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)

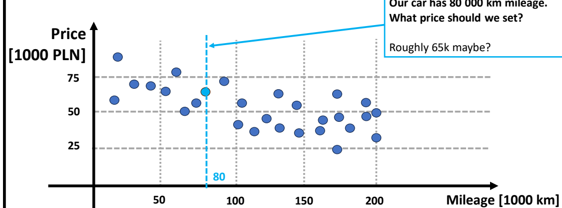
We have a **7-year old** Opel Astra, with **80 000 km mileage**.
We want to **sell it quickly** with as **high price** as possible.

2

Example 1: Used car retail (2023)

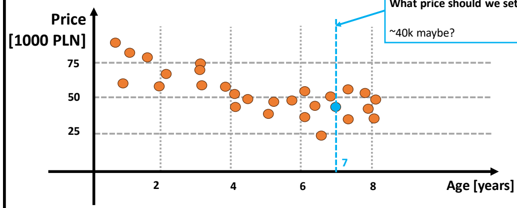
Our car has 80 000 km mileage.
What price should we set?

Roughly 65k maybe?



3

Example 1: Used car retail (2023)

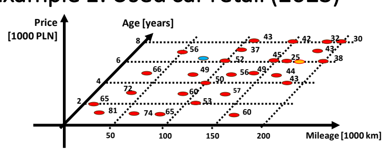


Our car is 7 years old.
What price should we set now?

~40k maybe?

4

Example 1: Used car retail (2023)



Of course we still don't see all the contributing factors.

- These cars can have different equipment versions (e.g. manual vs automatic gearbox)
- They can have different levels of general wear
- Prices can be affected by geographical factors

These could be included as further data dimensions

But then we should probably gather much more data to densely fill the feature space

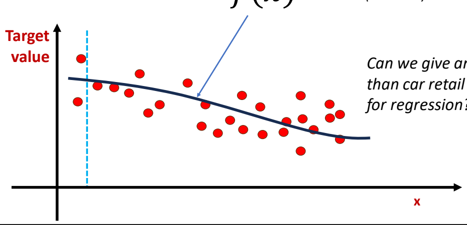
5

Let's recap:

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

$f(\boldsymbol{x})$

A vector of inputs (Features)



Can we give any other than car retail examples for regression?

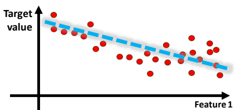
6

Let's recap:

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

$$f(\boldsymbol{w}, \boldsymbol{x})$$

Linear model
3D+: using **hyperplane**

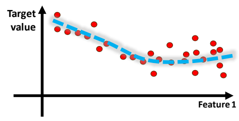


$f(\boldsymbol{w}, \boldsymbol{x}) = \boldsymbol{w}\boldsymbol{x} + b$

$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$

- **Features**
- **Model parameters**

Nonlinear model
3D+: using **surface**



We will still have a general $f(\boldsymbol{w}, \boldsymbol{x})$ 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

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

Target value

a.k.a:
Dependent variable
Outcome
Response
(Label)

Model parameters

a.k.a:
Weights
Unknown parameters

(often also denoted as $\boldsymbol{\beta}$)

Features

a.k.a:
Independent variables
Explanatory variables
Inputs
Predictors

8

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

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

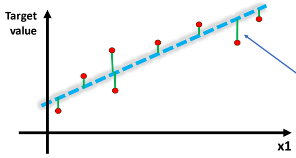
We want to minimize error between known targets $\boldsymbol{Y}$ and targets predicted by model for a known set of input data $\boldsymbol{X}$ by adjusting model parameters $\boldsymbol{w}$

For that we <u>can</u> use least squares minimization:

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

9

Linear regression



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

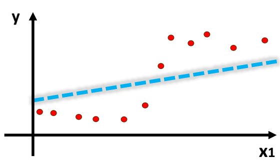
Model fitting:

Model:

$$y = w_1 x + w_b \quad \Rightarrow \quad arg\,min_w \sum_i (y_i - (w_1 x_i + w_b))^2$$

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

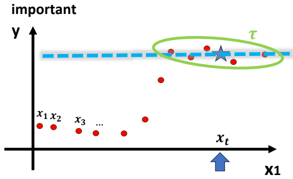
Locally weighted regression

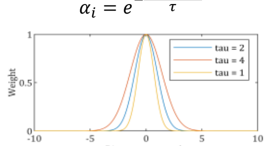
We want to use the same model: $y = w_1 x + w_b$

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

Given a point $\boldsymbol{x_t}$ assign weights $\boldsymbol{\alpha_i}$ for each data sample $\boldsymbol{x_i}$ where $\boldsymbol{\tau}$ serves as a „width" metaparameter

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



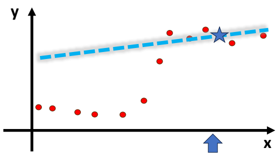


Lets answer what should be the model output (y) for this value of x

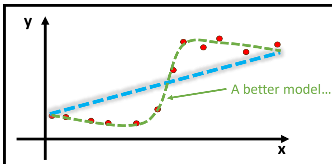
$$arg\,min_w \sum_i \alpha_i \cdot (y_i - (w_1 x_i + w_b))^2$$

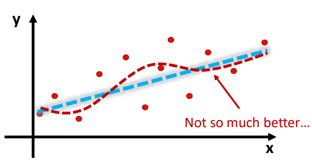
12

Locally weighted regression is nice, but requires calculation of a regression model each time we need an answer – **it is not viable if our model needs to be fast and memory efficient**



13





How do we know that we actually need a nonlinear model?

~~We could measure fitting error…~~

We just need to try a nonlinear one and check if it helps…

14

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

Linear model
3D+: using **hyperplane**

$$f(\boldsymbol{w}, \boldsymbol{x})$$

Nonlinear model
3D+: using **surface**

And now we just need to think how we want to introduce nonlinearities…

15

Polynomial model

$$f(\boldsymbol{w},\boldsymbol{x}) = \underbrace{\underbrace{b + \boldsymbol{w_1x^1}}_{\text{Linear model}} + \boldsymbol{w_2x^2}}_{\text{Quadratic model}} + \boldsymbol{w_3x^3} + \dots$$

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?

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

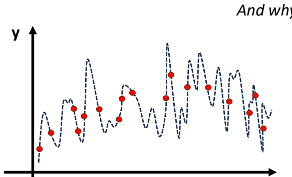
…we will learn some of these models along the way

17

Overfitting

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…?

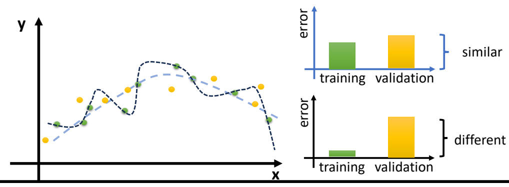


18

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

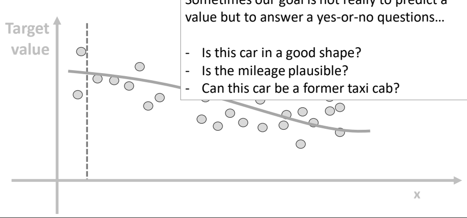
Regression:

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.

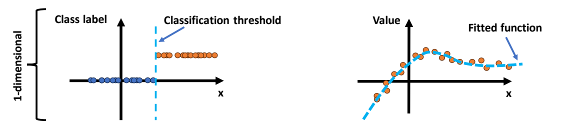
$$f(x)$$

Classification:

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

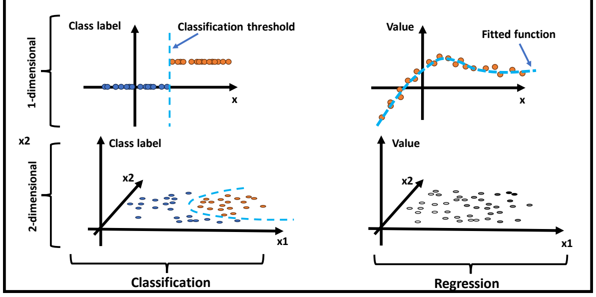
21

Classification / regression analogy



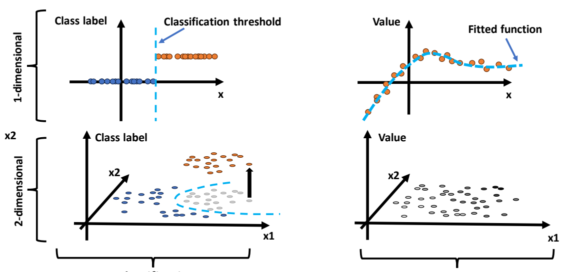
22

Classification / regression analogy



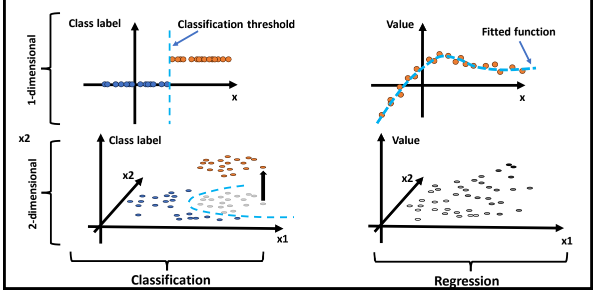
23

Classification / regression analogy



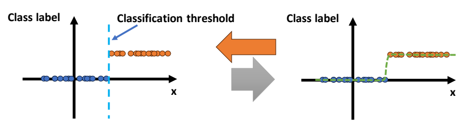
24

Classification / regression analogy



25

Classification / regression analogy



You can interpret classification as regression where your aim is to find a function that codes class labels.

To this end we can (similarily as in classification) strive to minimize error between class label and this function's output

26

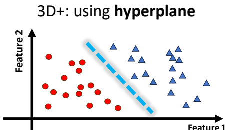
Classification:

We want to have a model that predict **Class label** based on features' values and model parameters.

Linear model $\leftarrow f(\boldsymbol{w}, \boldsymbol{x}) \rightarrow$ **Nonlinear model**

3D+: using **hyperplane**

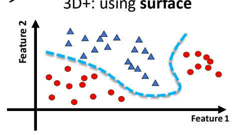
3D+: using **surface**



$f(\boldsymbol{w}, \boldsymbol{x}) > \boldsymbol{wx} + b$

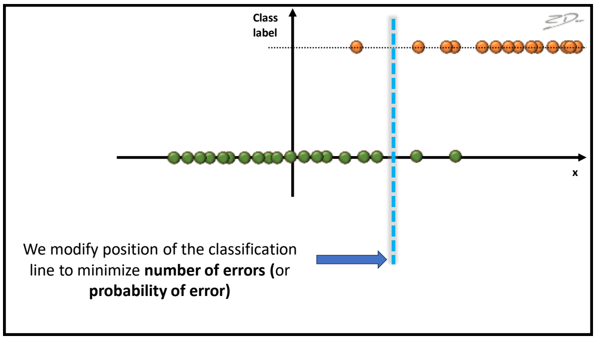
$w_1 x_1 + \; w_2 x_2 + \ldots + w_n x_n + b$

- **Features**
- **Model parameters**

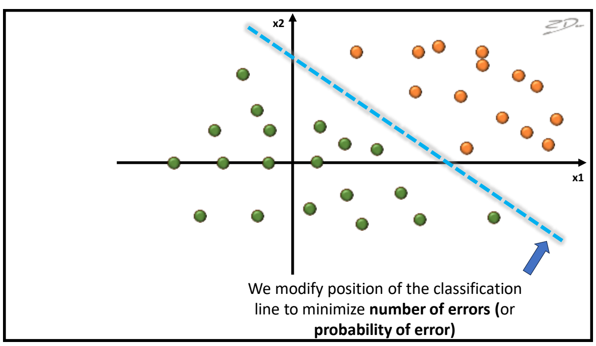


We will still have a general $f(\boldsymbol{w}, \boldsymbol{x})$ form, but the relations will be nonlinear now and depending on the model.

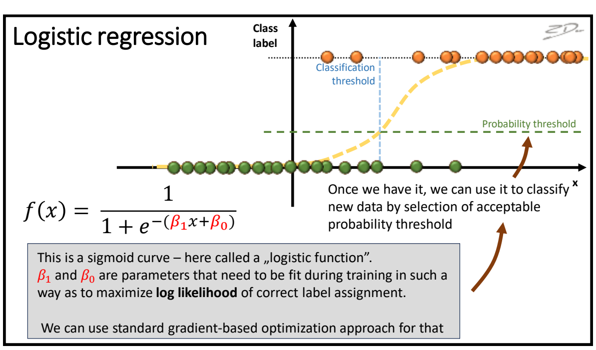
27


Class label
x
We modify position of the classification line to minimize **number of errors** (or **probability of error)**

28


x2
x1
We modify position of the classification line to minimize **number of errors** (or **probability of error)**

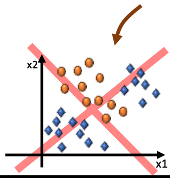
29

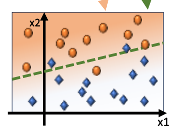

Logistic regression
Class label
Classification threshold
Probability threshold
x
$f(x) = \frac{1}{1 + e^{-(\beta_1 x + \beta_0)}}$
Once we have it, we can use it to classify new data by selection of acceptable probability threshold
This is a sigmoid curve – here called a „logistic function”.
β_1 and β_0 are parameters that need to be fit during training in such a way as to maximize **log likelihood** of correct label assignment.
We can use standard gradient-based optimization approach for that

30

Logistic regression

- 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:

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