Model Evaluation

7B

CS 3244
Machine Learning

NUS National University of Singapore | Computing

[Instructor] Brian Lim

brianlim@comp.nus.edu.sg

Academic Experience

- Asst. Prof. in Computer Science
- Ph.D. in HCI, Carnegie Mellon University
- B.S. in Engineering Physics, Cornell University

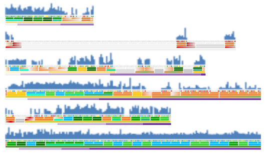
Research Interests

- Explainable Artificial Intelligence
- Human-Computer Interaction
- Ubiquitous Computing
- Data analysis and visualization
- Smart Health and Smart Cities

IoT Sensors

Health Behavior Change

Data Analytics

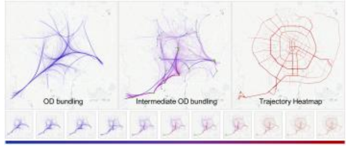


NUS Ubicomp Lab

Apps and Analytics for Smart Cities and Healthcare

http://ubiquitous.comp.nus.edu.sg

Explainable Artificial Intelligence



Interactive Data Visualization

Week 07b: Learning Outcomes

- Describe various evaluation metrics of model performance
- Understand that model performance depends on prediction task and data
  - Describe several challenges in evaluating model performances
  - Choose appropriate **evaluation metric** for different prediction tasks

Week 07b: Lecture Outline

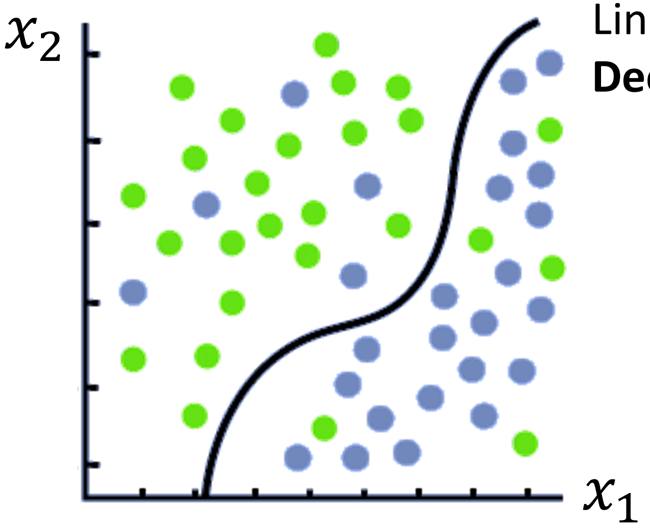
1. Recap: Supervised learning Classification vs. Regression
2. Classification Metrics
3. Regression Metrics
4. Unsupervised learning metrics [W12]

Classification vs. Regression

Classification

$y \in \{0,1\}$ binary

$y \in \{y_A, y_B, \ldots\}$ multi-class

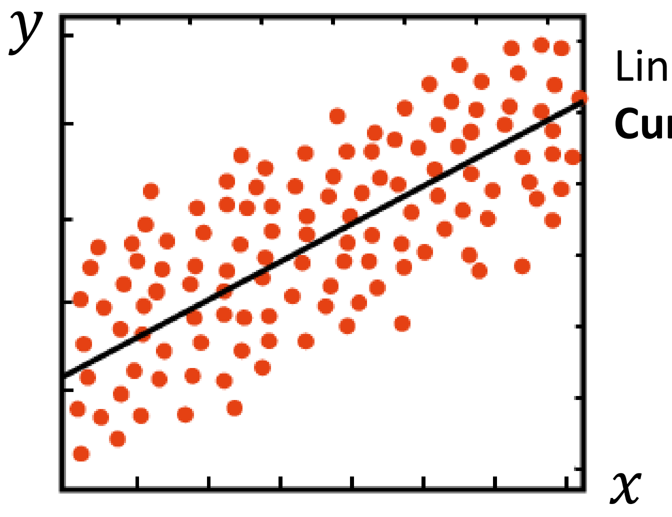


Line is a
Decision Boundary

Note
different axes!
The are not
the same type

Regression

$y \in \mathbb{R}$ any real number



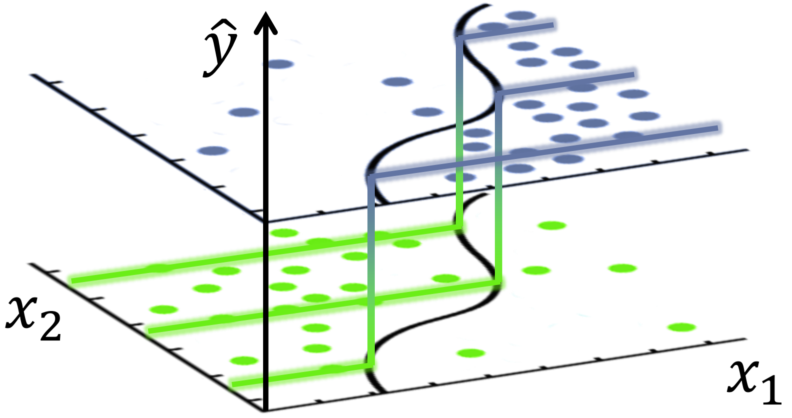
Line is a
Curve Fit

Image credit:
https://www.javatpoint.com/regression-vs-classification-in-machine-learning

Classification

$y \in \{0,1\}$ binary

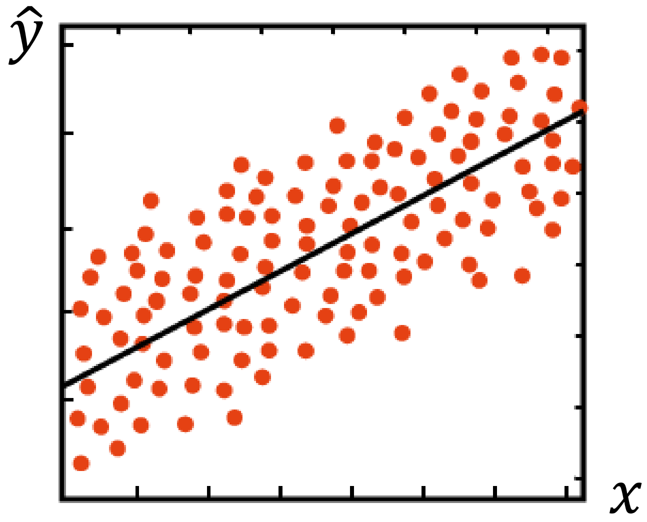
$y \in \{y_A, y_B, \dots\}$ multi-class



$$y = M(\boldsymbol{x}), \quad \boldsymbol{x} = \vec{x} = (x_1, x_2)^\top$$

Regression

$y \in \mathbb{R}$ any real number



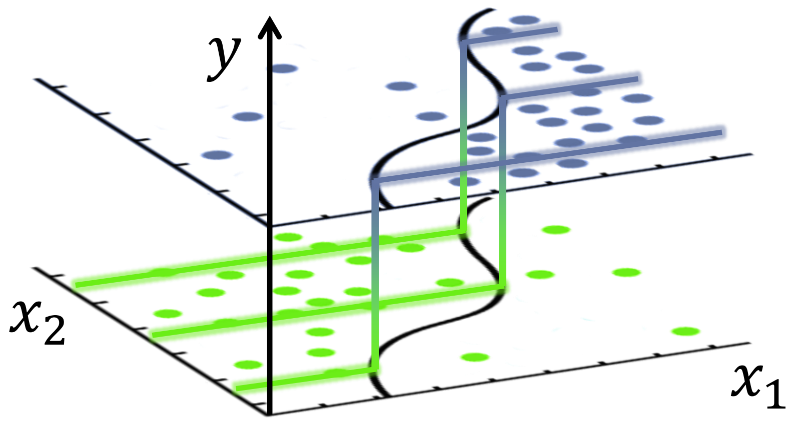
$$y = M(x), \quad x = x_1$$

Image credit:
https://www.javatpoint.com/regression-vs-classification-in-machine-learning

Classification

$y \in \{0,1\}$ binary

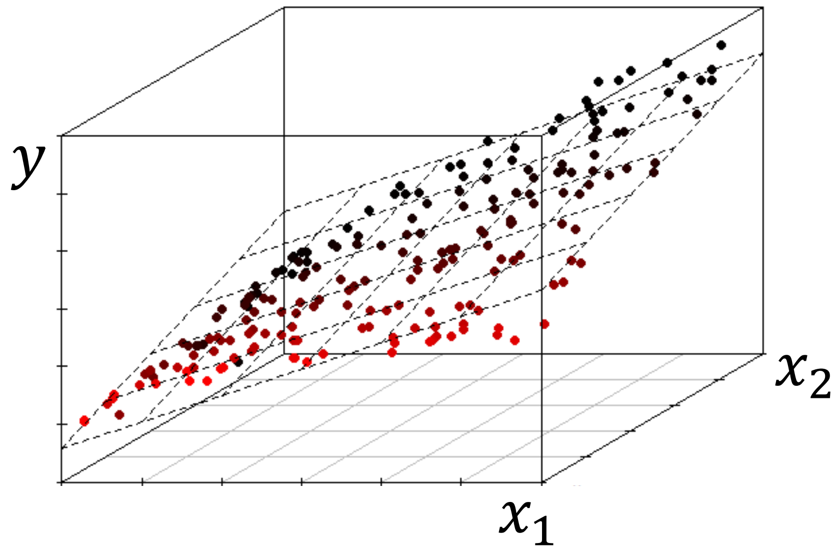
$y \in \{y_A, y_B, \dots\}$ multi-class



$$y = M(\boldsymbol{x}), \quad \boldsymbol{x} = \vec{x} = (x_1, x_2)^\top$$

Regression

$y \in \mathbb{R}$ any real number



$$y = M(x), \quad x = x_1$$

$$y = M(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, \dots, x_n)^\top$$

Image credit:
https://www.javatpoint.com/regression-vs-classification-in-machine-learning,
https://stackoverflow.com/q/26431800

Classification Evaluation Metrics

NUS National University of Singapore | Department of Computer Science
School of Computing

Week 07: Lecture Outline

1. Recap: Classification vs. Regression
2. Classification Metrics
    1. Accuracy
    2. Confusion Matrix, TP, TN, FP, FN
    3. Precision, Recall, F_1
    4. ROC, AUC
    5. Micro- and Macro-Averaging
    6. PR-AUC (Average Precision)
3. Regression Metrics

Accuracy

"Average correctness" across test dataset with m instances:

$$A = \frac{1}{m} \sum_{j=1}^{m} [\hat{y}_j = y_j]$$

where

- $\hat{y}_j = M(\boldsymbol{x}_j)$ is the predicted value from model M of the jth instance $\boldsymbol{x}_j$
- y_j is the ground truth value of the jth instance
- $[P] = \begin{cases} 1 & \text{if } P \text{ is true} \\ 0 & \text{otherwise} \end{cases}$ is the Iverson bracket notation for if/else

Confusion Matrix

Student alertness prediction

Inst.	Predicted $\hat{y}$	Actual y	
1	Alert	Alert	TP
2	Alert	Alert	
3	Sleepy	Alert	FN
4	Sleepy	Alert	
5	Sleepy	Alert	
6	Sleepy	Sleepy	TN
7	Sleepy	Sleepy	
8	Sleepy	Sleepy	
9	Sleepy	Sleepy	
10	Alert	Sleepy	FP

Is the student **Alert**?

		Actual Label	
		Pos	**Neg**
Predicted Label	**Pos**	**TP** True Positive	**FP** False Positive
	Neg	**FN** False Negative	**TN** True Negative

$$\text{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+TN}}$$

Confusion Matrix

Student alertness prediction

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy

Is the student **Alert**?

		Actual Label	
		Pos	Neg
Predicted Label	Pos	TP True Positive	FP False Positive
	Neg	FN False Negative	TN True Negative

Is the student **Sleepy**?

		Actual Label	
		Pos	Neg
Predicted Label	Pos	TP True Positive	FP False Positive
	Neg	FN False Negative	TN True Negative

Which class is Positive? Negative?

You define based on your application

Confusion Matrix

Student alertness prediction

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Not	Alert
4	Not	Alert
5	Not	Alert
6	Not	Not
7	Not	Not
8	Not	Not
9	Not	Not
10	Alert	Not

Is the student **Alert**?

		Actual Label	
		Pos	Neg
Predicted Label	Pos	TP True Positive	FP False Positive
	Neg	FN False Negative	TN True Negative

Two types of False mistakes. Which is worse?

- FN: Accuse **alert** students, or
- FP: Neglect **sleepy** students?

Confusion Matrix: which mistake is costlier?

Student alertness prediction

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Not	Alert
4	Not	Alert
5	Not	Alert
6	Not	Not
7	Not	Not
8	Not	Not
9	Not	Not
10	Alert	Not

Actual Label

Predicted Label	Pos	Neg
Pos	2 True Positive	1 False Positive
Neg	3 False Negative	4 True Negative

1/10 students predicted as **Alert** when actually **Sleepy**
=> Few undeserved credits

3/10 students predicted as **Sleepy** when actually **Alert**
=> Many false accusations

Cost-Sensitive Evaluation Metrics

1. Report **Precision** vs. **Recall**
2. Vary **Prediction Threshold**

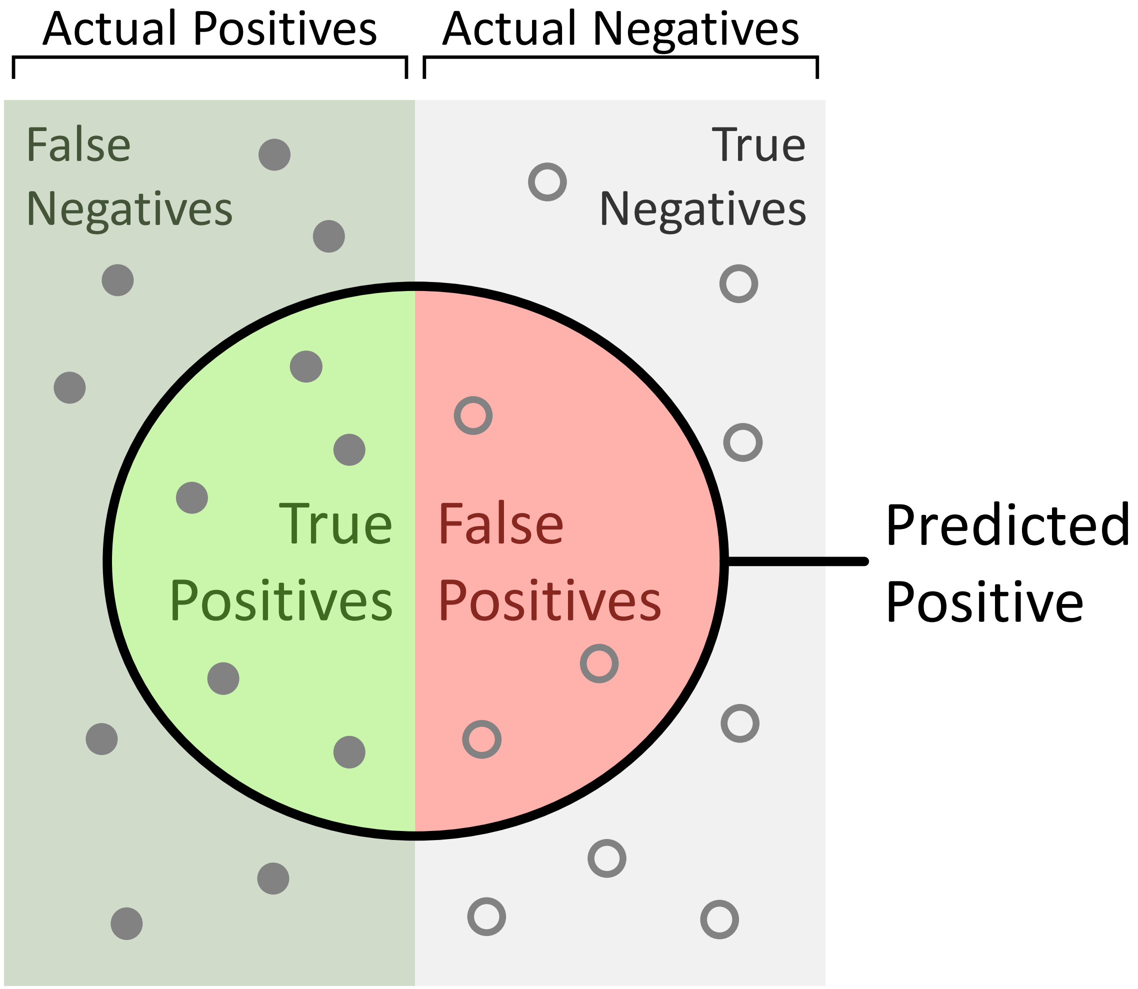
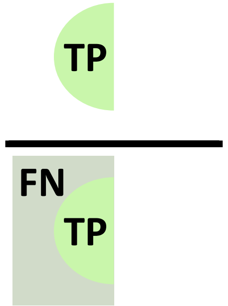
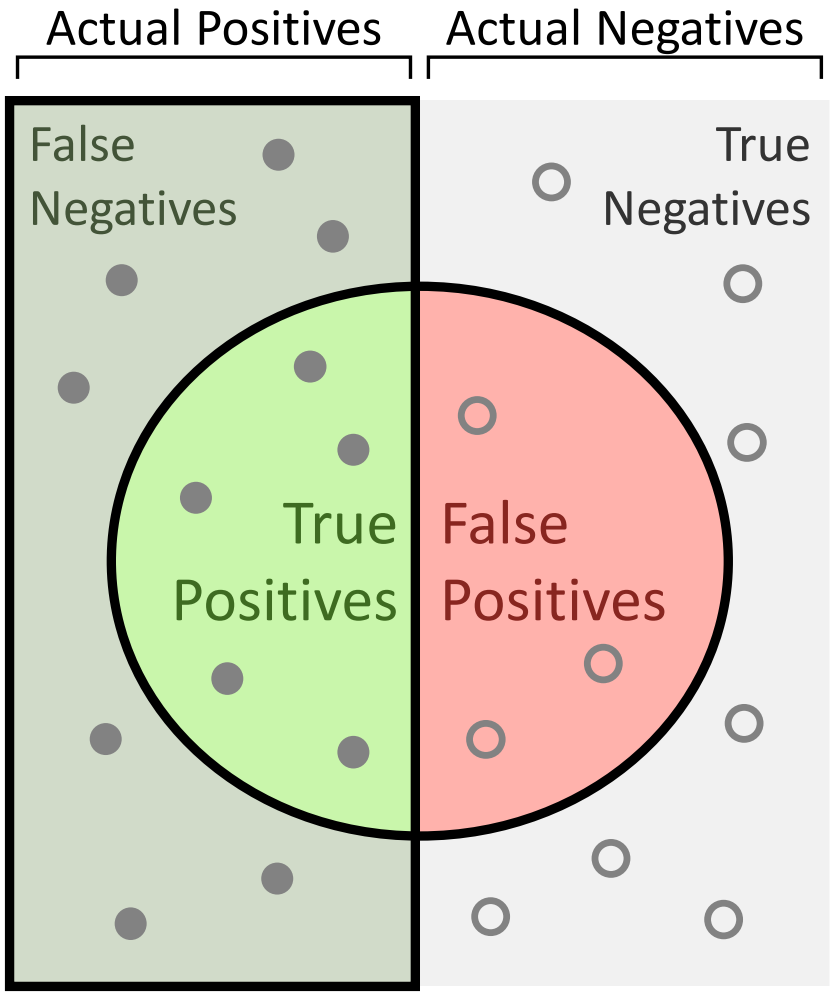


Image credit:
https://en.wikipedia.org/wiki/Precision_and_recall

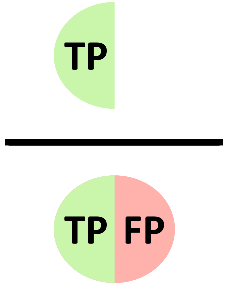
Among actual positives, what fraction of instances were **recalled**?

Recall = 

Maximize this if false negative (FN) is costly.
E.g., cancer prediction, not music recommendation



Among positive predictions, how **precisely** were actual positive instances predicted?

Precision = 

Maximize this if false positive (FP) is costly.
E.g., email spam, satellite launch date prediction.

Image credit: https://en.wikipedia.org/wiki/Precision_and_recall

Precision and Recall

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Not	Alert
4	Not	Alert
5	Not	Alert
6	Not	Not
7	Not	Not
8	Not	Not
9	Not	Not
10	Alert	Not

Actual Label

Predicted Label	Alert	Not	
Alert	2 True Positive	1 False Positive	3 ∑ Pred. Pos.
Not	3 False Negative	4 True Negative	7 ∑ Pred. Neg.
	5 ∑ Actual Pos.	5 ∑ Actual Neg.	

Precision
P = TP / (TP+FP)

Recall
R = TP / (TP+FN)

Precision and Recall → F_1 Score

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Not	Alert
4	Not	Alert
5	Not	Alert
6	Not	Not
7	Not	Not
8	Not	Not
9	Not	Not
10	Alert	Not

Actual Label

Predicted Label	Alert	Not	
Alert	2 True Positive	1 False Positive	3 ∑ Pred. Pos.
Not	3 False Negative	4 True Negative	7 ∑ Pred. Neg.
	5 ∑ Actual Pos.	5 ∑ Actual Neg.	

Precision
P = TP / (TP+FP)

Recall
R = TP / (TP+FN)

F_1 Score

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

F_1 Score: Why not just use simple average?

1. The measure is more **robust** (less sensitive to extreme values)

Ref: https://stackoverflow.com/a/26360501

2. It considers that the numerators of P and R are the same, so it compares their denominators

$$F_1 = \left(\frac{P^{-1} + R^{-1}}{2}\right)^{-1} = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2\,TP}{(TP + FP) + (TP + FN)}$$

Other "fairer" metrics that consider true negatives (TN):
Matthews correlation coefficient, Youden's index, Cohen's kappa

Imbalanced Data

Balanced

Predicted Label \ Actual Label	Alert	Not	
Alert	2 True Positive	1 False Positive	3 ∑ Pred. Pos.
Not	3 False Negative	4 True Negative	7 ∑ Pred. Neg.
	5 ∑ Actual Pos.	5 ∑ Actual Neg.	

Precision $P = .66$

Accuracy $A = .60$

Recall $R = .40$

F_1 Score $F_1 = .50$

Imbalanced

Predicted Label \ Actual Label	Alert	Not	
Alert	2 True Positive	10 False Positive	12 ∑ Pred. Pos.
Not	3 False Negative	40 True Negative	43 ∑ Pred. Neg.
	5 ∑ Actual Pos.	50 ∑ Actual Neg.	

Precision $P = .17$

Accuracy $A = .76$

Recall $R = .40$

F_1 Score $F_1 = .24$

Imbalanced Data

		Actual Label: Alert	Actual Label: Not	
Predicted Label	Alert	2 True Positive	1 False Positive	3 ∑ Pred. Pos.
	Not	3 False Negative	4 True Negative	7 ∑ Pred. Neg.
		5 ∑ Actual Pos.	5 ∑ Actual Neg.	

Precision $P = .66$

Accuracy $A = .60$

Recall $R = .40$

F_1 Score $F_1 = .50$

		Actual Label: Alert	Actual Label: Not	
Predicted Label	Alert	36 True Positive	1 False Positive	37 ∑ Pred. Pos.
	Not	14 False Negative	4 True Negative	18 ∑ Pred. Neg.
		50 ∑ Actual Pos.	5 ∑ Actual Neg.	

Precision $P = .97$

Accuracy $A = .73$

Recall $R = .40$

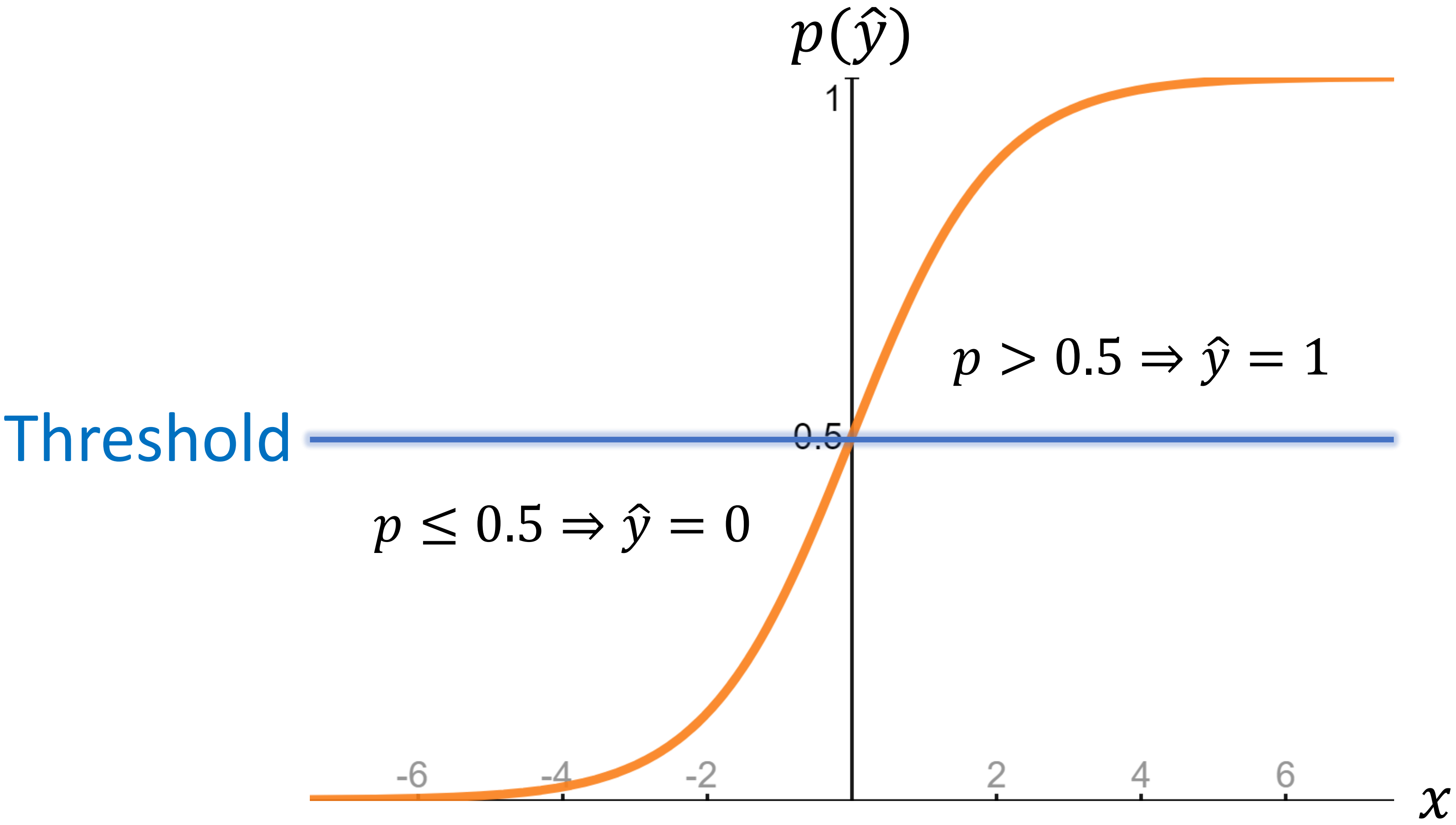
F_1 Score $F_1 = .57$

Cost-Sensitive Evaluation Metrics

1. Report **Precision** vs. **Recall**
2. Vary **Prediction Threshold**

Prediction Confidence

Logistic regression example



Cost-Sensitive Confusion Matrix (Threshold = **0.5**)

Inst.	**Confidence** $p(\hat{y})$	**Prediction** $\hat{y}$ $p(\hat{y}) > 0.5$	**Actual** y
1	0.9	Alert	Alert
2	0.6	Alert	Alert
3	0.5	Not	Alert
4	0.4	Not	Alert
5	0.3	Not	Alert
6	0.2	Not	Not
7	0.3	Not	Not
8	0.4	Not	Not
9	0.5	Not	Not
10	0.55	Alert	Not

Actual Label

Predicted Label	Alert	Not	
Alert	2 True Positive	1 False Positive	3 $\sum$ Pred. Pos.
Not	3 False Negative	4 True Negative	7 $\sum$ Pred. Neg.
	5 $\sum$ Actual Pos.	5 $\sum$ Actual Neg.	

True Positive Rate
TPR = TP / (TP+FN)

False Positive Rate
FPR = FP / (FP+TN)

Cost-Sensitive Confusion Matrix (Threshold = **0.5**)

Inst.	**Confidence** $p(\hat{y})$	**Prediction** $\hat{y}$ $p(\hat{y}) > 0.5$	**Actual** y
1	0.9	Alert	Alert
2	0.6	Alert	Alert
3	0.5	Not	Alert
4	0.4	Not	Alert
5	0.3	Not	Alert
6	0.2	Not	Not
7	0.3	Not	Not
8	0.4	Not	Not
9	0.5	Not	Not
10	0.55	Alert	Not

Actual Label

Predicted Label	Alert	Not	
Alert	2 True Positive	1 False Positive	3 ∑ Pred. Pos.
Not	3 False Negative	4 True Negative	7 ∑ Pred. Neg.
	5 ∑ Actual Pos.	5 ∑ Actual Neg.	

True Positive Rate
TPR = 2/5 = 0.4

False Positive Rate
FPR = 1/5 = 0.2

Cost-Sensitive Confusion Matrix (Threshold = **0.3**)

Inst.	**Confidence** $\mathrm{p}(\hat{y})$	**Prediction** $\hat{y}$ $\mathrm{p}(\hat{y}) > 0.3$	**Actual** y
1	0.9	Alert	Alert
2	0.6	Alert	Alert
3	0.5	Alert	Alert
4	0.4	Alert	Alert
5	0.3	Not	Alert
6	0.2	Not	Not
7	0.3	Not	Not
8	0.4	Alert	Not
9	0.5	Alert	Not
10	0.55	Alert	Not

Actual Label

Predicted Label		Alert	Not	
	Alert	4 True Positive	3 False Positive	7 $\sum$ Pred. Pos.
	Not	1 False Negative	2 True Negative	3 $\sum$ Pred. Neg.
		5 $\sum$ Actual Pos.	5 $\sum$ Actual Neg.	

True Positive Rate
TPR = 4/5 = 0.8

False Positive Rate
FPR = 3/5 = 0.6

Cost-Sensitive Confusion Matrix (Threshold = **0.9**)

Inst.	Confidence $p(\hat{y})$	Prediction $\hat{y}$ $p(\hat{y}) > 0.6$	Actual y
1	0.9	Alert	Alert
2	0.6	Not	Alert
3	0.5	Not	Alert
4	0.4	Not	Alert
5	0.3	Not	Alert
6	0.2	Not	Not
7	0.3	Not	Not
8	0.4	Not	Not
9	0.5	Not	Not
10	0.55	Not	Not

Actual Label

Predicted Label	Alert	Not	
Alert	1 True Positive	0 False Positive	1 $\sum$ Pred. Pos.
Not	4 False Negative	5 True Negative	9 $\sum$ Pred. Neg.
	5 $\sum$ Actual Pos.	5 $\sum$ Actual Neg.	

True Positive Rate
TPR = 1/5 = 0.2

False Positive Rate
FPR = 0/5 = 0.0

Cost-Sensitive Confusion Matrix

...

p > 0.3

		Actual Label: Alert	Actual Label: ¬Alert	
Predicted Label	Alert	4 True Positive	3 False Positive	7 ∑ Pred. Pos.
Predicted Label	¬Alert	1 False Negative	2 True Negative	3 ∑ Pred. Neg.
		5 ∑ Actual Pos.	5 ∑ Actual Neg.	

True Positive Rate
TPR = 4/5 = 0.8

False Positive Rate
FPR = 3/5 = 0.6

...

p > 0.5

		Actual Label: Alert	Actual Label: ¬Alert	
Predicted Label	Alert	2 True Positive	1 False Positive	3 ∑ Pred. Pos.
Predicted Label	¬Alert	3 False Negative	4 True Negative	7 ∑ Pred. Neg.
		5 ∑ Actual Pos.	5 ∑ Actual Neg.	

True Positive Rate
TPR = 2/5 = 0.4

False Positive Rate
FPR = 1/5 = 0.2

...

p > 0.9

		Actual Label: Alert	Actual Label: ¬Alert	
Predicted Label	Alert	1 True Positive	0 False Positive	1 ∑ Pred. Pos.
Predicted Label	¬Alert	4 False Negative	5 True Negative	9 ∑ Pred. Neg.
		5 ∑ Actual Pos.	5 ∑ Actual Neg.	

True Positive Rate
TPR = 1/5 = 0.2

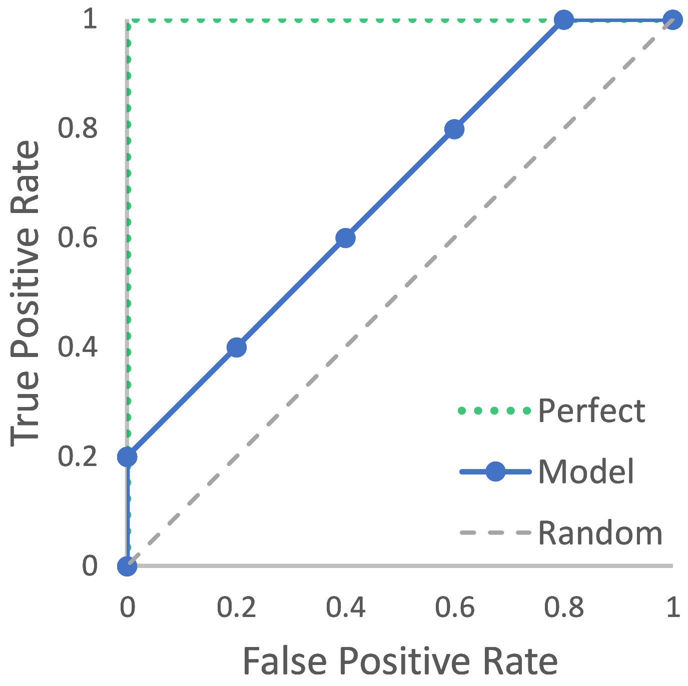
False Positive Rate
FPR = 0/5 = 0.0

...

Confusion matrix depends on **prediction threshold**

Receiver Operator Characteristic (ROC) Curve

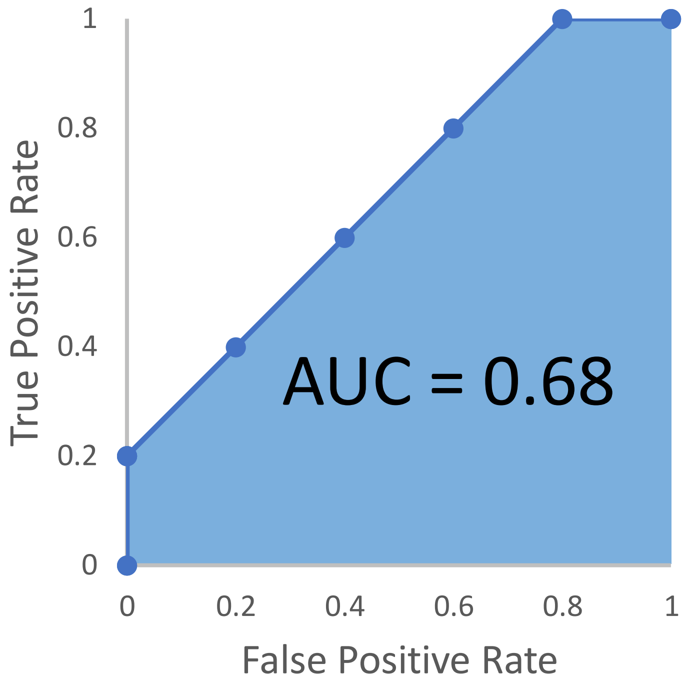
Threshold	TPR	FPR
0	1	1
0.1	1.0	1.0
0.2	1.0	0.8
0.3	0.8	0.6
0.4	0.6	0.4
0.5	0.4	0.2
0.6	0.2	0.0
0.7	0.2	0.0
0.8	0.2	0.0
0.9	0.0	0.0
1	0	0



- Diagonal random line indicates 50% chance of correctness.
- If **ROC curve** is above the random line, model is more accurate than chance.
- **Perfect curve** has TPR = 1 and FPR = 0 always.

Area Under Curve (AUC) of ROC

Threshold	TPR	FPR
0	1	1
0.1	1.0	1.0
0.2	1.0	0.8
0.3	0.8	0.6
0.4	0.6	0.4
0.5	0.4	0.2
0.6	0.2	0.0
0.7	0.2	0.0
0.8	0.2	0.0
0.9	0.0	0.0
1	0	0



- AUC is a **concise metric** instead of a full figure.
- Concise metrics enable *clearer comparisons*.
- **AUC > 0.5** means the model is better than chance.
- AUC ≈ 1 means model is very accurate.

Area Under Curve (AUC) of ROC (example)

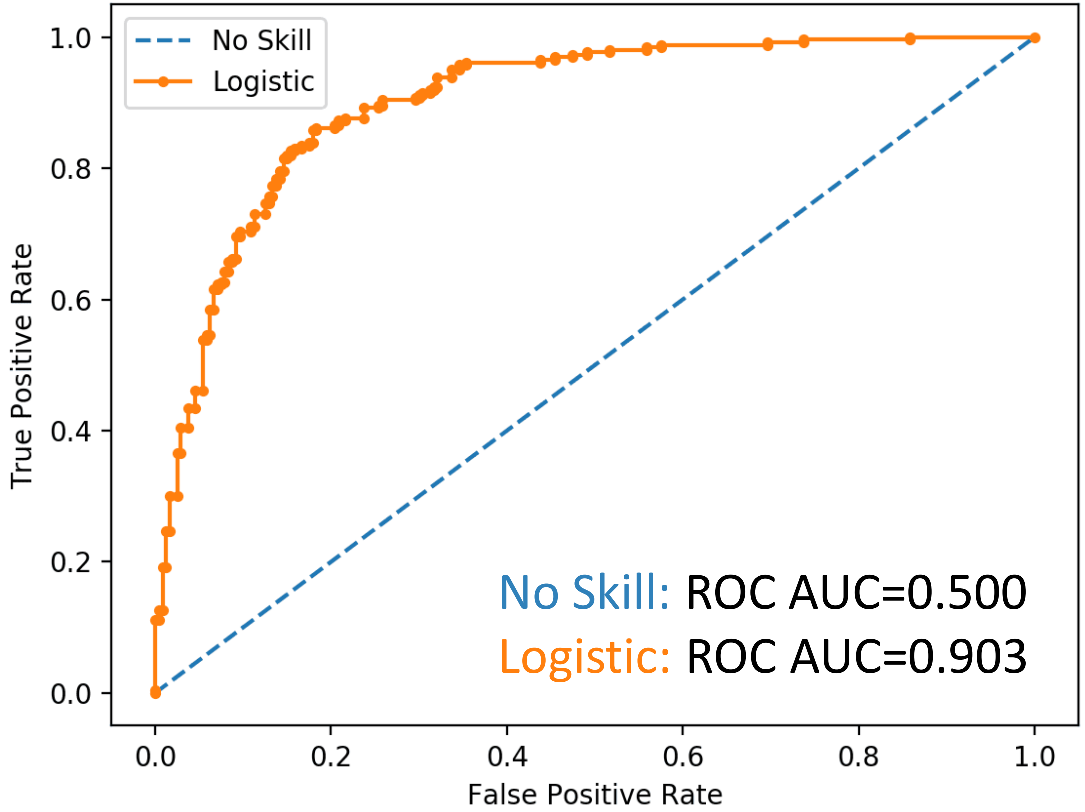


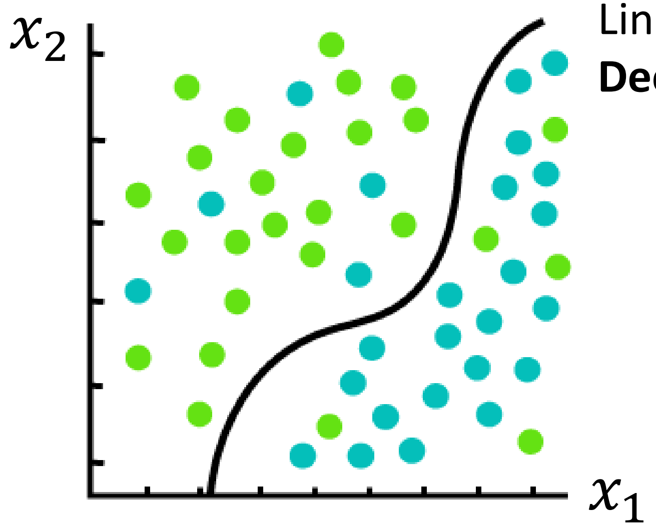
Image credit: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/


?
Questions!

Classification

$\boxed{\hat{y} \in \{0,1\} \text{ binary}}$

$\boxed{\hat{y} \in \{y_A, y_B, \dots\} \text{ multi-class}}$



Line is a
Decision Boundary

$$\hat{y} = M(\boldsymbol{x}), \quad \boldsymbol{x} = \vec{x} = (x_1, x_2)^\top$$

Regression

$\hat{y} \in \mathbb{R}$ any real number (scalar)

Line is a
Curve Fit

$$\hat{y} = M(x), \quad x = x_1$$

$$\hat{y} = M(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, \dots, x_n)^\top$$

Image credit:
https://www.javatpoint.com/regression-vs-classification-in-machine-learning

Confusion Matrix (binary classification)

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy

		Actual Label	
		Alert	Sleepy
Predicted Label	Alert	2	1
	Sleepy	3	4

Confusion Matrix (multiclass classification)

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy
11	Away	Away
12	Away	Alert
…	…	…

Actual Label

Predicted Label	Alert	Sleepy	Away
Alert	3	2	#
Sleepy	2	3	#
Away	#	#	#

Confusion Matrix (multiclass example)

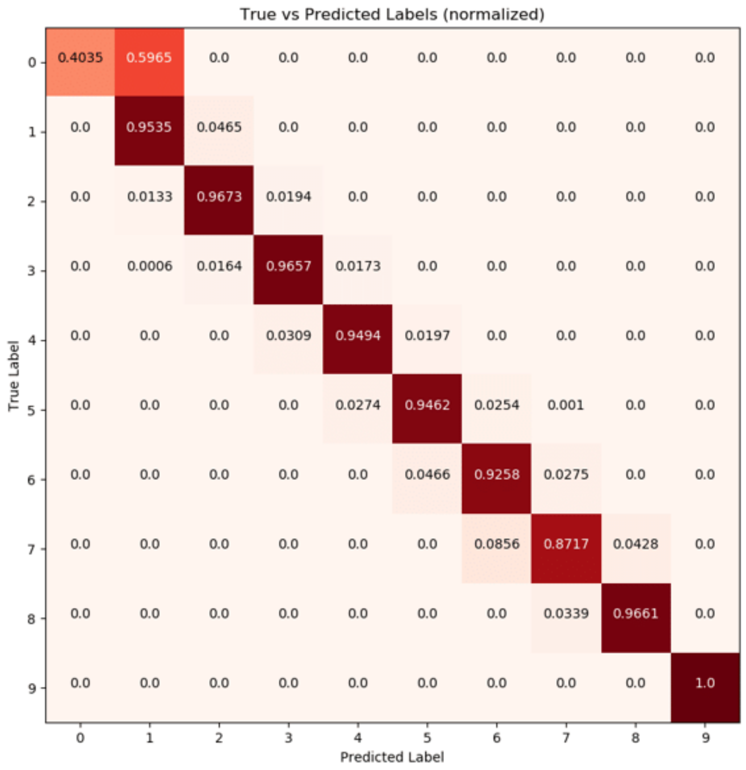


Image Credit

How to calculate:

- Accuracy
- Precision, Recall, F_1
- AUC?

Confusion Matrix (binary classification)

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy

Actual Label

Predicted Label	Alert	Sleepy
Alert	2 True Positive	1 False Positive
Sleepy	3 False Negative	4 True Negative

Confusion Matrix (multiclass classification)

Inst.	**Predicted $\hat{y}$**	**Actual y**
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy
11	Away	Away
12	Away	Alert
…	…	…

		Actual Label		
		Alert	Sleepy	Away
Predicted Label	Alert	2 ~~True Positive~~	1 ~~False Positive~~	#
	Sleepy	3 ~~False Negative~~	4 ~~True Negative~~	#
	Away	#	#	#

Which class is Positive? Negative?

Confusion Matrix (multiclass classification)

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy
11	Away	Away
12	Away	Alert
…	…	…

		Actual Label		
		Alert	Not	Not
Predicted Label	Alert	TP	FP	FP
	Not	FN	TN	TN
	Not	FN	TN	TN

Alert class is Positive, others Neg.

Confusion Matrix (multiclass classification)

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy
11	Away	Away
12	Away	Alert
…	…	…

		Actual Label		
		Not	Sleepy	Not
Predicted Label	Not	TN	FN	TN
	Sleepy	FP	TP	FP
	Not	TN	FN	TN

Sleepy class is Positive, others Neg.

Confusion Matrix (multiclass classification)

Inst.	Predicted $\hat{y}$	Actual y
1	Alert	Alert
2	Alert	Alert
3	Sleepy	Alert
4	Sleepy	Alert
5	Sleepy	Alert
6	Sleepy	Sleepy
7	Sleepy	Sleepy
8	Sleepy	Sleepy
9	Sleepy	Sleepy
10	Alert	Sleepy
11	Away	Away
12	Away	Alert
…	…	…

Actual Label

Predicted Label	Not	Not	Away
Not	TN	TN	FN
Not	TN	TN	FN
Away	FP	FP	TP

Away class is Positive, others Neg.

Multiclass evaluation metrics: **Average?**

		Actual Label		
		Alert	Not	Not
Predicted Label	Alert	TP_{c1}	FP_{c1}	FP_{c1}
	Not	FN_{c1}	TN_{c1}	TN_{c1}
	Not	FN_{c1}	TN_{c1}	TN_{c1}

		Actual Label		
		Not	Sleepy	Not
Predicted Label	Not	TN_{c2}	FN_{c2}	TN_{c2}
	Sleepy	FP_{c2}	TP_{c2}	FP_{c2}
	Not	TN_{c2}	FN_{c2}	TN_{c2}

		Actual Label		
		Not	Not	Away
Predicted Label	Not	TN_{c3}	TN_{c3}	FN_{c3}
	Not	TN_{c3}	TN_{c3}	FN_{c3}
	Away	FP_{c3}	FP_{c3}	TP_{c3}

How to combine?

Multiclass evaluation metrics: **Micro-Average**

Predicted Label \ Actual Label	Alert	Not	Not
Alert	TP_{c1}	FP_{c1}	FP_{c1}
Not	FN_{c1}	TN_{c1}	TN_{c1}
Not	FN_{c1}	TN_{c1}	TN_{c1}

Predicted Label \ Actual Label	Not	Sleepy	Not
Not	TN_{c2}	FN_{c2}	TN_{c2}
Sleepy	FP_{c2}	TP_{c2}	FP_{c2}
Not	TN_{c2}	FN_{c2}	TN_{c2}

Predicted Label \ Actual Label	Not	Not	Away
Not	TN_{c3}	TN_{c3}	FN_{c3}
Not	TN_{c3}	TN_{c3}	FN_{c3}
Away	FP_{c3}	FP_{c3}	TP_{c3}

Predicted Label \ Actual Label	Pos	Neg
Pos	TP_Σ	FP_Σ
Neg	FN_Σ	TN_Σ

$$TP_\Sigma = \frac{1}{|C|}\Sigma_{c\in C}\, TP_c$$
$$= (TP_{c1} + TP_{c2} + TP_{c3})/3$$
$$TN_\Sigma = \frac{1}{|C|}\Sigma_{c\in C}\, TN_c$$
$$FP_\Sigma = \frac{1}{|C|}\Sigma_{c\in C}\, FP_c$$
$$FN_\Sigma = \frac{1}{|C|}\Sigma_{c\in C}\, FN_c$$

$$\boxed{A_\Sigma, P_\Sigma, R_\Sigma, F1_\Sigma, AUC_\Sigma}$$

Multiclass evaluation metrics: **Macro-Average**

Predicted Label \ Actual Label	Alert	Not	Not
Alert	TP_{c1}	FP_{c1}	FP_{c1}
Not	FN_{c1}	TN_{c1}	TN_{c1}
Not	FN_{c1}	TN_{c1}	TN_{c1}

$$A_{c1}, P_{c1}, R_{c1}, F1_{c1}, AUC_{c1}$$

Predicted Label \ Actual Label	Not	Sleepy	Not
Not	TN_{c2}	FN_{c2}	TN_{c2}
Sleepy	FP_{c2}	TP_{c2}	FP_{c2}
Not	TN_{c2}	FN_{c2}	TN_{c2}

$$A_{c2}, P_{c2}, R_{c2}, F1_{c2}, AUC_{c2}$$

Predicted Label \ Actual Label	Not	Not	Away
Not	TN_{c3}	TN_{c3}	FN_{c3}
Not	TN_{c3}	TN_{c3}	FN_{c3}
Away	FP_{c3}	FP_{c3}	TP_{c3}

$$A_{c3}, P_{c3}, R_{c3}, F1_{c3}, AUC_{c3}$$

$$A_{\Sigma}, P_{\Sigma}, R_{\Sigma}, F1_{\Sigma}, AUC_{\Sigma}$$

$$A_{\Sigma} = \frac{1}{|C|}\sum_{c \in C} A_c \qquad P_{\Sigma} = \frac{1}{|C|}\sum_{c \in C} P_c \qquad F1_{\Sigma} = \frac{1}{|C|}\sum_{c \in C} F1_c$$

$$R_{\Sigma} = \frac{1}{|C|}\sum_{c \in C} R_c \qquad AUC_{\Sigma} = \frac{1}{|C|}\sum_{c \in C} AUC_c$$

Micro- vs. Macro Average

		Actual Label		
		Alert	Not	Not
Predicted Label	Alert	TP_{c1}	FP_{c1}	FP_{c1}
	Not	FN_{c1}	TN_{c1}	TN_{c1}
	Not	FN_{c1}	TN_{c1}	TN_{c1}

		Actual Label		
		Not	Sleepy	Not
Predicted Label	Not	TN_{c2}	FN_{c2}	TN_{c2}
	Sleepy	FP_{c2}	TP_{c2}	FP_{c2}
	Not	TN_{c2}	FN_{c2}	TN_{c2}

		Actual Label		
		Not	Not	Away
Predicted Label	Not	TN_{c3}	TN_{c3}	FN_{c3}
	Not	TN_{c3}	TN_{c3}	FN_{c3}
	Away	FP_{c3}	FP_{c3}	TP_{c3}

How to combine?

Micro-Average

Weighs each **instance** equally.
Accounts for imbalanced data.

Macro-Average

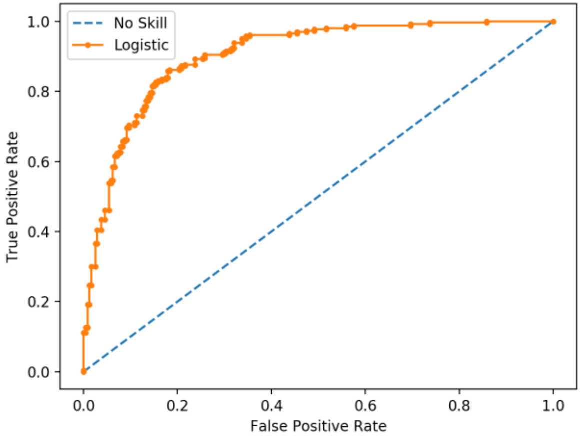
Weighs each **class** equally.
Use this if you think your real-world test data is balanced.

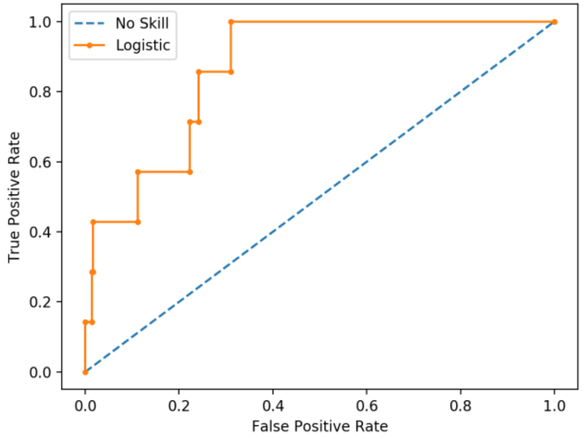
Imbalanced Classification evaluation with Precision-Recall (PR) Curve AUC

Further reading: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/

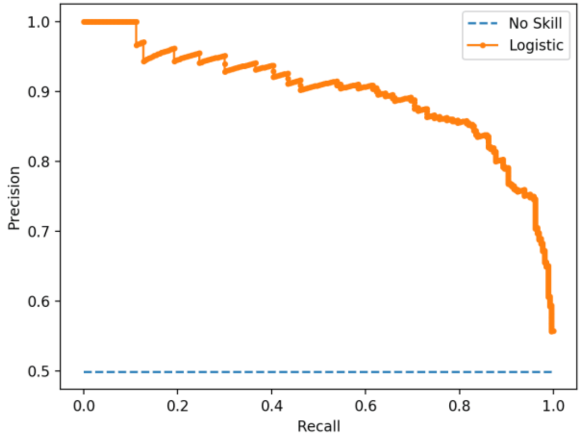
Balanced Classes | Imbalanced Classes

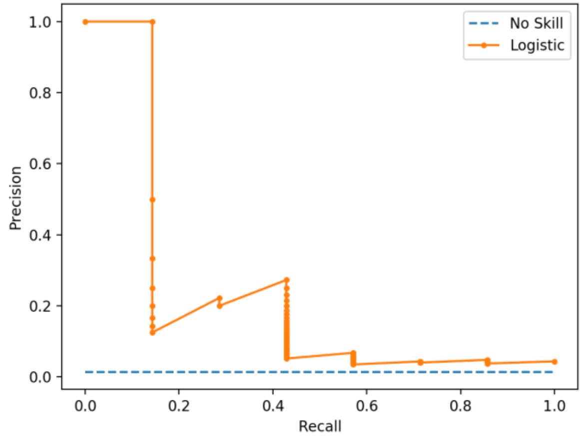
ROC Curve





PR Curve





Match appropriate **evaluation metric** to **challenge**

<table>
<tr><th>Challenge</th><th>Evaluation Metric</th></tr>
<tr><td>Imbalanced actual classes</td><td rowspan="3">1️⃣ Accuracy (Emote :one:)
2️⃣ Precision (:two:)
3️⃣ Recall (:three:)
4️⃣ F_1 Score (:four:)
5️⃣ ROC AUC (:five:)
6️⃣ PRC AUC (:six:)
7️⃣ Micro-Average (:seven:)
8️⃣ Macro-Average (:eight:)</td></tr>
<tr><td>Multiclass classification</td></tr>
<tr><td>Cost-dependent classes</td></tr>
</table>

Emote (react) in Slack #general channel <u>one or more options</u> (MRQ) for each challenge

Match appropriate **evaluation metric** to **challenge**

Challenge	**Evaluation Metric**
Imbalanced actual classes 2️⃣ 3️⃣ 6️⃣ 7️⃣	1️⃣ Accuracy (Emote :one:) 2️⃣ Precision (:two:) 3️⃣ Recall (:three:)
Multiclass classification 7️⃣ 8️⃣	4️⃣ F_1 Score (:four:) 5️⃣ ROC AUC (:five:) 6️⃣ PRC AUC (:six:)
Cost-dependent classes 2️⃣ 3️⃣ 5️⃣	7️⃣ Micro-Average (:seven:) 8️⃣ Macro-Average (:eight:)

Regression Evaluation Metrics

Department of Computer Science
School of Computing

Classification

$\hat{y} \in \{0,1\}$ binary

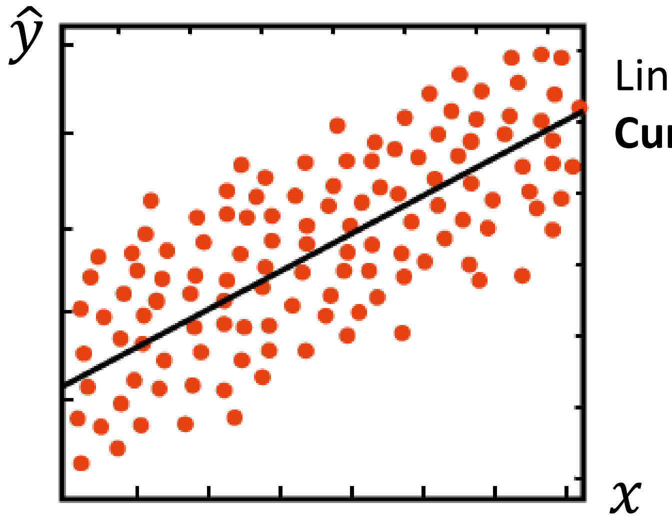
$\hat{y} \in \{y_A, y_B, \dots\}$ multi-class

Line is a
Decision Boundary

$$\hat{y} = M(\boldsymbol{x}), \quad \boldsymbol{x} = \vec{x} = (x_1, x_2)^\top$$

Regression

$\hat{y} \in \mathbb{R}$ any real number



Line is a
Curve Fit

$$\hat{y} = M(x), \quad x = x_1$$

$$\hat{y} = M(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, \dots, x_n)^\top$$

Image credit:
https://www.javatpoint.com/regression-vs-classification-in-machine-learning

Week 07b: Lecture Outline

1. Recap: Classification vs. Regression
2. Classification Metrics
3. Regression Metrics
    1. 1D regression: MSE, MAE
    2. Vector regression: Euclidean distance, Angular distance / Cosine Similarity
    3. Complex metrics for unstructured data

Note: intuition is opposite to “correctness”.

- Longer distance means worse performance
- Smaller distance is better performance

Average difference metrics **for test dataset**

Mean Absolute Error (MAE)

$$MAE = \frac{1}{m}\sum_{j=1}^{m}\left|\hat{y}_j - y_j\right|$$

Mean Squared Error (MSE)

$$MSE = \frac{1}{m}\sum_{j=1}^{m}\left(\hat{y}_j - y_j\right)^2$$

Root Mean Squared Error (RMSE)

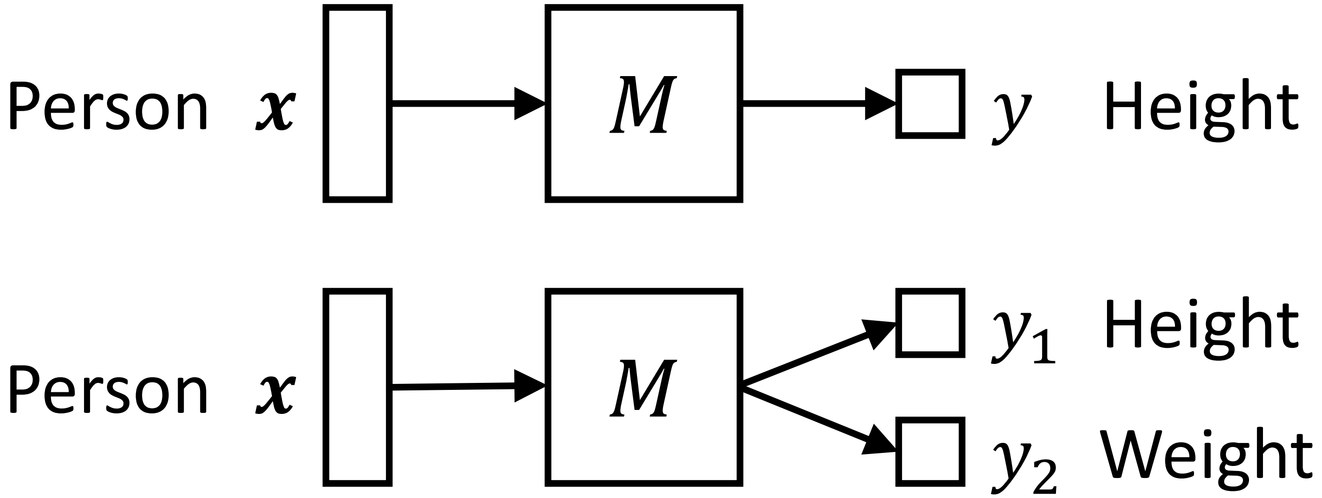
$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\hat{y}_j - y_j\right)^2}$$

MSE and RMSE penalize larger differences **more** than MAE

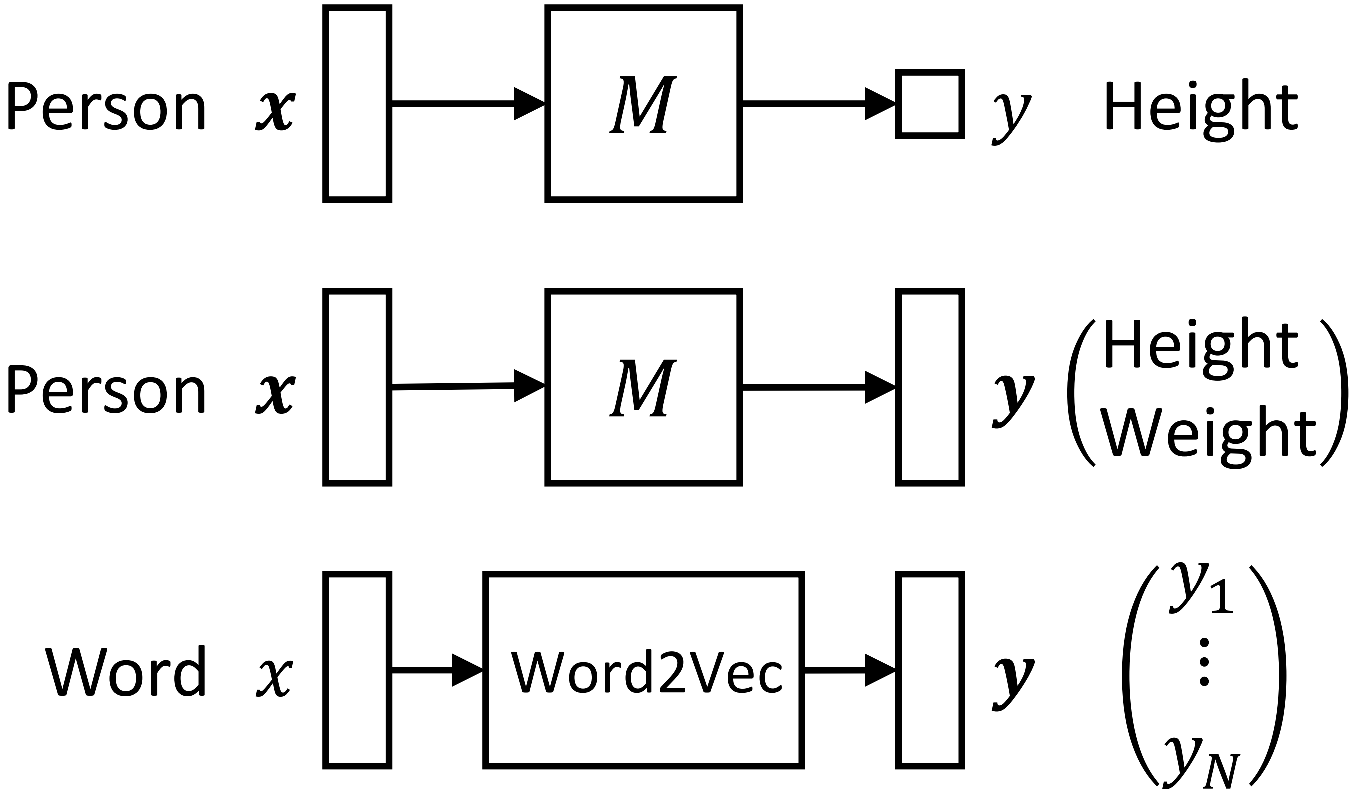
Week 07b: Lecture Outline

1. Recap: Classification vs. Regression
2. Classification Metrics
3. Regression Metrics
    1. 1D regression: MSE, MAE
    2. Vector regression: Euclidean distance, Angular distance / Cosine Similarity
    3. Complex metrics for unstructured data

Multi-task prediction



Multi-task prediction: predicting a vector $\boldsymbol{y}$

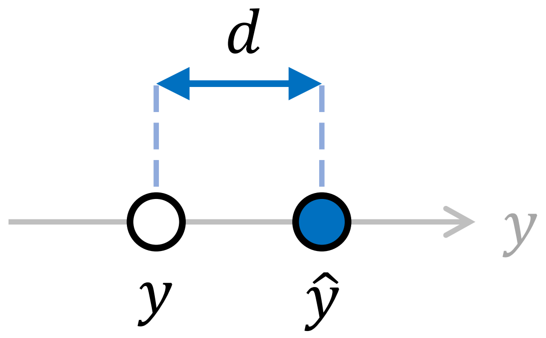


https://www.tensorflow.org/tutorials/text/word2vec

$\boldsymbol{y}$ is a vector representation of the text. Also known as "**embedding**"

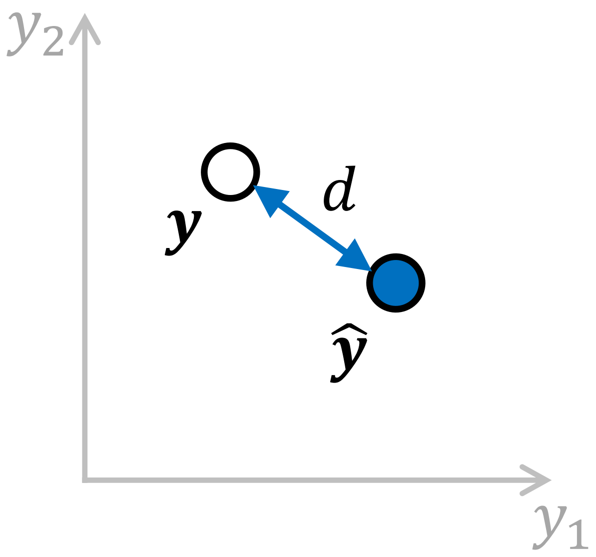
Vector Distances and Similarity

Cosine similarity is often used for text embeddings, since their vectors are unit length, i.e., $\|\hat{\boldsymbol{y}}\|_2 \approx \|\boldsymbol{y}\|_2 \approx 1$



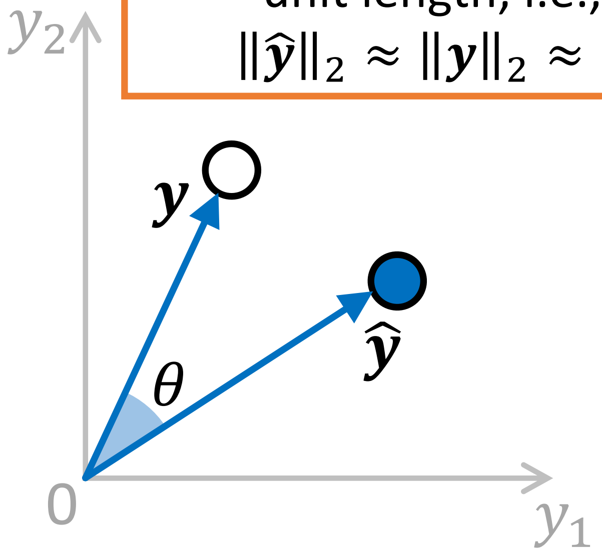
Squared Distance

$$d = (\hat{y} - y)^2$$



Euclidean Distance

$$d = \sqrt{\underbrace{(\hat{\boldsymbol{y}} - \boldsymbol{y})^\top (\hat{\boldsymbol{y}} - \boldsymbol{y})}_{\text{Dot Product}}}$$



Cosine Similarity

$$s = \cos(\theta) = \frac{\hat{\boldsymbol{y}}}{\|\hat{\boldsymbol{y}}\|_2} \cdot \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_2}$$

Angular Distance

$$\theta = \cos^{-1}(s)$$

Week 07b: Lecture Outline

1. Recap: Classification vs. Regression
2. Classification Metrics
3. Regression Metrics
    1. 1D regression: MSE, MAE
    2. Vector regression: Euclidean distance, Angular distance / Cosine Similarity
    3. Complex metrics for unstructured data

Advanced Evaluation Metrics for Images, Time Series, Unstructured Data (with Deep Learning)

1. Similarity between (probability) distributions
    1. Kullback-Leibler Divergence
    2. Jensen-Shannon Distance
2. Similarity between images
    1. Mean Squared Error
    2. Peak Signal-to-Noise Ratio (PSNR)
    3. Structural Similarity Index Measure (SSIM)
    4. Pearson Correlation Coefficient
3. Segmentation (region) overlap
    1. Jaccard Index / Intersection-over-Union (IoU)

Won't be in the exam!

Wrapping Up

What did we learn for Evaluation?

1. Classification vs. Regression
2. Classification Metrics
    1. Accuracy
    2. Confusion Matrix, TP, TN, FP, FN
    3. Precision, Recall, F_1
    4. ROC, AUC
    5. Micro- and Macro-Averaging
    6. PR-AUC (Average Precision)
3. Regression Metrics
    1. 1D regression: MSE, MAE
    2. Vector regression: Euclidean distance, Angular distance / Cosine Similarity

Appropriate **evaluation metric** depends on <u>prediction task</u> and <u>data issues</u>.

Next week: Data Preparation for ML

Image credit:
https://img2.thejournal.ie/article/5047666/river?version=5047733&width=1340

W08 Pre-Lecture Task (due before next Mon)

Read

1. Discover Feature Engineering, How to Engineer Features and How to Get Good at It by Jason Brownlee
2. 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset by Jason Brownlee

Task

1. Identify cases of **bad data** in machine learning
2. Propose **mitigation strategies**

   Tip: you can your own projects too; you don't have to be correct
3. Post a 1–2 sentence answer to the topic in your tutorial group: #tg-xx