Metrics

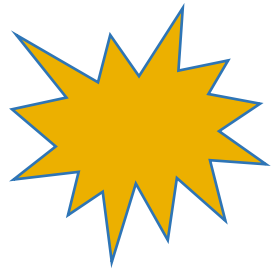
Introduction to Data Science

9th lecture

Izv. prof. dr. sc. Ana Sović Kržić

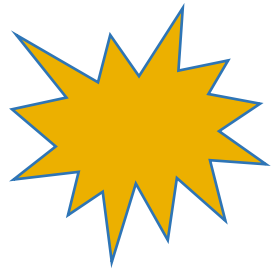
2025/2026

Metrics



- used to monitor and measure model performance (during learning and testing)
- Different metrics for different tasks:
  - Classification
  - Regression
  - Ranking
  - Image processing
  - Deep learning
  - NLP

Classification



- **applications**: facial recognition, YouTube video categorization, content moderation, medical diagnoses, text classification, hate speech detection
- **popular models**: support vector machine (SVM), logistic regression, decision trees, random forests, XGboost, convolutional neural networks, recurrent neural network

Confusion matrix

- it is not a metric, it is a tabular visualization of the obtained predictions in relation to the annotations
- diagonal elements – correct predictions for different classes
- off-diagonal elements – the number of samples that are wrongly classified

		Real classes	
		Class I	Class II
Predicted classes	Class I	TP	FP
	Class II	FN	TN

Confusion matrix - example

- binary classification whether the picture shows a dog or not a dog
- the test set has 1100 images: 100 images represent dogs and 1000 do not contain dogs
- 80 images containing dogs are correctly predicted (true-positive TP), 20 are not (false negative FN)
- out of 1000 images that do not have dogs: 950 are predicted correctly (true-negative TN), 50 are predicted incorrectly (false-positive FP)

		Real classes	
		Dogs	Not dogs
Predicted classes	Dogs	80	50
	Not dogs	20	950

Accuracy (1)

- **the ratio of correct in relation to the total number of predictions**

$$CA = \frac{TP + TN}{TP + TN + FP + FN}$$

- example:

$$CA = \frac{80 + 950}{80 + 950 + 20 + 50} = 93.6\%$$

Accuracy (2)

- it is not a good measure if the distribution of classes is unbalanced = when we have many more of one class than others → it may happen that all samples are predicted to be from the most common class → the accuracy will be high, but the model does not predict anything, but simply puts everything in the most common class
- for example, all images are predicted as "not dogs"

$$\mathrm{CA} = \frac{0 + 1000}{0 + 1000 + 0 + 100} = 90.9\%$$

		Real classes	
		Dogs	Not dogs
Predicted classes	Dogs	0	0
	Not dogs	100	1000

Precision

- performance measure **of a particular class** (by rows)

$$PR = \frac{TP}{TP + FP}$$

$$PR = \frac{TN}{TN + FN}$$

- example:
  - PR for correctly predicted dogs: $PR = \frac{80}{80+50} = 61.5\%$
  - PR for correctly predicted that there are no dogs in image: $PR = \frac{950}{950+20} = 97.9\%$
  - much higher accuracy for predicting that there are no dogs in the image compared to predicting that there are dogs in the image → because there are many more images that do not have dogs in them

Recall

- **ratio of samples of the class that is correctly predicted** (by column)

$$RE = \frac{TP}{TP + FN}$$

$$RE = \frac{TN}{TN + FP}$$

- Example:
  - RE for correctly predicted dog: $RE = \frac{80}{80+20} = 80\%$
  - RE for correctly predicted that there are no dogs in image:

$$RE = \frac{950}{950 + 50} = 95\%$$

F1 score

- combination of PR and RE – harmonic mean of PR and RE

$$F1 = \frac{2 \cdot PR \cdot RE}{PR + RE}$$

- trade-off between PR and RE of a model – if PR is too large, RE becomes very small and vice versa

- Example: $F1 = \frac{2 \cdot 0.615 \cdot 0.8}{0.615 + 0.8} = \frac{0.984}{1.415} = 69.5\%$

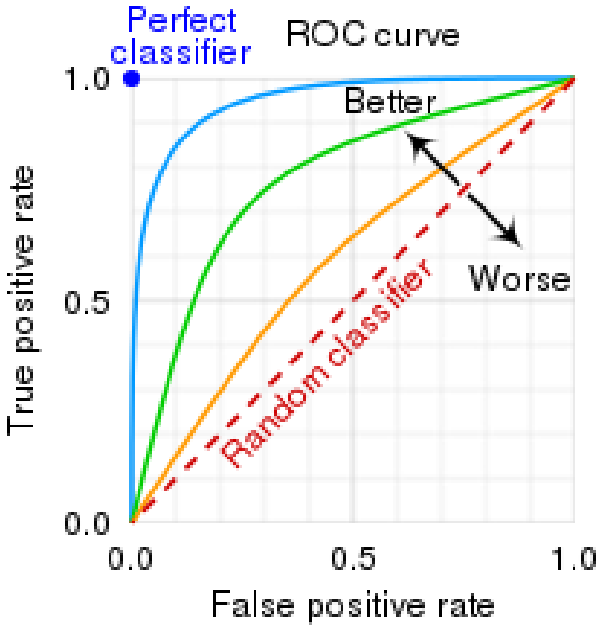
ROC curve

- receiver operating characteristic curve
- graphical representation of the performance of the **binary classifier as a function of the different thresholds** used in the classification
- true positive rate (TPR) in relation with false positive rate (FPR) for different thresholds
- **true positive rate (TPR)** (= recall)

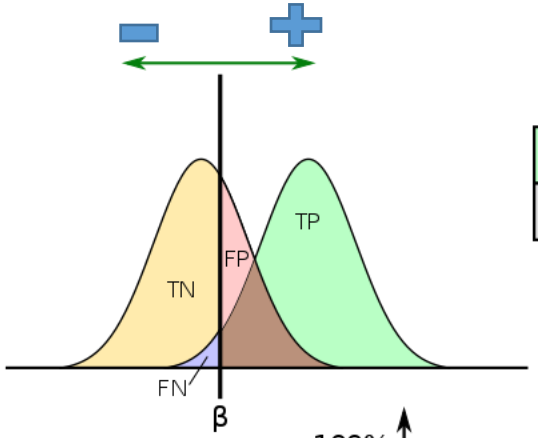
$$TPR = \frac{TP}{TP + FN}$$

- **false positive rate (FPR)**

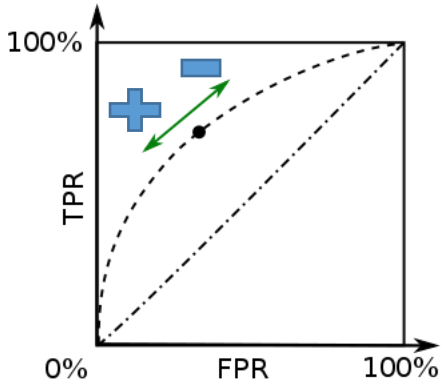
$$FPR = \frac{FP}{FP + TN}$$



ROC curve - example



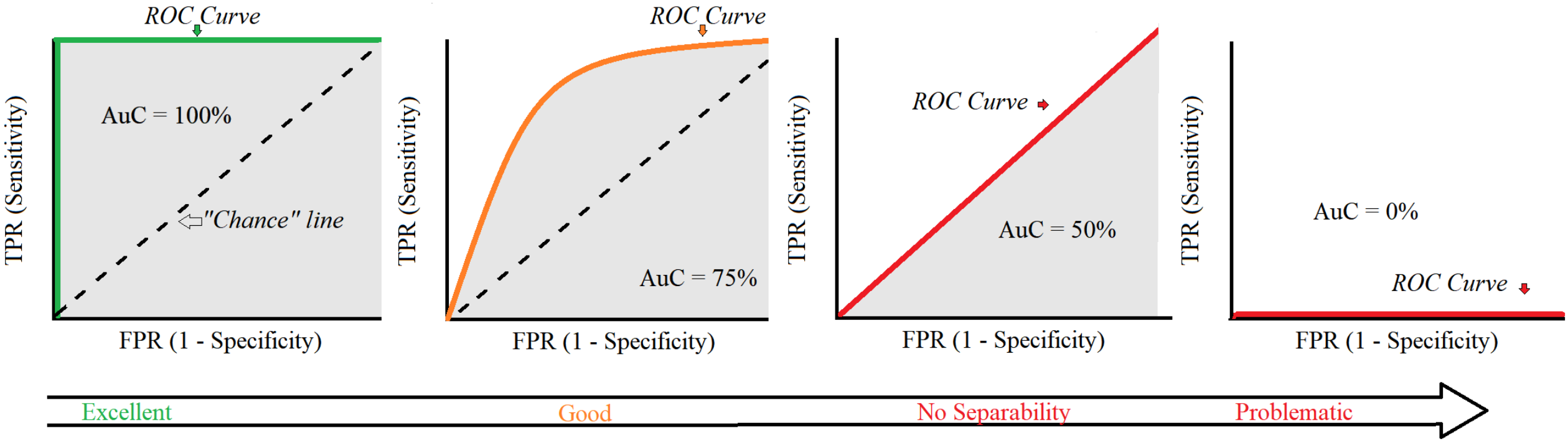
TP	FP
FN	TN



- most classification models work using **probabilities** – they predict the probability that there is a dog in the image
- the probability is **compared with the given threshold** – if the obtained probability is higher than the threshold, the image shows a dog, otherwise it does not
- e.g. obtained probabilities for 4 images are: $[0.4, 0.6, 0.3, 0.8]$.
- With different thresholds, we have:
  - threshold = 0.2 → prediction = $[1, 1, 1, 1]$
  - threshold = 0.5 → prediction = $[0, 1, 0, 1]$
  - threshold = 0.7 → prediction = $[0, 0, 0, 1]$
- lower threshold → more images will be predicted as positive: higher TPR (RE), higher FPR → right side of the curve
- the threshold is determined based on the ROC curve - in order to get a good ratio of TPR and FPR
- **ROC curve shows TPR and FPR for different thresholds**
- **ROC krivulja prikazuje TPR i FPR za različite pragove**

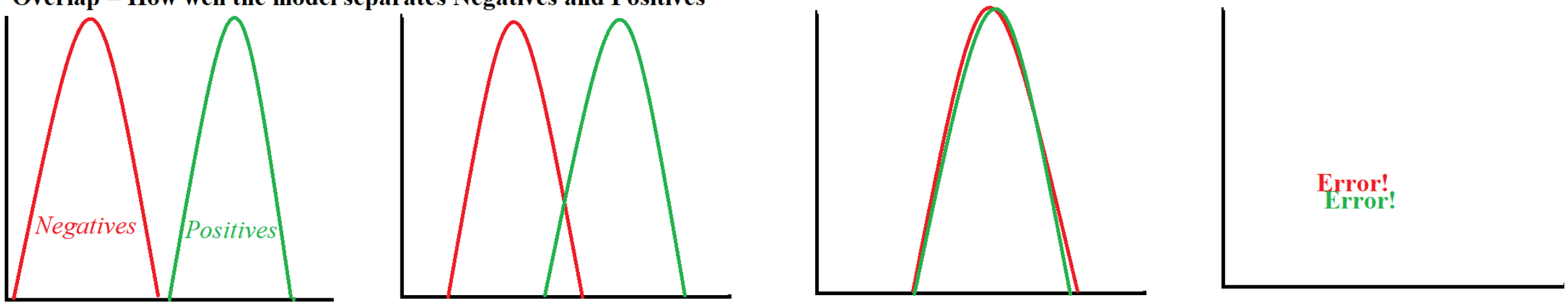
AUC

- area under the curve (AUC)
- an aggregated measure of the performance of a binary classifier for all possible threshold values (and is therefore threshold invariant)
- the probability that the model will rank a random positive example higher than a random negative example
- **the area under the ROC curve is between 0 and 1**
- higher AUC – better model
  - a model whose predictions are 100% wrong → AUC = 0.0
  - a model whose predictions are 100% correct → AUC = 1.0

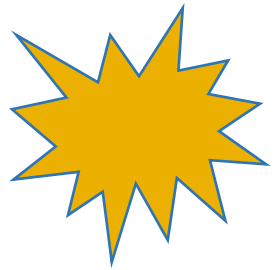


Excellent — Good — No Separability — Problematic

Overlap = How well the model separates Negatives and Positives



Regression



- regression is used to predict continuous values
- **applications**: house price forecasting, stock price forecasting, weather forecasting, image superresolution, image compression
- **used models**: linear regression, random forests, XGboost, convolutional neural networks, recurrent neural networks

Mean square error MSE

- **mean square error** between the predicted and actual values

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

RMSE

- **root of MSE**
- error measure has the same unit as the observed values
- average deviation in the model from the target value

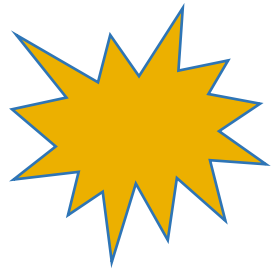
Mean absolute value MAE

- mean absolute value (MAE)
- the **mean absolute distance** between the predicted and actual values

$$MSE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

- more robust to outliers than MSE - because MSE squares the error, and outliers have a large error, so squaring it, it increases even more

References



https://www.v7labs.com/blog/data-labeling-guide

https://appen.com/blog/data-labeling/

https://www.cloudfactory.com/data-labeling-guide

https://labelyourdata.com/articles/data-labeling-quality-and-how-to-measure-it

https://towardsdatascience.com/cronbachs-alpha-theory-and-application-in-python-d2915dd63586

https://www.statology.org/cronbachs-alpha-in-python/

https://hackernoon.com/introduction-5-different-types-of-text-annotation-in-nlp-78523wwo

https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce

https://towardsdatascience.com/20-popular-machine-learning-metrics-part-2-ranking-statistical-metrics-22c3e5a937b6