

Mechatronic Engineering program

**Basics of AI and Deep Learning:  
2: Regression (and classification)**

Ziemowit Dworakowski  
*AGH University of Krakow*

1

---

---

---

---


---

---




---

---

**Example 1: Used car retail (2023)** *SD*

 We have a 7-year old Opel Astra, with 80 000 km mileage.  
We want to sell it quickly with as high price as possible.

**Initial price**

 **Too low**  **Too high** 

**We will not earn much**  **Car won't sell quickly**

2

---

---

---

---

---

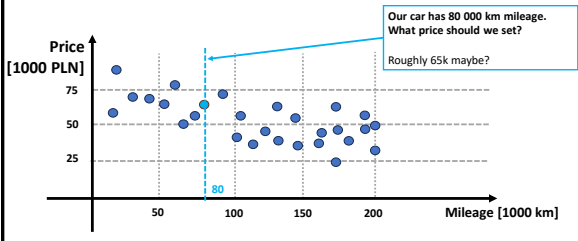
---

---

---

**Example 1: Used car retail (2023)** *SD*

Our car has 80 000 km mileage.  
What price should we set?  
Roughly 65k maybe?



The scatter plot shows a negative correlation between mileage and price. The y-axis is labeled 'Price [1000 PLN]' with values 25, 50, 75. The x-axis is labeled 'Mileage [1000 km]' with values 50, 100, 150, 200. A vertical dashed blue line is drawn at 80 on the x-axis. A callout box points to this line with the text 'Our car has 80 000 km mileage. What price should we set? Roughly 65k maybe?'.

3

---

---

---

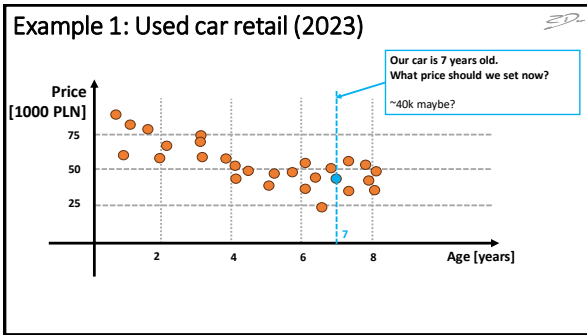
---

---

---

---

---




---

---

---

---

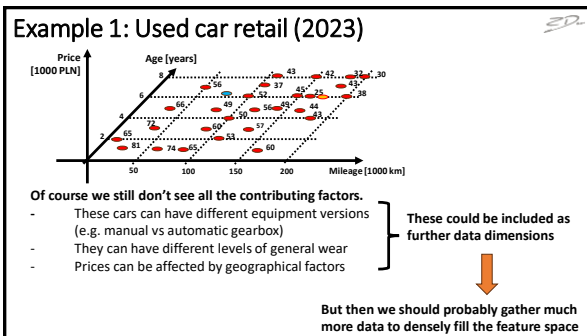
---

---

---

---

4




---

---

---

---

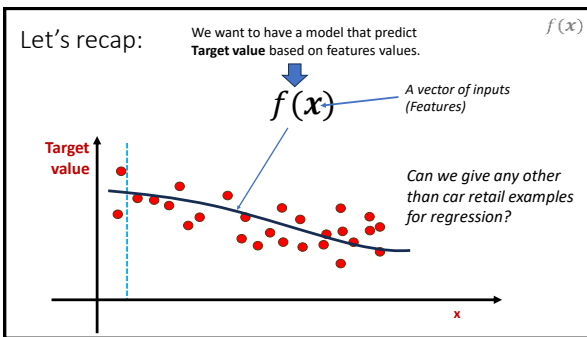
---

---

---

---

5




---

---

---

---

---

---

---

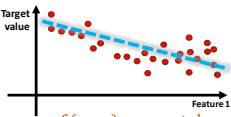
---

6

Let's recap: SD

We want to have a model that predict **Target value** based on features values and model parameters.

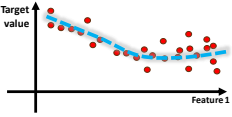
Linear model  
3D+: using hyperplane



$f(\mathbf{w}, \mathbf{x}) = \mathbf{w}\mathbf{x} + b$   
 $w_1x_1 + w_2x_2 + \dots + w_nx_n + b$

■ Features  
■ Model parameters

Nonlinear model  
3D+: using surface



We will still have a general  $f(\mathbf{w}, \mathbf{x})$  form, but the relations will be nonlinear now and depending on the model.

7

---

---

---

---

---

---

---

---

---

---

Regression means finding a **model** that estimates relationship between one data variable and the others SD

$$y = f(\mathbf{w}, \mathbf{x})$$

<p><b>Target value</b></p> <p>a.k.a: Dependent variable Outcome Response (Label)</p>	<p><b>Model parameters</b></p> <p>a.k.a: Weights Unknown parameters  (often also denoted as <math>\beta</math>)</p>	<p><b>Features</b></p> <p>a.k.a: Independent variables Explanatory variables Inputs Predictors</p>
--	---	--

8

---

---

---

---

---

---

---

---

---

---

Regression means finding a **model** that estimates relationship between one data variable and the others SD

$$y = f(\mathbf{w}, \mathbf{x})$$

We want to minimize error between known targets  $\mathbf{Y}$  and targets predicted by model for a known set of input data  $\mathbf{X}$  by adjusting model parameters  $\mathbf{w}$

For that we can use least squares minimization:

$$\arg \min_{\mathbf{w}} \sum_i (y_i - f(\mathbf{w}, \mathbf{x}_i))^2$$

9

---

---

---

---

---

---

---

---

---

---

### Linear regression

We want to find such line that sum of squares of these green line segments is as small as possible

**Model:**  
 $y = w_1x + w_b$

**Model fitting:**  
 $\arg \min_w \sum_i (y_i - (w_1x_i + w_b))^2$

10

---

---

---

---

---

---

---

---

### Locally weighted regression

A straight line does not really allow to model these data properly...

But what we actually use this line for, is to predict values for particular x

So maybe we could use a line model – with a small change of fitting method

11

---

---

---

---

---

---

---

---

### Locally weighted regression

We want to use the same model:  $y = w_1x + w_b$

This time we will fit it separately for each point so that neighboring points will be more important

Given a point  $x_i$  assign weights  $\alpha_i$  for each data sample  $x_i$  where  $\tau$  serves as a „width“ metaparameter

$$\alpha_i = e^{-\frac{(x_i - x_i)^2}{\tau}}$$

Lets answer what should be the model output (y) for this value of x

$$\arg \min_w \sum_i \alpha_i \cdot (y_i - (w_1x_i + w_b))^2$$

12

---

---

---

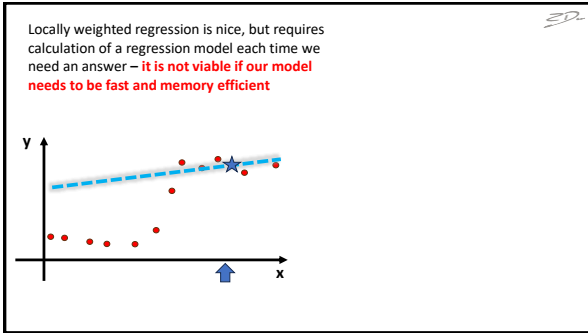
---

---

---

---

---



13

---

---

---

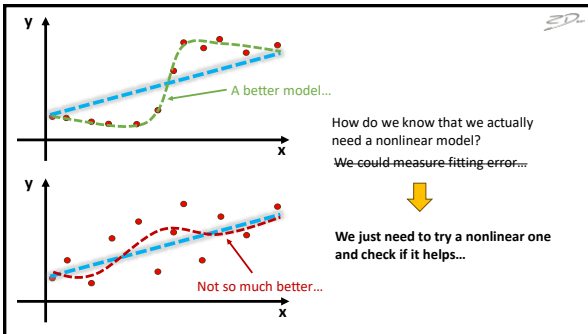
---

---

---

---

---



14

---

---

---

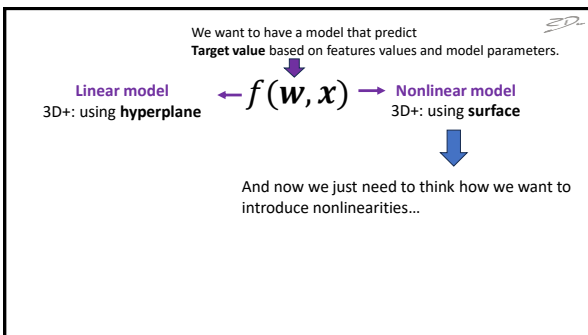
---

---

---

---

---



15

---

---

---

---

---

---

---

---

**Polynomial model** SD

$$f(w, x) = b + w_1x^1 + w_2x^2 + w_3x^3 + \dots$$

Linear model  
 Quadratic model

*Note that these are vectors of weights, not just single numbers!*

Polynomial models work just like linear models, with one exception – we actually need to set up a **metaparameter** – degree of the used polynomial

We do it usually by increasing the degree until the model stops improving significantly

---

---

---

---

---

---

---

---

16

**Is this all?** SD

*Gaussian mixture models*  
*Ensemble methods*   *Bayesian regression*  
**Support Vector Machine (SVM)**  
*Multivariate Adaptive Regression Splines*   *kNN regression*  
*Convolutional neural networks*   *Bayesian Neural Networks*  
**Artificial neural networks (ANNs)**   **Transformers**  
*Multilayered perceptrons*   *Echo state networks*  
*Radial neural networks*   **Long Short-Term Memory networks (LSTMs)**  
*Extreme learning machines*  
*Recurrent Neural Networks*

...we will learn some of these models along the way

---

---

---

---

---

---

---

---

17

**Overfitting** SD

**By changing model complexity (e.g. degree of the polynomial) we can adjust how well we can fit to the data**

*How do we know we went too far...?*  
*And why is this bad...?*

---

---

---

---

---

---

---

---

18

### Overfitting

**Overfitting** means that the model memorizes training samples at the cost of generalization capabilities

We recognize it by looking at the **error** on an **independent subset of data** (called validation subset). If it is significantly **higher** than for training subset – the model overfits.

19

---

---

---

---

---

---

---

---

### Regression:

We want to have a model that predict Target value based on features values.

$f(x)$

Sometimes our goal is not really to predict a value but to answer a yes-or-no questions...

- Is this car in a good shape?
- Is the mileage plausible?
- Can this car be a former taxi cab?

20

---

---

---

---

---

---

---

---

### Regression:

We want to have a model that predict Target value based on features values.

$f(x)$

### Classification:

We want to have a model that predict Target label based on features values.

21

---

---

---

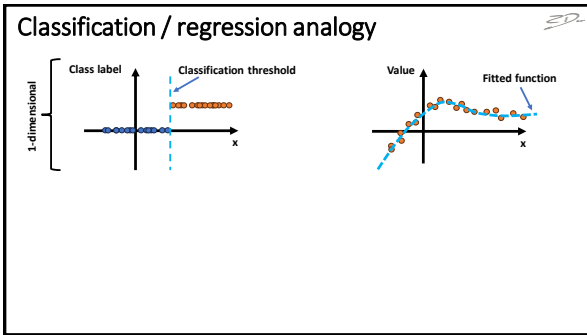
---

---

---

---

---



22

---

---

---

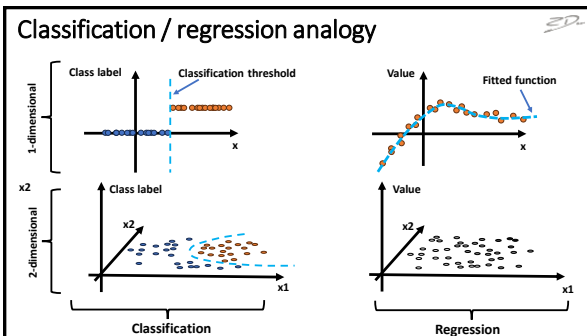
---

---

---

---

---



23

---

---

---

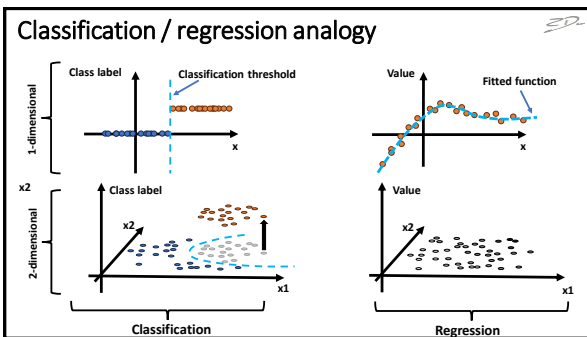
---

---

---

---

---



24

---

---

---

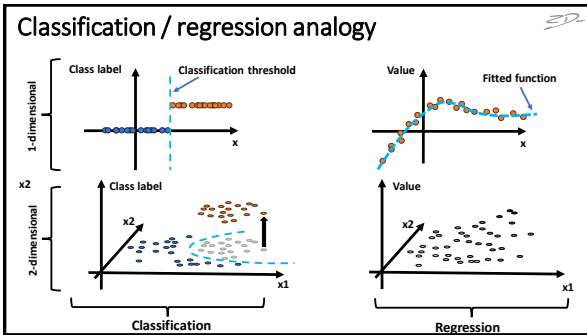
---

---

---

---

---



25

---

---

---

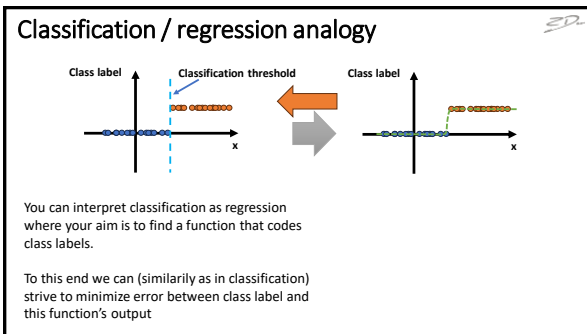
---

---

---

---

---



26

---

---

---

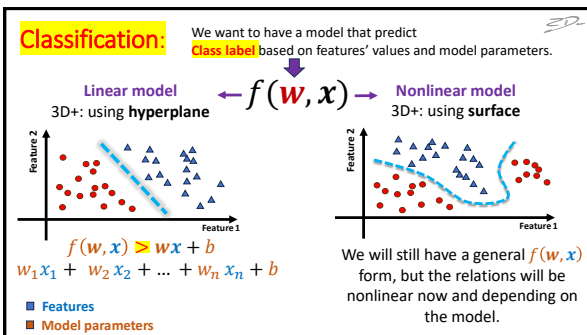
---

---

---

---

---



27

---

---

---

---

---

---

---

---



28

---

---

---

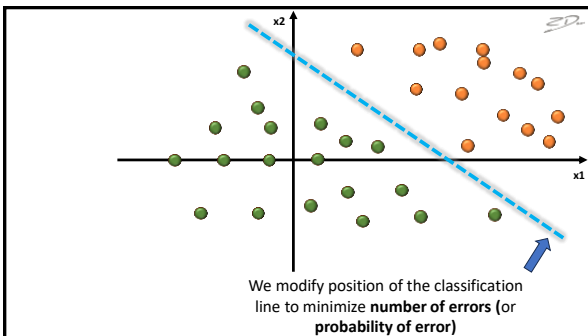
---

---

---

---

---



29

---

---

---

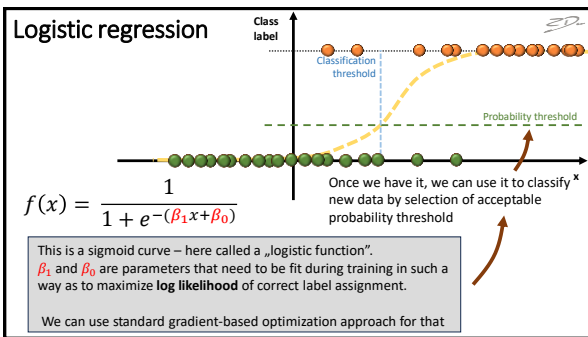
---

---

---

---

---



30

---

---

---

---

---

---

---

---

### Logistic regression SD

- LR is a classification equivalent of the linear regression
- It can work for multidimensional problems (we just have more parameters to learn)
- It can fit a „line classifier“ in the „best way possible“ (meaning that the classification has actually probabilistic interpretation)
- It cannot solve nonlinearly separable classification problems

31

---

---

---

---

---

---

---

---

### Things to remember: SD

1. Explain some real-life examples of regression (including new ones, not from the lecture!)
2. Explain generalized regression model  $y = f(w, x)$
3. Explain simple regression methods: linear, weighted, polynomial
4. Explain differences between linear and nonlinear models
5. Explain what overfitting is and how to avoid it (in regression context)
6. Explain how linear classification works (1D and 2D examples)
7. Explain logistic regression and its features

32

---

---

---

---

---

---

---

---