Systemy agentowe

Klasyfikacja

Jędrzej Potoniec

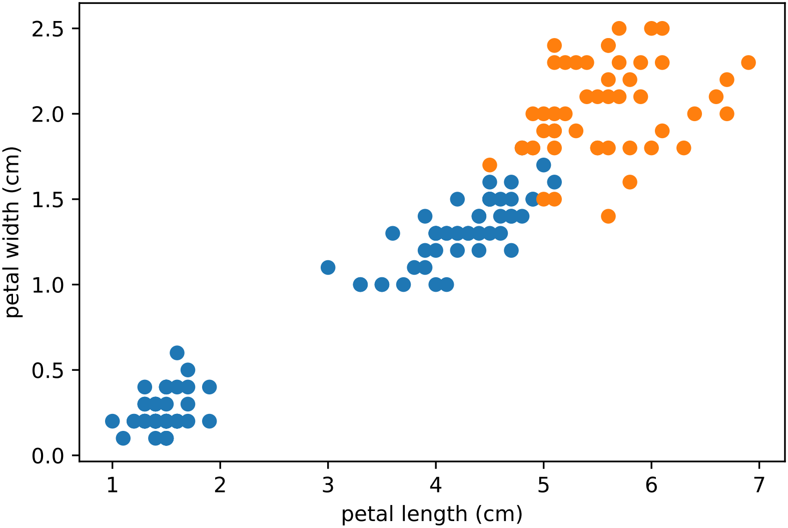
Klasyfikacja

Zadanie klasyfikacji binarnej

Dla danego wektora cech $\mathbf{x}$ opisującego obiekt przewidzieć czy obiekt należy do klasy *pozytywnej* $y = 1$ czy *negatywnej* $y = 0$ (lub $y = -1$).

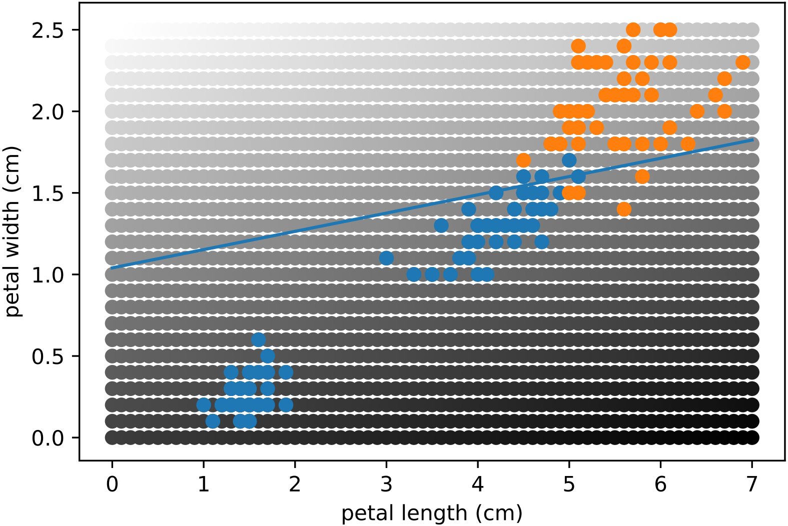
Irysy

Pomarańczowy: Iris Virginica, niebieski: pozostałe

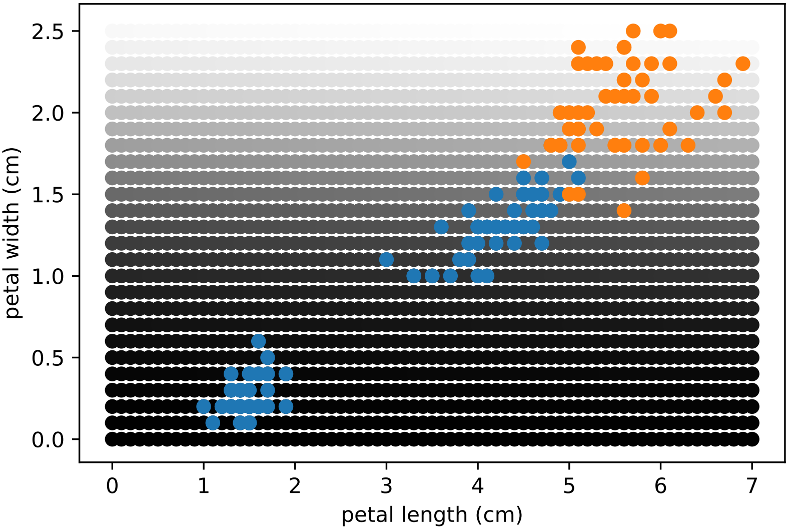


Przewidywanie prawdopodobieństwa – regresja liniowa

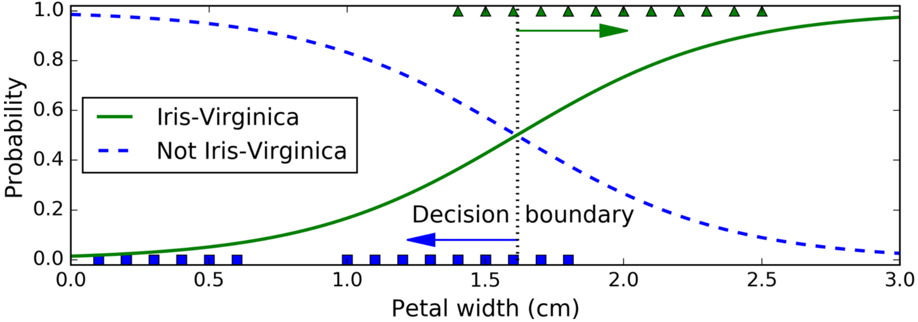
Punkty w tle: prawd. Iris Virginica (jaśniej=wyższe)



Przewidywanie prawdopodobieństwa – regresja logistyczna

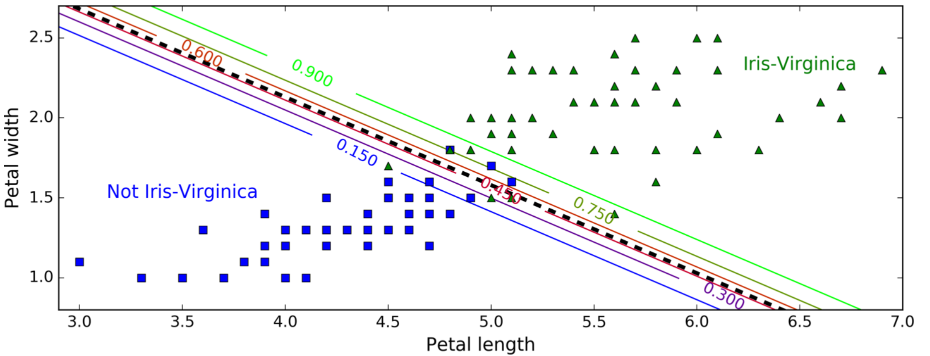


Granica decyzyjna (ang. decision boundary)



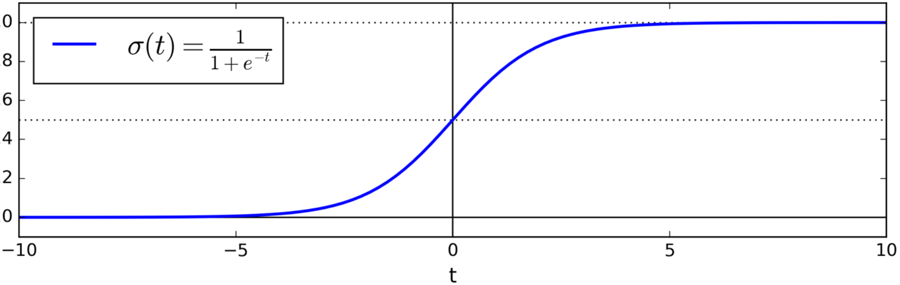
A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow* 2017

Granica decyzyjna (ang. decision boundary)



A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow* 2017

Funkcja logistyczna



A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow* 2017

Pochodna funkcji logisytcznej

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Zadanie

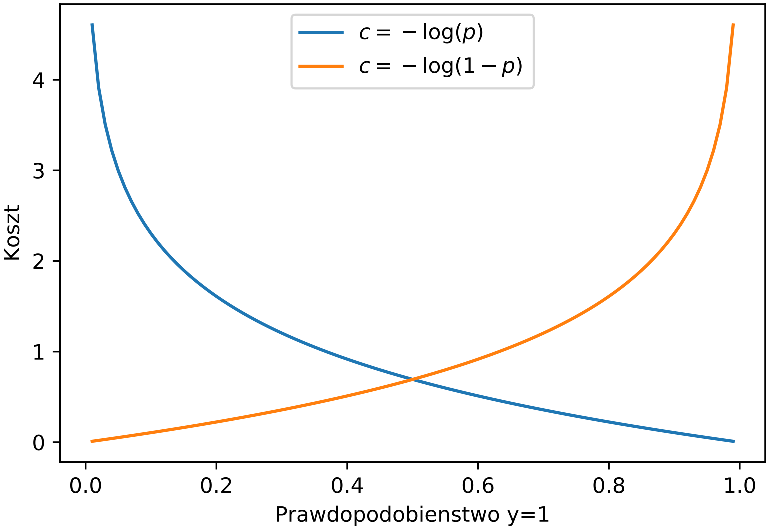
Wiadomo, że $\sigma(t_0) = .1$. Czy da się na tej podstawie prosto obliczyć wartość pochodnej $\sigma'(t)$ w punkcie t_0?

Regresja logistyczna

$$\hat{p} = \sigma(\mathbf{xw})$$

$$\hat{y} = \begin{cases} 1 & \hat{p} \geqslant 0{,}5 \\ 0 & \hat{p} < 0{,}5 \end{cases}$$

Funkcja kosztu dla pojedynczego przykładu



Funkcja kosztu dla pojedynczego przykładu

Zadanie

Zapisać poniższą funkcję jako pojedyncze wyrażenie (używając dodawania, mnożenia itp.):

$$c(y, p) = \begin{cases} -\log(p) & y = 1 \\ -\log(1 - p) & y = 0 \end{cases}$$

Funkcja kosztu dla całego problemu

$$\mathbf{p} = \sigma(\mathbf{X}\mathbf{w})$$

$$J(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} c\left(y_i, p_i\right) =$$

$$-\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log p_i + (1 - y_i) \log(1 - p_i)\right]$$

Funkcja kosztu dla całego problemu

$$\mathbf{p} = \sigma(\mathbf{X}\mathbf{w})$$

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} c\left(y_i, p_i\right) =$$

$$-\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log p_i + \left(1 - y_i\right)\log(1 - p_i)\right]$$

$$\frac{\partial J}{\partial w_i} = \frac{1}{n}\sum_{i=1}^{n}\left(\sigma(\mathbf{X_i}\mathbf{w}) - y_i\right)X_{i,j}$$

Softmax regression/Multinomial logistic regression

Zadanie klasyfikacji

Dla danego wektora cech x opisującego obiekt przewidzieć, do której z K predefiniowanych klas należy obiekt: $y \in \{1, 2, \dots, K\}$

$$\mathbf{W} \text{ macierz wag typu } p \times k$$

$$\mathbf{P} \text{ macierz prawdopodobieństw typu } n \times k$$

$$\hat{\mathbf{P}} = \sigma(\mathbf{XW}) = \left[\frac{\sigma(\mathbf{X_{i,\cdot}W_{\cdot,k}})}{\sum_{l=1}^{K} \sigma(\mathbf{X_{i,\cdot}W_{\cdot,l}})} \right]_{i,k}$$

$$\hat{y}_i = \arg\max_k \hat{P_{i,k}} = \arg\max_k \mathbf{X_{i,\cdot}W_{\cdot,k}}$$

Uczenie regresji softmax

$$Y_{i,k} = \begin{cases} 1 & y_i = k \\ 0 & \text{w przeciwnym przypadku} \end{cases}$$

$$J(\mathbf{W}) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} Y_{i,k} \log \hat{P_{i,k}}$$

$$\nabla_{\mathbf{W}_{\cdot,\mathbf{k}}} J(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{P_{i,k}} - Y_{i,k} \right) \mathbf{X_i}$$