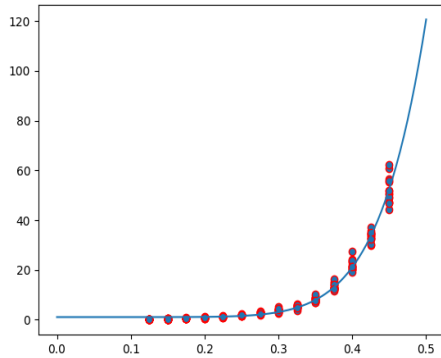
### Funkcija $e^x$

$$Y = t0 \cdot e^x + t1$$

## Linearna x funkcija

$$Y = t0 * x^i + t1, i = \{ 1 - 8 \}$$

„i“ = 6 - 8 su davala obećavajuća rešenja.

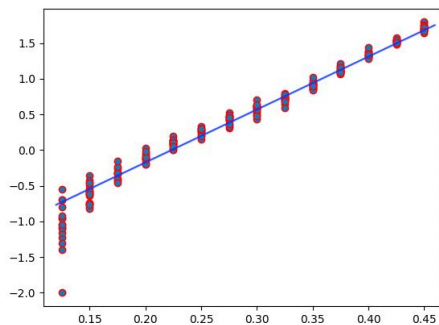


Ovaj model smo koristili za prvi submit, ali imali smo grešku prilikom izbacivanja outlajera (izbačen plavi deo sa slike) tako da smo dobili lošiji rezultat (4.91858265669722).

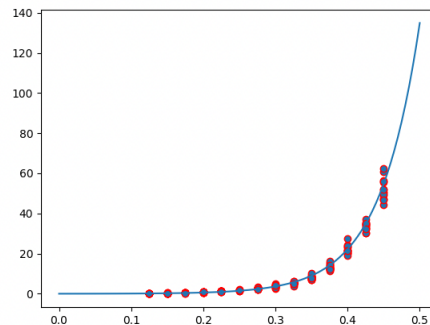
## Logaritamska funkcija (Odabrano rešenje)

$$Y = 8^{(t0 * x + t1)}$$

Drugi pristup je bio da logaritamujemo y parametar ulaznih podataka za obučavanje, i nad takvim podacima fitujemo pravu ( $t0 * x + t1$ ) i kao rezultat vraćamo tu funkciju kao stepen baze logaritma.



Logaritmovan ulaz



Stepenovan ulaz (poništen logaritam)

## Testiranje modela

Ulazne podatke smo podelili na train, validate i test skup 70% 15% 15% i računali smo RMSE nad test skupom. Validate smo koristili kao odabir najbolje iteracije gradient descent-a kako bi izbegli overfitting, ali nismo primetili neko poboljšanje tako da nismo iskoristili to za konačan model.