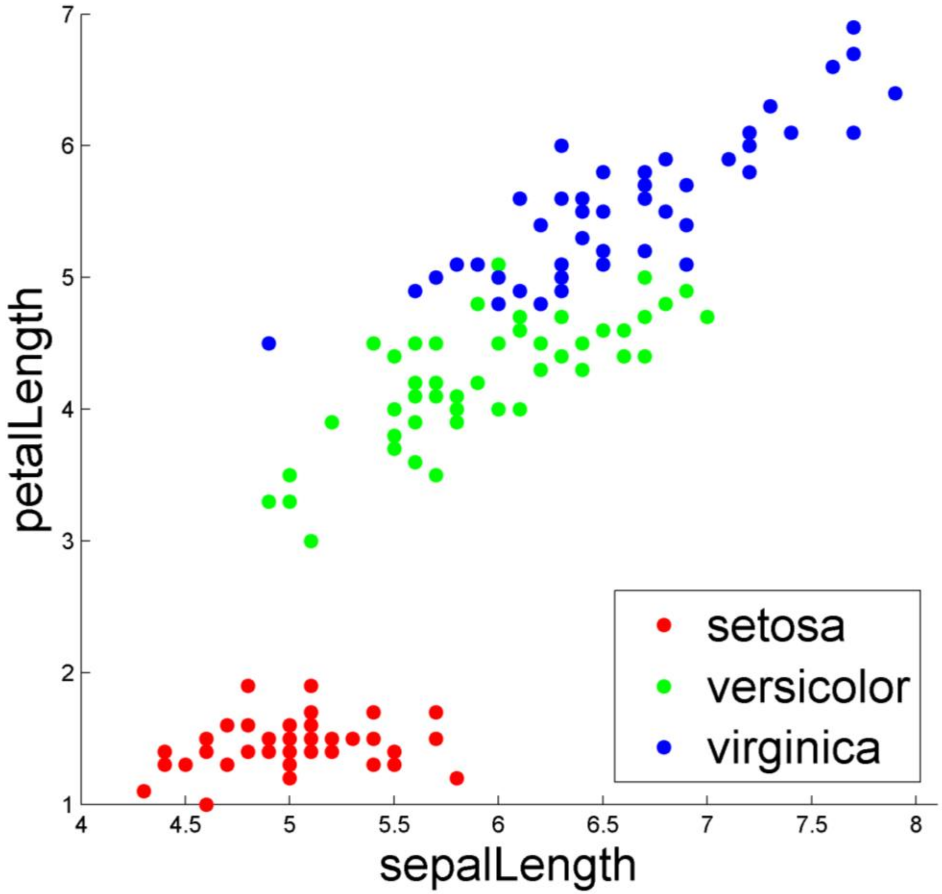


Praktična razmatranja

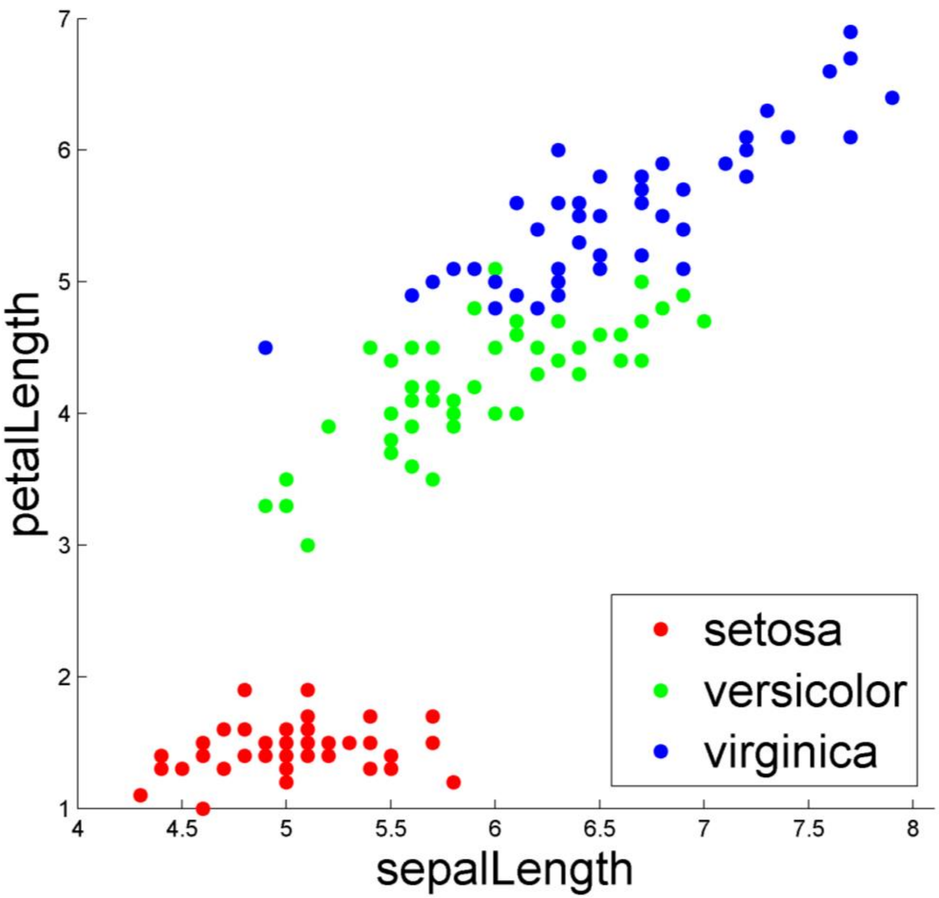
Višekategorijska klasifikacija

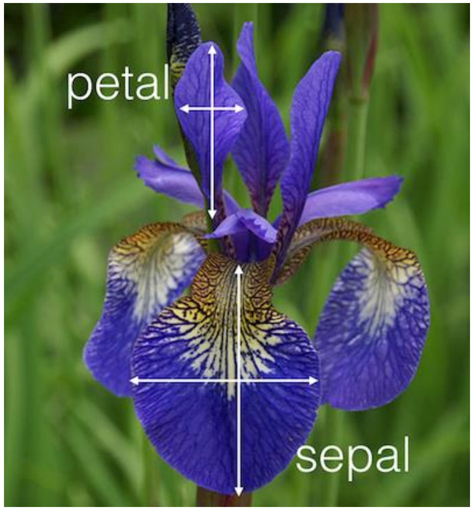
Višekategorijska klasifikacija

Iris setosa

Iris versicolor

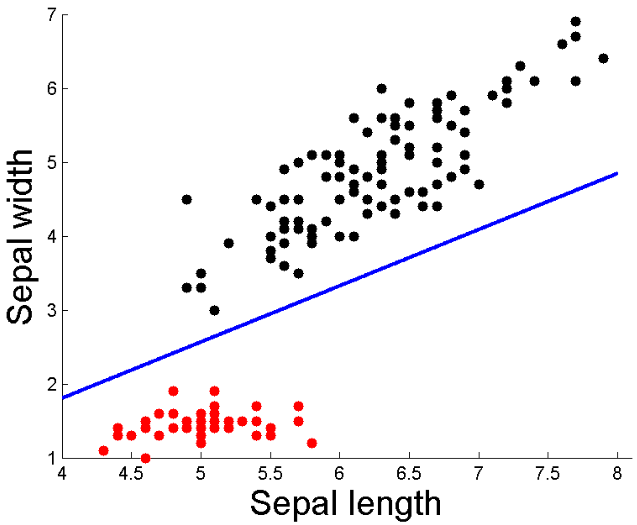
Iris virginica



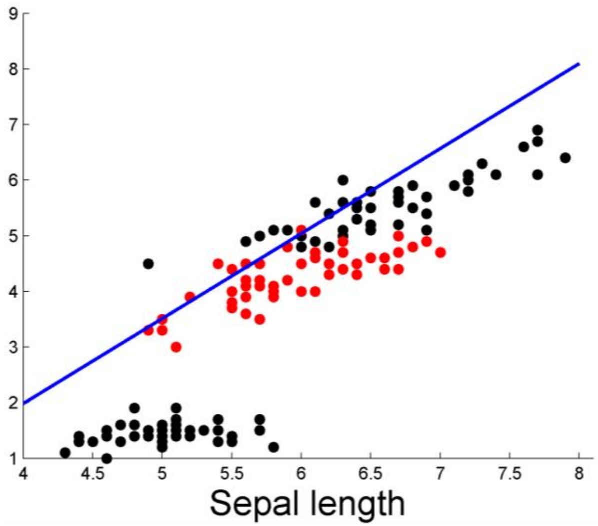


Petal length	Sepal length	species
5.1	1.4	setosa
4.9	1.4	setosa
7	4.7	versicolor
6.3	6	virginica
…	…	…

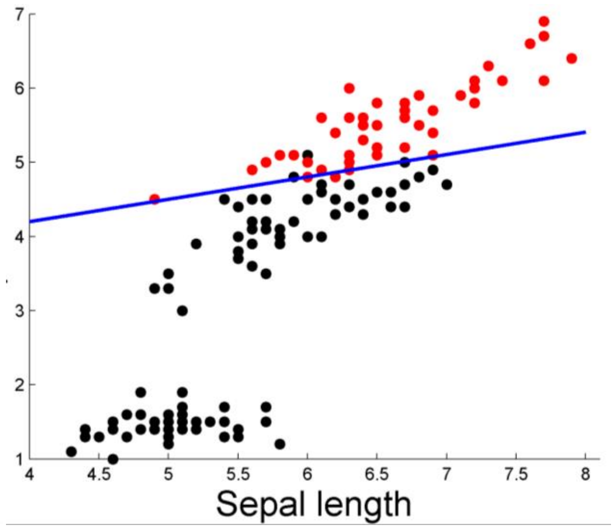
1-vs-all model



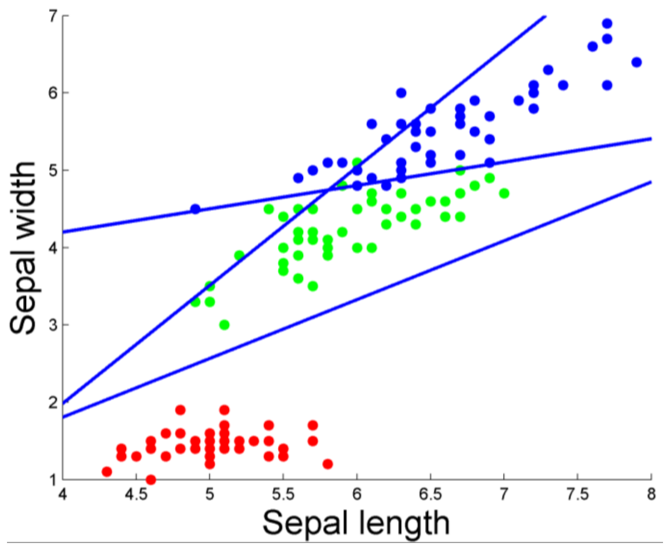
$y = 1$: *setosa*
$y = 0$: *versicolor, virginica*



$y = 1$: *versicolor*
$y = 0$: *setosa, virginica*



$y = 1$: *virginica*
$y = 0$: *setosa, versicolor*



Predvideti klasu sa najvećom verovatnoćom:

$P_{max} = 0;\ \hat{y} = 0$

Za svaku klasu $c \in \{1, \dots, C\}$:

ako je $P_c\left(y^{(i)} = 1 | x^{(i)}\right) > P_{max}$:

$\hat{y} = c$

1-vs-1 model

- Za svaki par klasa c_k i $c_{k'}$ napraviti problem binarne klasifikacije
  - Trening podaci sa labelom c_k će biti ‘pozitivni’ ($y = 1$)
  - Trening podaci sa labelom $c_{k'}$ će biti ‘negativni’ ($y = 0$)
  - Zanemariti sve ostale trening podatke

- Trenirati $C(C-1)/2$ binarna klasifikatora (C – broj klasa)
- Predikcija: kombinovati izlaz rezultujuća $C(C-1)/2$ klasifikatora

Kada koristiti koji pristup?

1-vs-all

C klasifikatora

Računarski manje zahtevan za veliko C

Izuzetno nebalansirani skupovi podataka za veliko C (značajno više negativnih primera)

1-vs-1

$\frac{C(C-1)}{2}$ klasifikatora

Za treniranje koristimo manji podskup podataka (brže ako imamo veliki trening skup)

Manje problematičan sa aspekta neizbalansiranosti

Softmax Regression – model

- Ili: *Multinomial Logistic, Maximum Entropy Classifier* ili *Multi-class Logistic Regression*

- Generalizacija logističke regresije za višekategorijsku klasifikaciju

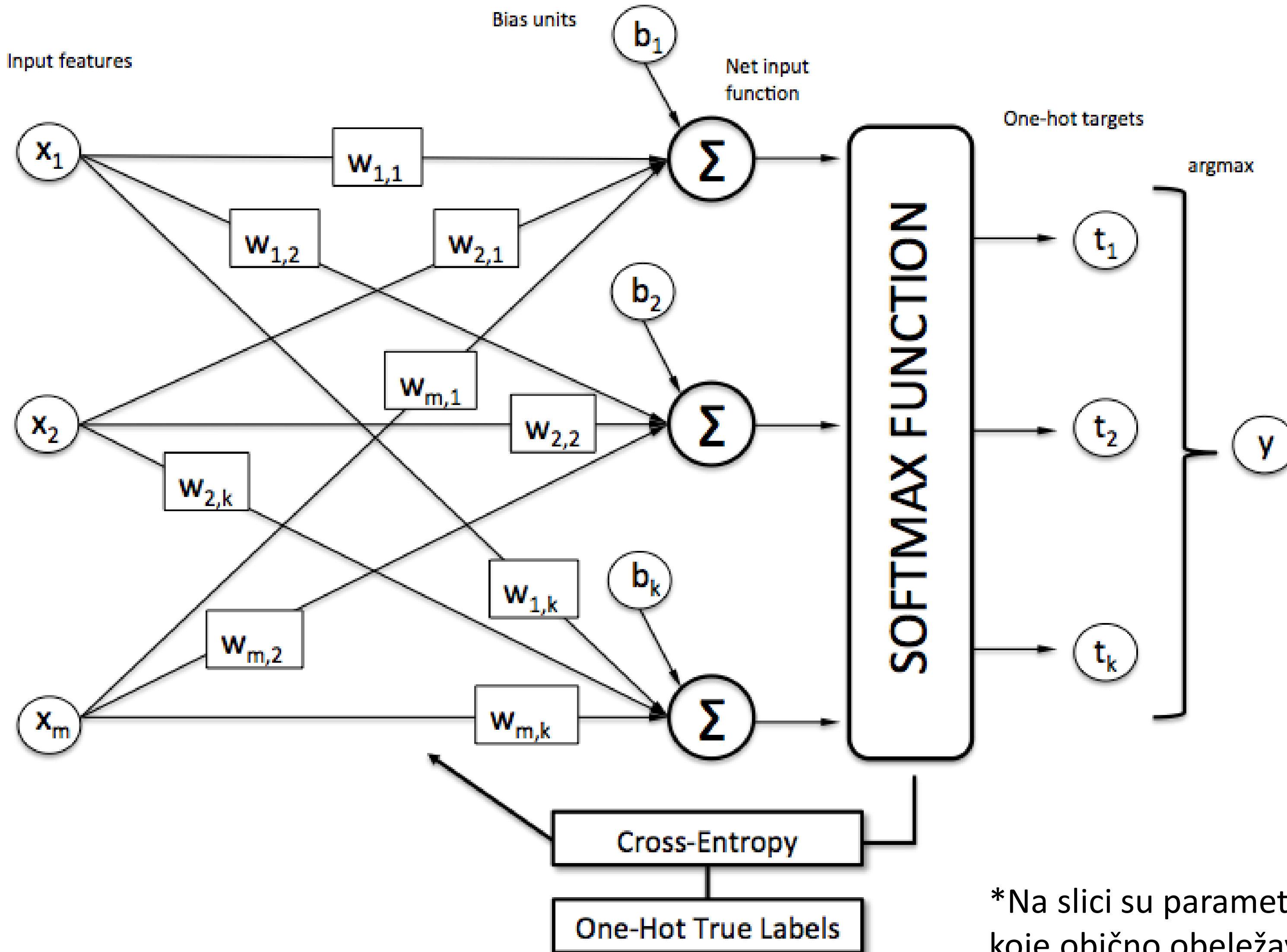
Softmax Regression – model

- Softmax funkcija računa verovatnoću da trening primer $x^{(i)}$ pripada klasi c_j
- Treba da izračunamo verovatnoću $P\left(y = c_j | s^{(i)}\right)$ za svaku od klasa $\{c_k, k = 1, \ldots, C\}$
- Dakle, izlaz $h_\theta(x)$ će biti *vektor* čiji su članovi verovatnoća „uspeha" za svaku od klasa:

$$h_\theta(x) = \begin{bmatrix} P(y = c_1 | x) \\ P(y = c_2 | x) \\ \cdots \\ P(y = c_C | x) \end{bmatrix}$$

- Naša predikcija: klasa sa navećom verovatnoćom „uspeha"
- Ali, mora da važi $\sum_{k=1}^{C} P(y = c_k | x) = 1$ - za ovo se brine *softmax* funkcija

Softmax Regression – model



*Na slici su parametri modela koje obično obeležavamo θ obeleženi sa w

Softmax Regression – model

- Parametri modela:

$$\theta = \begin{bmatrix} | & | & | & | \\ \theta^{(1)} & \theta^{(2)} & \dots & \theta^{(C)} \\ | & | & | & | \end{bmatrix}_{D \times C}$$

- gde je D broj obeležja, a C broj klasa

Softmax Regression – model

- Sigmoidnu funkciju (koju smo koristili u binarnoj logističkoj regresiji):

$$P\left(y = 1|s^{(i)}\right) = \frac{1}{1 + e^{-s(i)}}$$

- ćemo zameniti sa tzv. *softmax* funkcijom:

$$P\left(y = c_j|s^{(i)}\right) = f\left(s^{(i)}\right) = \frac{e^{s(i)}}{\sum_{(k=1)}^{C} e^{s(k)}}$$

- gde je $s = \theta^T x$ („score“), a C broj klasa

Primer

- Recimo da imamo skup podataka (4 primera i 3 moguće klase):

$$\left\{\left(x^{(1)}, c_1\right), \left(x^{(2)}, c_2\right), \left(x^{(3)}, c_3\right), \left(x^{(4)}, c_3\right)\right\}$$

- Izlaz y ćemo predstaviti putem *1-hot-encoding*:

$$y^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, y^{(2)} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, y^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, y^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Ulaz x ćemo predstaviti matricom čiji redovi predstavljaju trening primere, a kolone obeležja (recimo da imamo 2 obeležja):

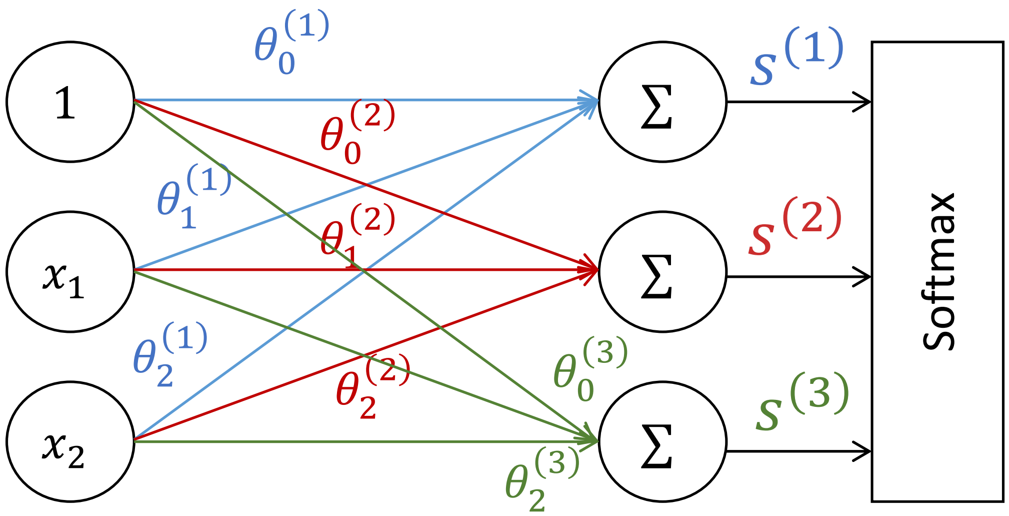
$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} \\ 1 & x_1^{(4)} & x_2^{(4)} \end{bmatrix}_{4\times 3}$$

Primer

- Matrica koeficijenata θ će biti dimenzije 3×3:

$$\theta = \begin{bmatrix} \boxed{\begin{matrix} \theta_0^{(1)} \\ \theta_1^{(1)} \\ \theta_2^{(1)} \end{matrix}} & \boxed{\begin{matrix} \theta_0^{(2)} \\ \theta_1^{(2)} \\ \theta_2^{(2)} \end{matrix}} & \boxed{\begin{matrix} \theta_0^{(3)} \\ \theta_1^{(3)} \\ \theta_2^{(3)} \end{matrix}} \end{bmatrix}$$

- Da bismo dobili „score“ s množimo matrice $x_{4\times 3}$ i $\theta_{3\times 3}$ kao izlaz dobijamo matricu $S = X\theta$ dimenzije 4×3



Primer

- Matricu $S_{4\times3}$ propuštamo kroz softmax funkciju

$$sum \approx 1$$

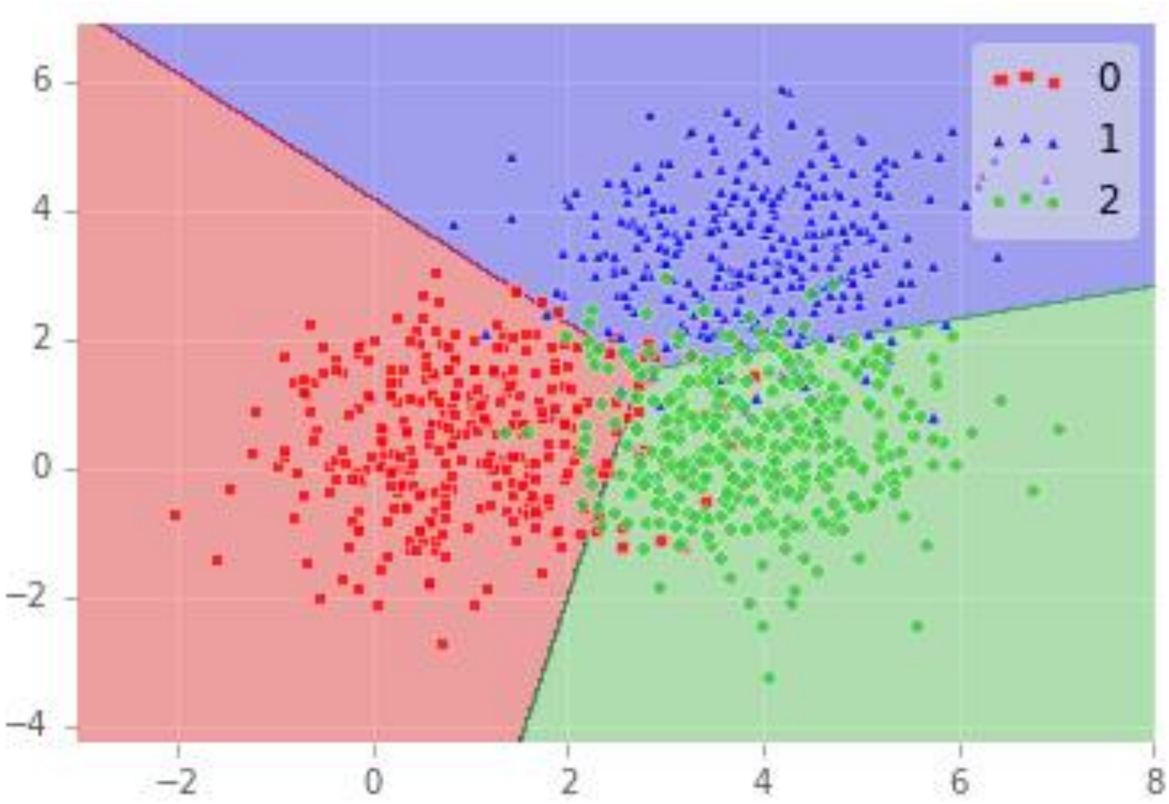
$$S = \begin{bmatrix} 0.07 & 0.22 & 0.28 \\ 0.35 & 0.78 & 1.12 \\ -0.33 & -0.58 & -0.92 \\ 0.39 & -0.7 & -1.1 \end{bmatrix} \rightarrow \begin{bmatrix} 0.294 & 0.342 & \mathbf{0.363} \\ 0.213 & 0.327 & \mathbf{0.460} \\ \mathbf{0.429} & 0.334 & 0.238 \\ \mathbf{0.449} & 0.330 & 0.221 \end{bmatrix}$$

- Prvi i drugi primer bismo stavili u klasu c_3, a treći i četvrti u c_1
- Trening skup: $\{(x^{(1)}, c_1), (x^{(2)}, c_2), (x^{(3)}, c_3), (x^{(4)}, c_3)\}$
- Naše predikcije nisu dobre, tako da treba rafinirati θ

Pun primer možete naći na https://rasbt.github.io/mlxtend/user_guide/classifier/SoftmaxRegression/

Granica odluke

- Granica odluke: granica između bilo koje dve klase je linearna (prava linija)



Napomena: numerička stabilnost

- Kod računanja vrednosti *softmax* funkcije, vrednosti e^{s_j} i $\sum_k e^{s_k}$ mogu biti veoma velike zbog eksponenta (potencialni *overflow*)
- Zbog toga koristimo sledeći trik: ako pomnožimo imenilac i brojilac konstantom C, dobijamo ekvivalentan izraz:

$$\frac{e^{s_j}}{\sum_k e^{s_k}} = \frac{Ce^{s_j}}{C\sum_k e^{s_k}} = \frac{e^{s_j + \log C}}{\sum_k e^{s_k + \log C}}$$

- Čest izbor $\log(C) = -\max_j s_j \rightarrow$ „pomerili“ smo vrednosti vektora s tako da je najveća vrednost 0