

Classifiers and their metrics

Hicham Janati · INSEA · hjanati@insea.ac.ma

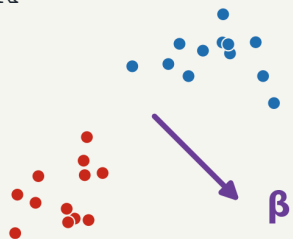
Classification

From scores to classes

New target: a **label** $y \in \{0, 1\}$. Our models output a **score** $z \stackrel{\text{def}}{=} f_{\beta}(\mathbf{x}) \in \mathbb{R}$. What is a good f_{β} ?

Example, a linear score: $z_i = f_{\beta}(\mathbf{x}_i) = \beta^{\top} \mathbf{x}_i$

$\mathbf{x}_i \in \mathbb{R}^2$



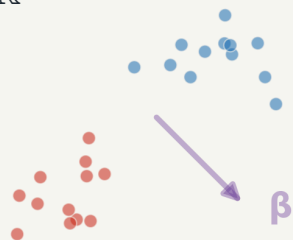
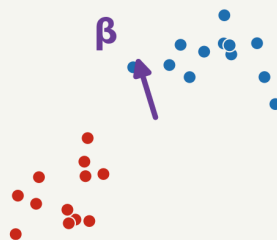
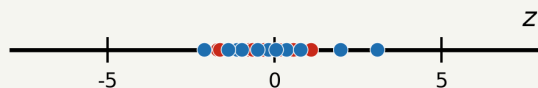
What are the z_i here ?

From scores to classes

New target: a **label** $y \in \{0, 1\}$. Our models output a **score** $z \stackrel{\text{def}}{=} f_{\beta}(\mathbf{x}) \in \mathbb{R}$. What is a good f_{β} ?

Example, a linear score: $z_i = f_{\beta}(\mathbf{x}_i) = \beta^{\top} \mathbf{x}_i$

$\mathbf{x}_i \in \mathbb{R}^2$



random

What are the z_i here ?

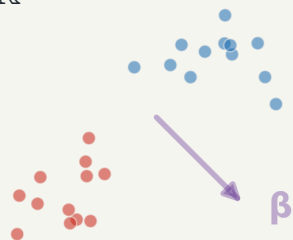
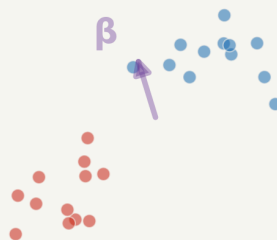
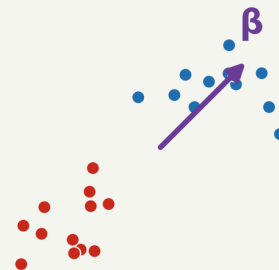
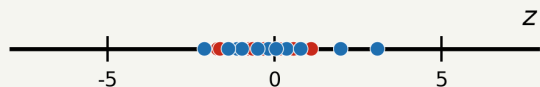
Is this a good f_{β} ?

From scores to classes

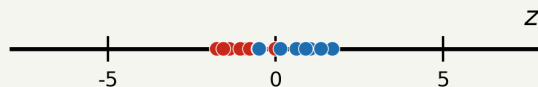
New target: a **label** $y \in \{0, 1\}$. Our models output a **score** $z \stackrel{\text{def}}{=} f_{\beta}(\mathbf{x}) \in \mathbb{R}$. What is a good f_{β} ?

Example, a linear score: $z_i = f_{\beta}(\mathbf{x}_i) = \beta^{\top} \mathbf{x}_i$

$\mathbf{x}_i \in \mathbb{R}^2$



random



good but uncertain

What are the z_i here ?

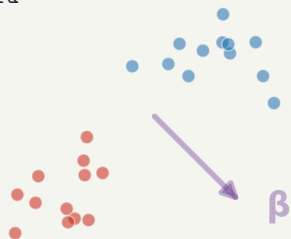
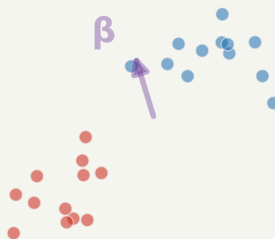
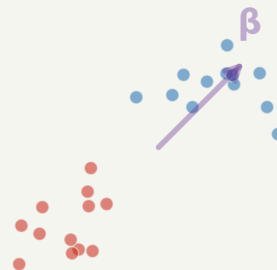
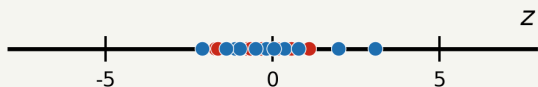
Is this a good f_{β} ?

From scores to classes

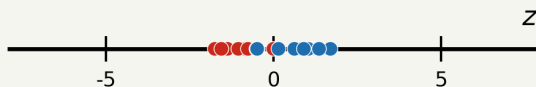
New target: a **label** $y \in \{0, 1\}$. Our models output a **score** $z \stackrel{\text{def}}{=} f_{\beta}(\mathbf{x}) \in \mathbb{R}$. What is a good f_{β} ?

Example, a linear score: $z_i = f_{\beta}(\mathbf{x}_i) = \beta^{\top} \mathbf{x}_i$

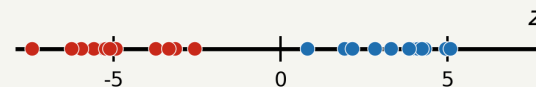
$\mathbf{x}_i \in \mathbb{R}^2$



random



good but uncertain



perfect

What are the z_i here ?

Is this a good f_{β} ?

What a good score does

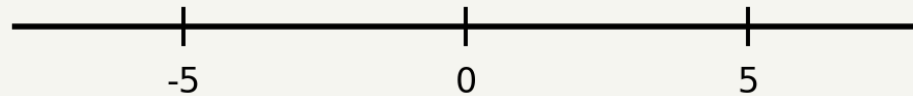
$$\mathbf{x}_i \in \mathbb{R}^d$$



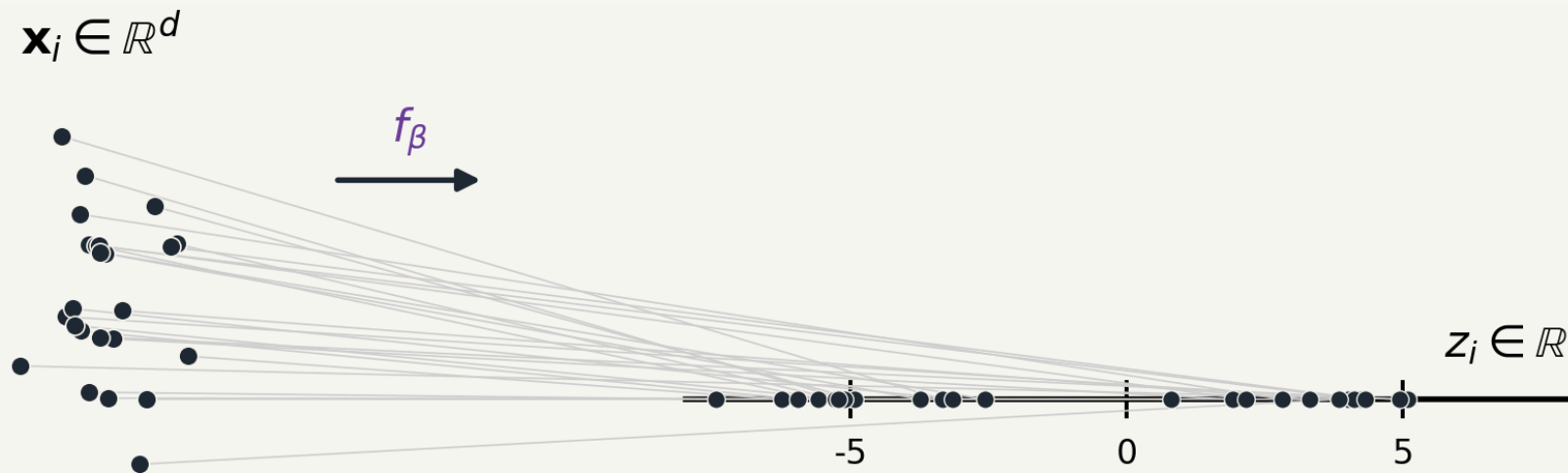
$$f_\beta$$



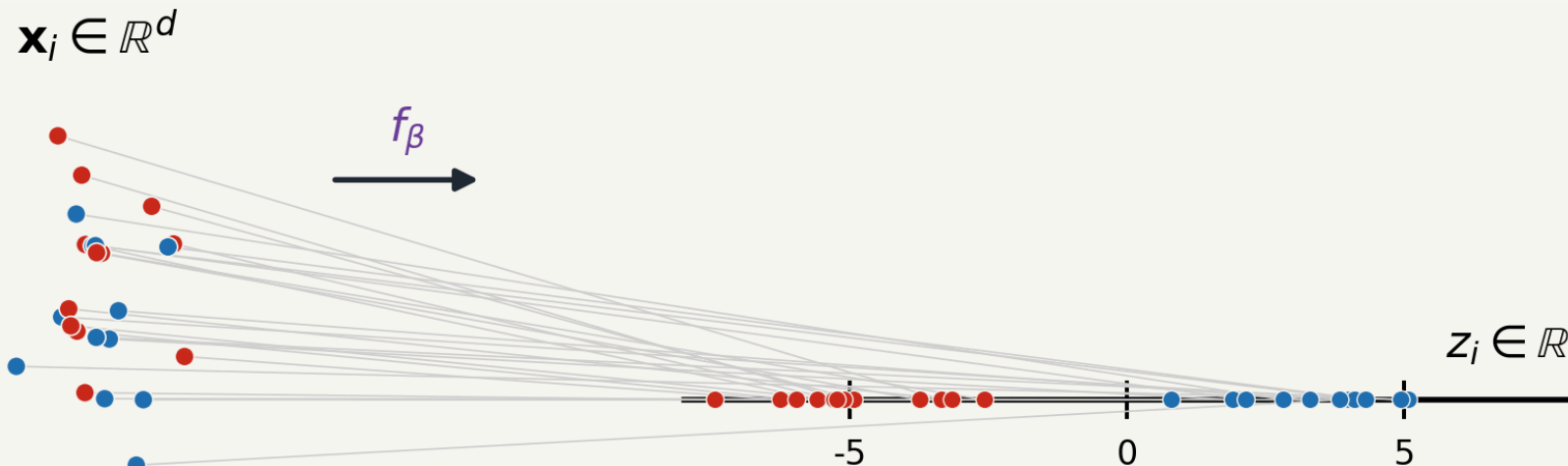
$$z_i \in \mathbb{R}$$



What a good score does



What a good score does



a good f_β : large **positive** $z \mapsto y = 1$, large **negative** $z \mapsto y = 0$

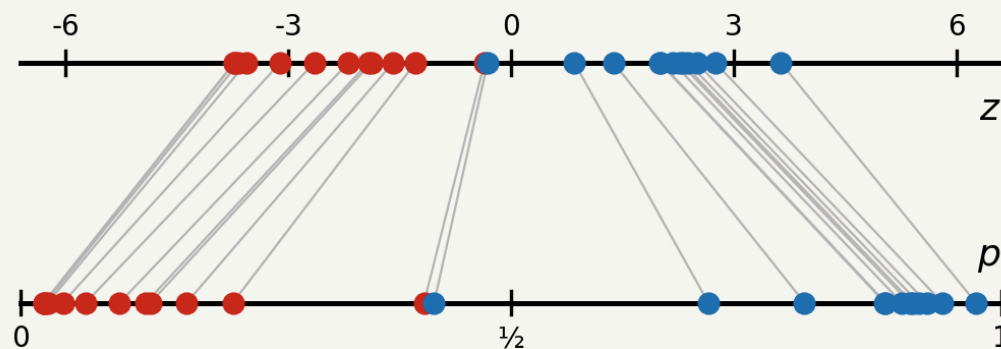
$|z|$ reads as a **confidence**: far from 0, the model is sure; near 0, it hesitates.

Can we turn a score into a **probability** ?

Mapping the scores to $[0, 1]$

To compare a score z_i with the labels $\{0, 1\}$, it would be nice to map it to $[0, 1]$, keeping the order:

$$z_i \in \mathbb{R} \longmapsto p_i \in [0, 1]$$

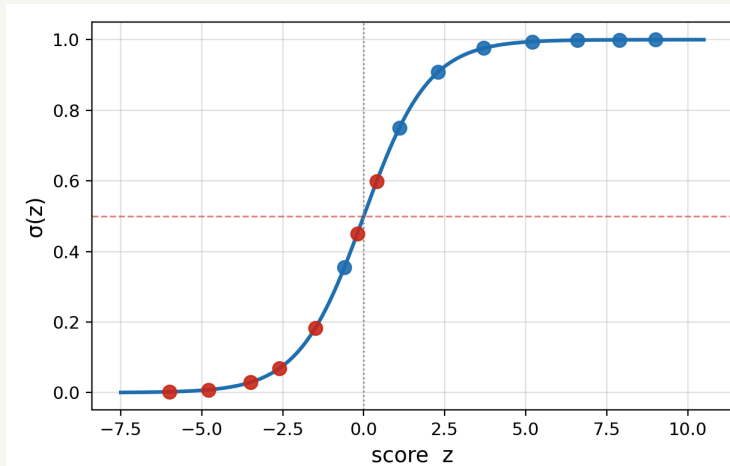


We can read p_i as a probability $\widehat{\mathbb{P}}(Y = 1 \mid \mathbf{x}_i)$.

Which map sends $\mathbb{R}$ onto $(0, 1)$, increasing, with $0 \mapsto \frac{1}{2}$?

From scores to probabilities

The **sigmoid** $\sigma(z) = \frac{1}{1 + e^{-z}}$



$$z \rightarrow -\infty: \sigma(z) \rightarrow 0$$

$$z = 0: \sigma(z) = \frac{1}{2}$$

$$z \rightarrow +\infty: \sigma(z) \rightarrow 1$$

$$\hat{p} = \sigma(z) \in (0, 1)$$

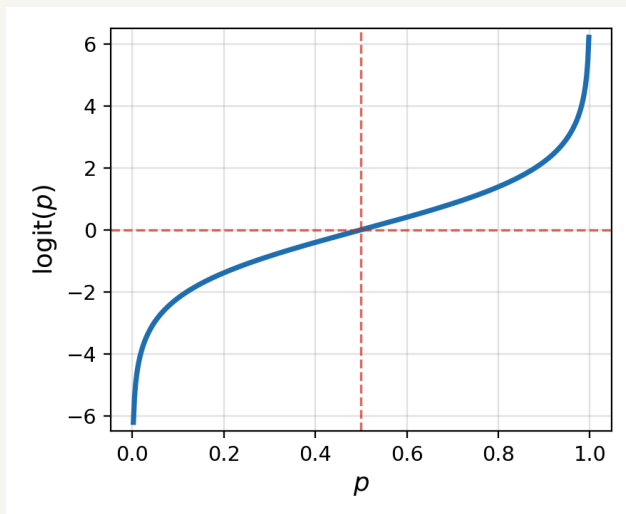
viewed as a probability $\hat{\mathbb{P}}(Y = 1 \mid \mathbf{x})$

To make a prediction, we threshold $\hat{p}$:

$$\hat{y} = 1 \text{ if } \hat{p} > \frac{1}{2}, \quad \hat{y} = 0 \text{ otherwise}$$

The logit

The sigmoid maps score $\rightarrow$ probability. Can we go **back** from $\hat{p}$ to the score z ?



$$z = \sigma^{-1}(\hat{p}) = \log \frac{\hat{p}}{1 - \hat{p}}$$

the **logit function**: the log of the **odds** $\frac{\hat{p}}{1 - \hat{p}}$ of class 1

The pre-sigmoid scores z_i are called the **logits**.

The log loss

We must compare $\hat{p} = \sigma(\beta^\top \mathbf{x})$ to the true label y .

true label	prediction	loss should be
$y = 1$	$\hat{p} \rightarrow 1$	small $\rightarrow 0$
$y = 0$	$\hat{p} \rightarrow 0$	large $\rightarrow +\infty$

Can you think of such a function $\hat{p} \mapsto \dots$?

$$-\log \hat{p}$$

symmetrically, if $y = 0$: $\text{cost}(\hat{p}) = -\log(1 - \hat{p})$

merge the two cases:

$$L(y, \hat{p}) = -[y \log \hat{p} + (1 - y) \log(1 - \hat{p})]$$

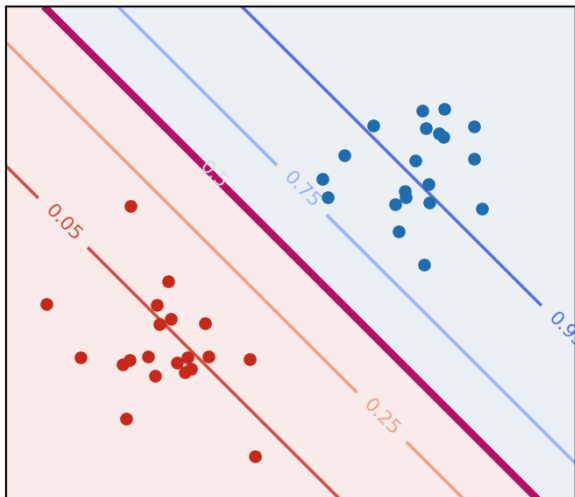
This is called the **log loss**.

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \frac{1}{n} \sum_i -[y_i \log \sigma(\beta^\top \mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\beta^\top \mathbf{x}_i))]$$

Even though this is a classification problem, this model is called **logistic regression**:

The decision boundary

The fitted prediction function: $\hat{p}(\mathbf{x}) = \sigma(\hat{\beta}^\top \mathbf{x})$



What do the level sets $\{\mathbf{x} : \hat{p}(\mathbf{x}) = c\}$ look like ?

level sets $\{\mathbf{x} : \hat{p}(\mathbf{x}) = c\}$: **parallel lines**, $\hat{\beta}^\top \mathbf{x} = \sigma^{-1}(c)$

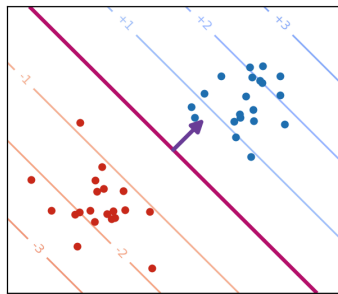
Where is $\hat{p} = \frac{1}{2}$?

It's a hyperplane. It's called the *decision boundary*

Logistic regression is a **linear classifier**

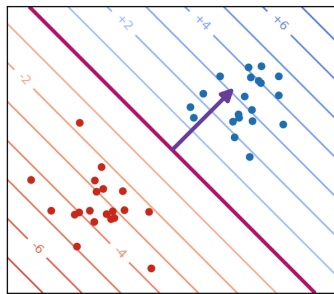
Same flags, smaller loss ?

level sets of
the scores
 $z = \beta^\top \mathbf{x}$



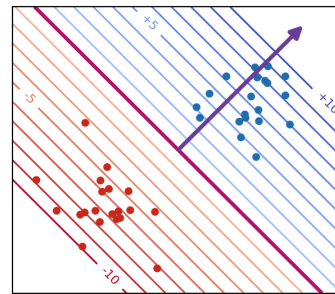
$$\|\beta\| = 1$$

average loss 0.200



$$\|\beta\| = 2$$

average loss 0.059



$$\|\beta\| = 4$$

average loss 0.007

How good is this separation ? → perfect: accuracy 100 %, every point on its side of $\beta^\top \mathbf{x} = 0$.

Larger $\|\beta\|$: what changed ? → same boundary and **predictions**; only the **confidence** grows.

And the average loss ? → $\frac{1}{n} \sum_i L(y_i, \sigma(\beta^\top \mathbf{x}_i))$ keeps falling toward 0.

Is that a problem ?

Yes! Optimization would never stop, driving
 $\|\beta\| \rightarrow +\infty$.

How do we fix it ?

Penalize the size of β

Ridge (chapter 5)

$$\min_{\beta} \underbrace{\frac{1}{n} \sum_{i=1}^n L(y_i, \sigma(\beta^\top \mathbf{x}_i))}_{\text{data fitting}} + \underbrace{\lambda \|\beta\|_2^2}_{\text{regularization}}, \quad \lambda > 0$$

→ a large $\|\beta\|$ now **costs**: a unique, finite minimizer.

scikit-learn

LogisticRegression(C=1.0)

$$\min_{\beta} C \sum_{i=1}^n L(y_i, \sigma(\beta^\top \mathbf{x}_i)) + \frac{1}{2} \|\beta\|_2^2$$

→ the same minimizer for $\lambda = \frac{1}{2nC}$: a **large** C is a **weak** penalty.

Lasso: `l1_ratio=1`

$\|\beta\|_1$ in place of $\frac{1}{2} \|\beta\|_2^2$: coefficients set **exactly to zero**.

Elastic net: `0 < l1_ratio < 1`

a mix of both penalties (with `solver="saga"`, as for the lasso).

Evaluating a classifier

The fraud desk

Seven transactions of the desk's **test file**, scored by its logistic model:

id	amount (Dh)	hour	$\hat{p}$	flagged	fraud	outcome
74083	899	04:26	0.55	yes	0	False alarm
81371	774	05:15	0.90	yes	1	Correct
100024	345	02:43	0.12	no	0	Correct
107258	1135	00:04	0.62	yes	1	Correct
109218	879	17:30	0.04	no	0	Correct
112468	825	04:59	0.72	yes	0	False alarm
118566	218	05:42	0.32	no	1	Undetected

The bank calls the card holder when the model flags a transaction: $\hat{p} > \frac{1}{2}$. Is this model **good** ?

Put each flag next to the truth: analyze the type of errors.

The confusion matrix

1 000 transactions, flagged when $\hat{p} > \frac{1}{2}$

TP, true positive: a fraud flagged

FP, false positive: a **false alarm**

FN, false negative: a **missed** fraud

TN, true negative: legit not flagged

True ↓ / Predicted →	legit	fraud
legit	TN = 984	FP = 4
fraud	FN = 6	TP = 6

Using the confusion matrix, what is the model's accuracy ? Is it reliable ?

$$\text{accuracy} = \frac{\# \text{ correct predictions}}{\# \text{ total}} = \frac{\text{TP} + \text{TN}}{n} = \frac{6 + 984}{1\,000} = 0.990$$

$$\text{error} = 1 - \text{accuracy} = 0.01$$

Would you trust this model when it says “**Fraud**” ?

Precision and recall

True ↓ / Predicted →	legit	fraud
legit	TN = 984	FP = 4
fraud	FN = 6	TP = 6

When the model **flags**, how often is it a real fraud ?

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{6}{6 + 4} = 0.60$$

Of the real frauds, how many does the model **flags** ?

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{6}{6 + 6} = 0.50$$

One number for precision and recall ?

One number from precision P and recall R ? First idea: the plain **average** $\frac{P + R}{2}$.

$$\text{our 1 000 transactions, } t = \frac{1}{2}: \quad \frac{0.60 + 0.50}{2} = 0.55$$

Its limitation ? Try a model that flags *almost* all transactions.

$$P \approx 0.012 \text{ (the fraud rate), } R = 1: \quad \frac{0.012 + 1}{2} \approx 0.51, \text{ half marks for a model that does **nothing**}$$

Which mean stays **small** as soon as **one** of P and R is small ?

The F1 score

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R} \quad \text{the **harmonic** mean}$$

F_1 is high only when the model has **both** precision and recall.

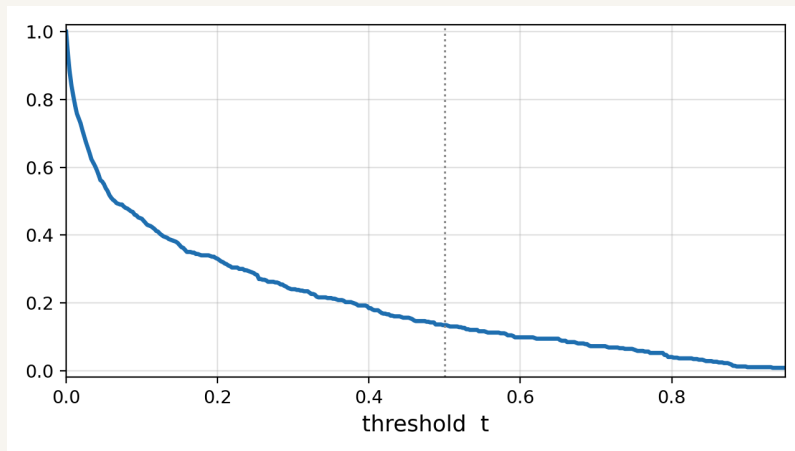


But P , R and F_1 all depend on the **threshold** t : why $t = \frac{1}{2}$ if another threshold gives better scores ?

Moving the threshold

$$\hat{y} = 1 \text{ if } \hat{p} > t, \quad \hat{y} = 0 \text{ otherwise}$$

Lower the threshold t : the desk flags **more**. What happens to recall ? And to precision ?

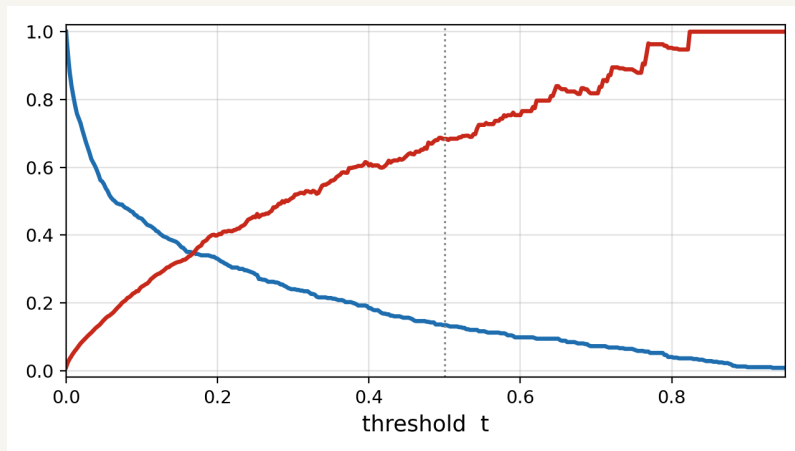


recall: falls as t grows

Moving the threshold

$$\hat{y} = 1 \text{ if } \hat{p} > t, \quad \hat{y} = 0 \text{ otherwise}$$

Lower the threshold t : the desk flags **more**. What happens to recall ? And to precision ?



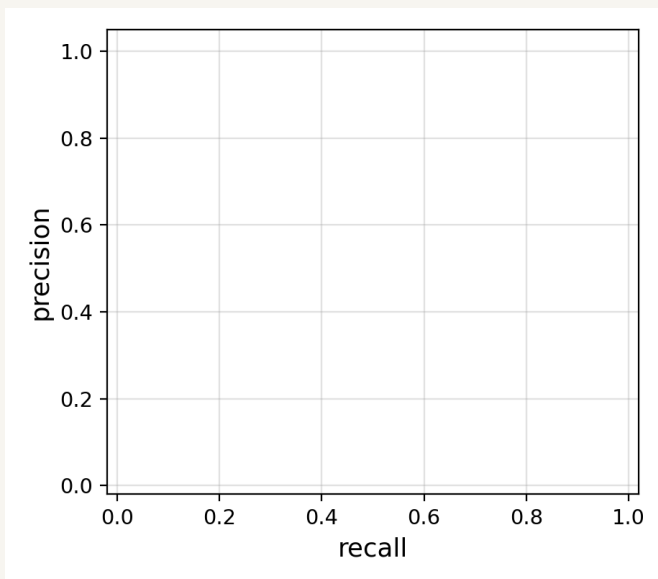
recall: falls as t grows

precision: rises as t grows

A **trade-off**: lowering t buys recall with precision.

Comparing models across thresholds

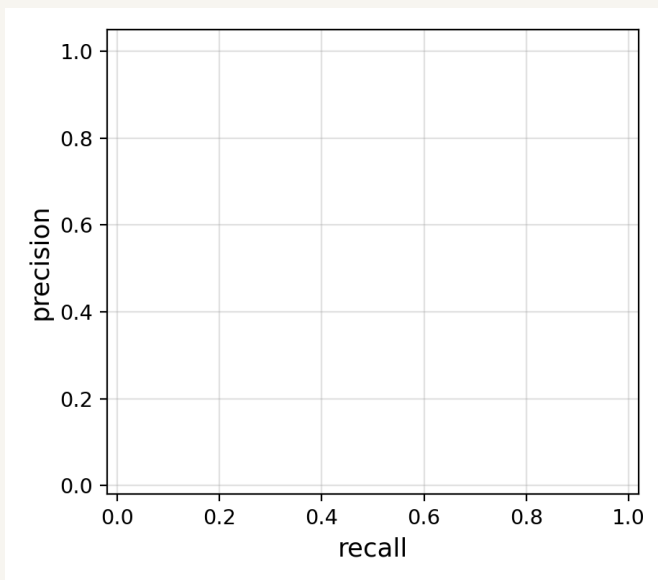
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
$t = 1?$		

Comparing models across thresholds

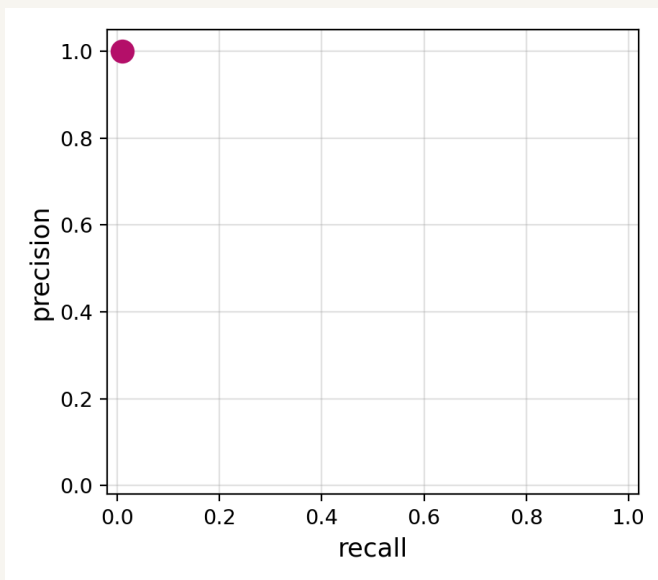
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
$t = 1?$	0	nothing flagged
$t = 0.9?$		

Comparing models across thresholds

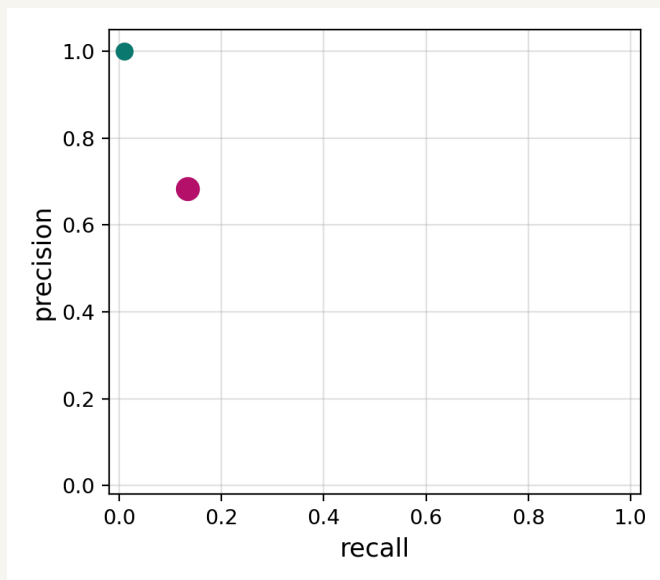
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
t = 1 ?	0	nothing flagged
t = 0.9 ?	0.01	1.00
t = 0.5 ?		

Comparing models across thresholds

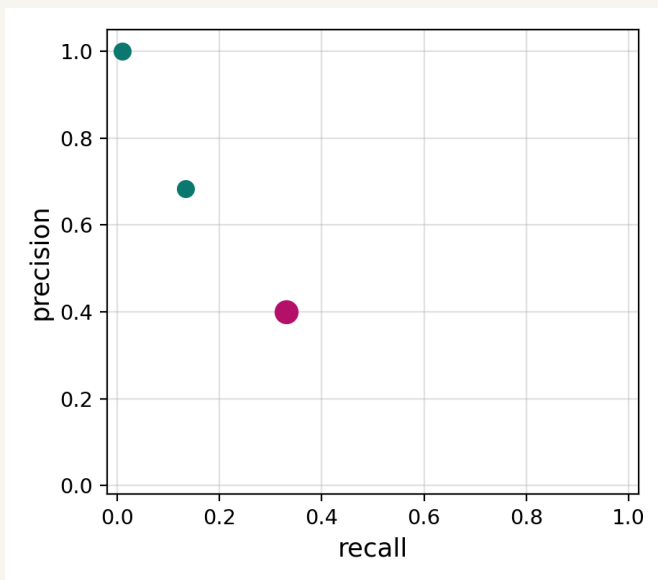
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
$t = 1$?	0	nothing flagged
$t = 0.9$?	0.01	1.00
$t = 0.5$?	0.13	0.68
$t = 0.2$?		

Comparing models across thresholds

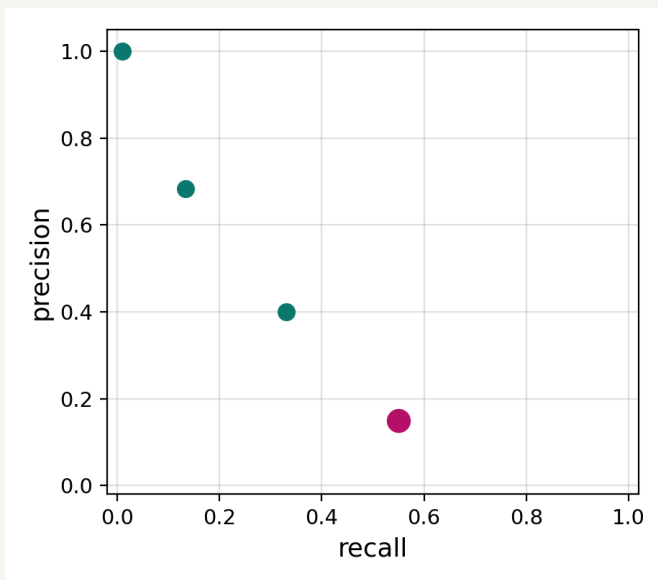
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
$t = 1$?	0	nothing flagged
$t = 0.9$?	0.01	1.00
$t = 0.5$?	0.13	0.68
$t = 0.2$?	0.33	0.40
$t = 0.05$?		

Comparing models across thresholds

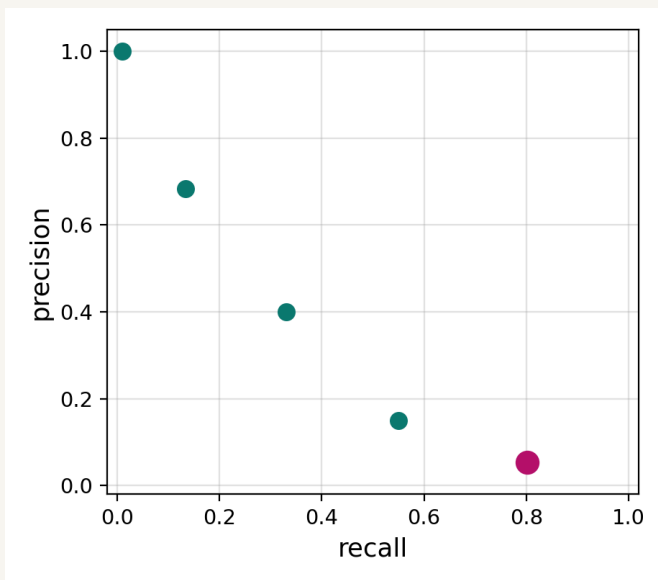
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
$t = 1$?	0	nothing flagged
$t = 0.9$?	0.01	1.00
$t = 0.5$?	0.13	0.68
$t = 0.2$?	0.33	0.40
$t = 0.05$?	0.55	0.15
$t = 0.01$?		

Comparing models across thresholds

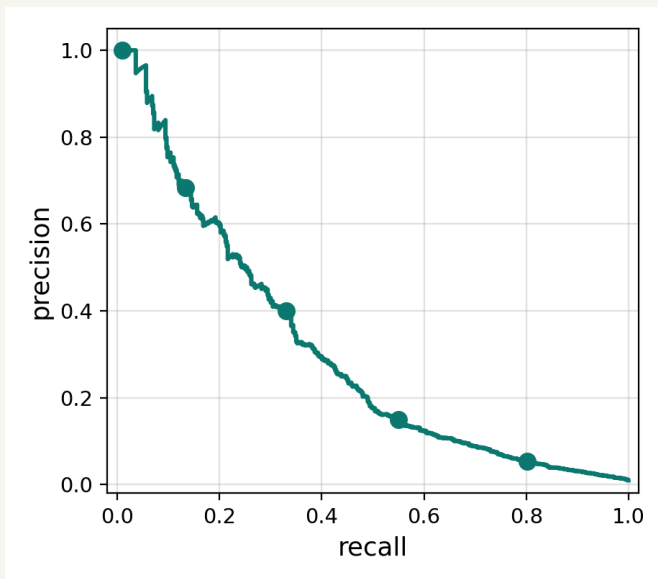
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
$t = 1 ?$	0	nothing flagged
$t = 0.9 ?$	0.01	1.00
$t = 0.5 ?$	0.13	0.68
$t = 0.2 ?$	0.33	0.40
$t = 0.05 ?$	0.55	0.15
$t = 0.01 ?$	0.80	0.05

Comparing models across thresholds

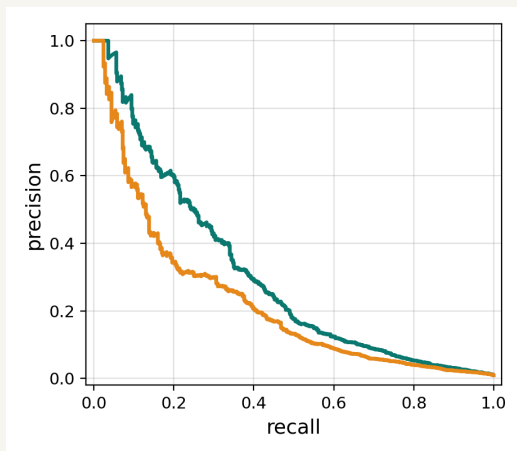
To compare models across **all** thresholds: sweep t , and track **recall** and **precision** together.



threshold	recall	precision
$t = 1$?	0	nothing flagged
$t = 0.9$?	0.01	1.00
$t = 0.5$?	0.13	0.68
$t = 0.2$?	0.33	0.40
$t = 0.05$?	0.55	0.15
$t = 0.01$?	0.80	0.05

Each t is one point; all of them together: the **precision-recall curve**.

The area under the precision-recall curve



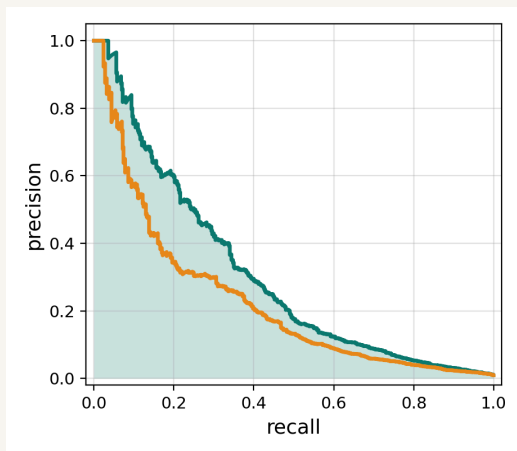
— model **A**

— model **B**

compare the **areas** under the curves: the **PR-AUC**
(average precision)

Two models, two curves: which one is **better** ?

The area under the precision-recall curve



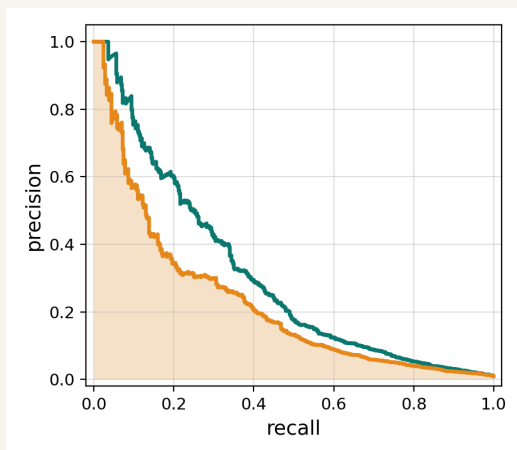
— model **A**: **PR-AUC = 0.31**

— model **B**

compare the **areas** under the curves: the **PR-AUC**
(average precision)

Two models, two curves: which one is **better** ?

The area under the precision-recall curve



— model A: **PR-AUC = 0.31**

— model B: **PR-AUC = 0.23**

compare the **areas** under the curves: the **PR-AUC**
(average precision)

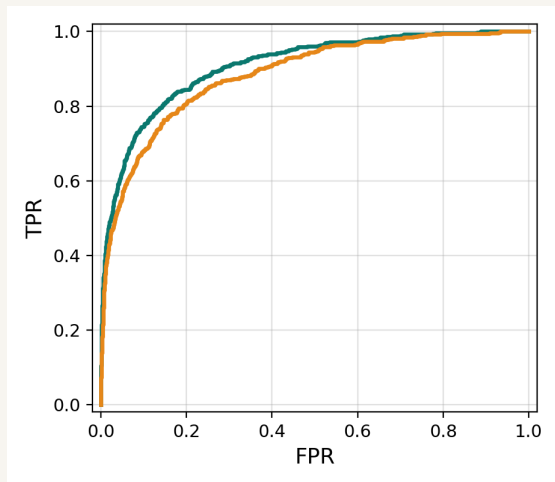
Two models, two curves: which one is **better** ?

The larger area is better **across all** thresholds: here, model A.

The PR-AUC value is however not very intuitive: **another area** is often used.

Another area: the ROC curve

Same sweep of t , two other rates:



$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{the share of frauds flagged (the recall)}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad \text{the share of legit transactions flagged}$$

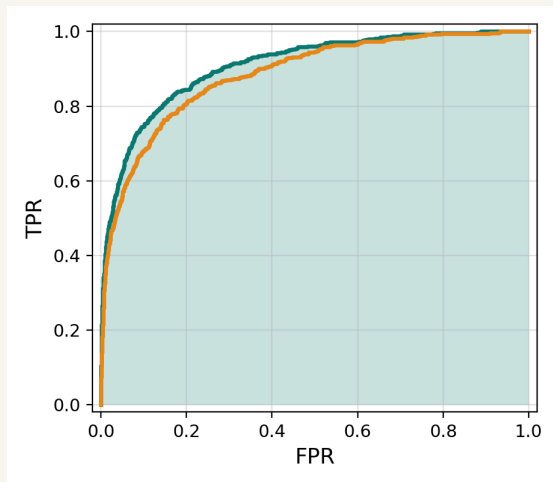
— model **A**

— model **B**

Which model is **better** ?

Another area: the ROC curve

Same sweep of t , two other rates:



$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{the share of frauds flagged (the recall)}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad \text{the share of legit transactions flagged}$$

— model **A**: **ROC-AUC = 0.91**

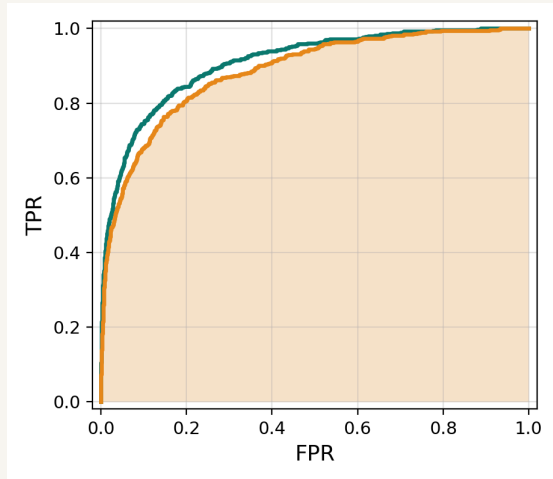
— model **B**

the area under each curve: the **ROC-AUC**

Which model is **better** ?

Another area: the ROC curve

Same sweep of t , two other rates:



$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{the share of frauds flagged (the recall)}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad \text{the share of legit transactions flagged}$$

— model **A**: **ROC-AUC = 0.91**

— model **B**: **ROC-AUC = 0.89**

the area under each curve: the **ROC-AUC**

Which model is **better** ?

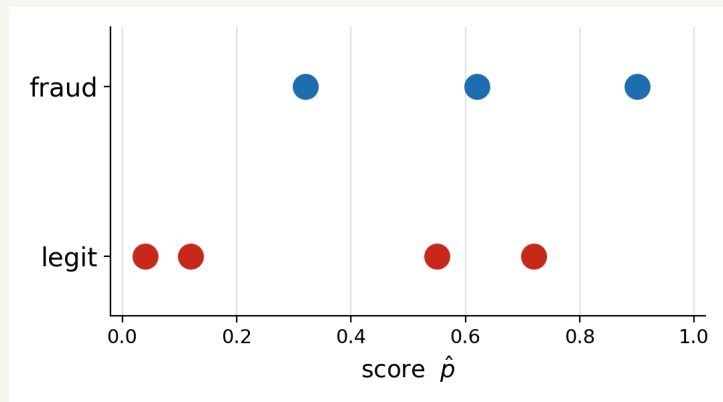
A perfect ranking gets 1. What does **0.91** mean ?

ROC-AUC is a probability

Scores: S^+ for a fraud, S^- for a legit transaction. A **good** model tends to have $S^+ > S^-$.

$$\text{ROC-AUC} = \mathbb{P}(S^+ > S^-)$$

$$\overline{\text{ROC-AUC}} = \frac{\text{number of pairs } (S^+, S^-) \text{ in the right order}}{\text{number of pairs } (S^+, S^-)}$$



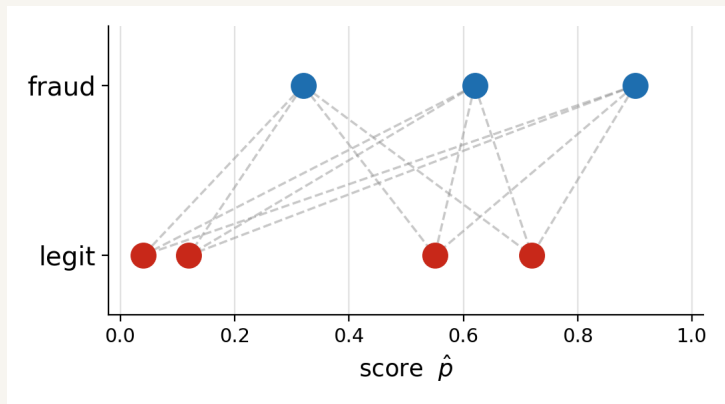
How many pairs (S^+, S^-) ?

ROC-AUC is a probability

Scores: S^+ for a fraud, S^- for a legit transaction. A **good** model tends to have $S^+ > S^-$.

$$\text{ROC-AUC} = \mathbb{P}(S^+ > S^-)$$

$$\overline{\text{ROC-AUC}} = \frac{\text{number of pairs } (S^+, S^-) \text{ in the right order}}{\text{number of pairs } (S^+, S^-)}$$



How many pairs (S^+, S^-) ?

$$3 \times 4 = 12$$

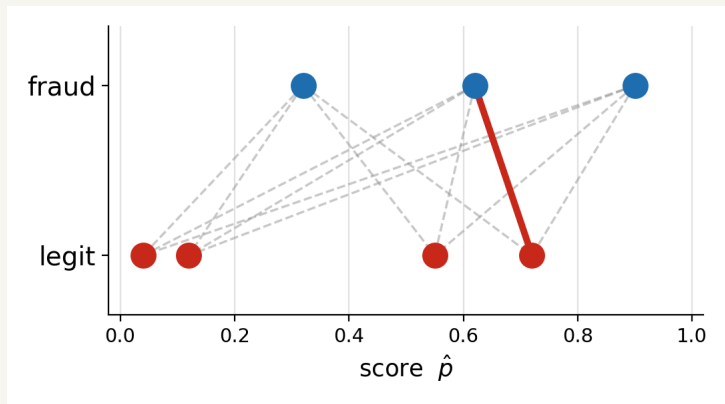
How many in the **wrong** order ?

ROC-AUC is a probability

Scores: S^+ for a fraud, S^- for a legit transaction. A **good** model tends to have $S^+ > S^-$.

$$\text{ROC-AUC} = \mathbb{P}(S^+ > S^-)$$

$$\overline{\text{ROC-AUC}} = \frac{\text{number of pairs } (S^+, S^-) \text{ in the right order}}{\text{number of pairs } (S^+, S^-)}$$



How many pairs (S^+, S^-) ?

$$3 \times 4 = 12$$

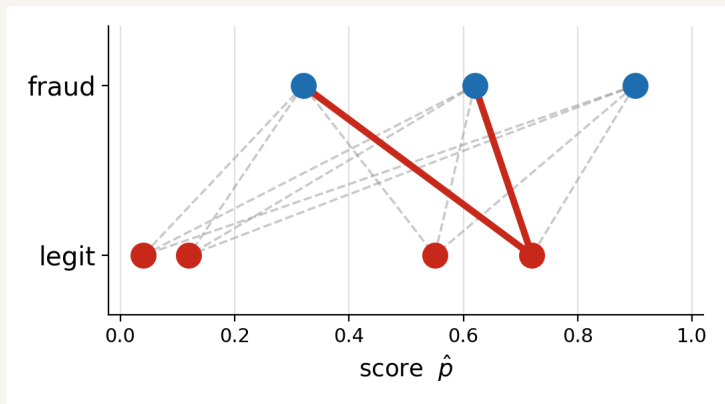
How many in the **wrong** order ?

ROC-AUC is a probability

Scores: S^+ for a fraud, S^- for a legit transaction. A **good** model tends to have $S^+ > S^-$.

$$\text{ROC-AUC} = \mathbb{P}(S^+ > S^-)$$

$$\overline{\text{ROC-AUC}} = \frac{\text{number of pairs } (S^+, S^-) \text{ in the right order}}{\text{number of pairs } (S^+, S^-)}$$



How many pairs (S^+, S^-) ?

$$3 \times 4 = 12$$

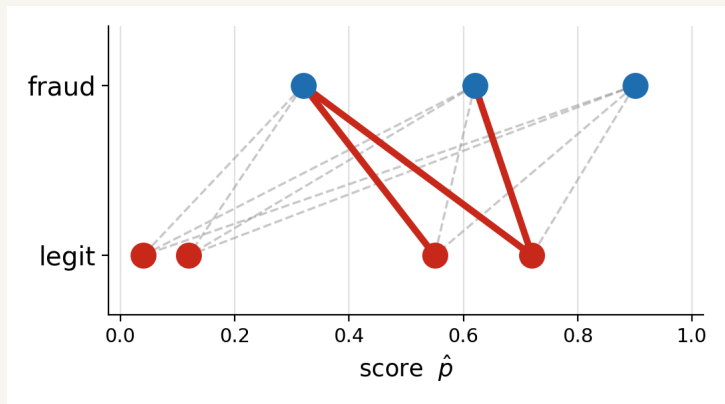
How many in the **wrong** order ?

ROC-AUC is a probability

Scores: S^+ for a fraud, S^- for a legit transaction. A **good** model tends to have $S^+ > S^-$.

$$\text{ROC-AUC} = \mathbb{P}(S^+ > S^-)$$

$$\overline{\text{ROC-AUC}} = \frac{\text{number of pairs } (S^+, S^-) \text{ in the right order}}{\text{number of pairs } (S^+, S^-)}$$



How many pairs (S^+, S^-) ?

$$3 \times 4 = 12$$

How many in the **wrong** order ?

3

$$\overline{\text{ROC-AUC}} = \frac{12 - 3}{12} = \frac{9}{12} = 0.75$$

The ROC-AUC quantifies the quality of the *ranking* of the probability scores.

Exercise 1

Random scores

A model outputs scores S that are **independent** of the label Y (random scores), with a continuous distribution. The positive rate is $\pi = \mathbb{P}(Y = 1)$.

1. Consider a threshold t , what metric corresponds to $\mathbb{P}(Y = 1 \mid S > t)$? Deduce the precision-recall curve of random scores and its PR-AUC.
2. Show that $\text{TPR}(t) = \text{FPR}(t)$ for every t : the ROC curve is the diagonal. Deduce its ROC-AUC.

Exercise 2

The threshold as a decision

A bank's fraud desk scores each transaction $\mathbf{x}$ with $p = \mathbb{P}(Y = 1 \mid \mathbf{x})$, where $Y = 1$ means fraud. Flagging a legit transaction costs $c_{FP} = 20$ Dh (a phone call); letting a fraud through costs $c_{FN} = 664$ Dh (the average stolen amount). A correct prediction costs 0.

1. We denote this new *loss* by c ; Write down the general formula for $c(\hat{y}, y)$ with $\hat{y}$ being the model's prediction (discuss each case $\hat{y} = 0$ and $\hat{y} = 1$).
2. Write the expected cost $\mathbb{E}(c(\hat{y}, y) \mid \mathbf{x})$ when $\hat{y} = 1$ and when $\hat{y} = 0$ as a function of p , c_{FP} and c_{FN} .
3. When should the desk predict 1? Find the optimal threshold t^* in terms of c_{FP} and c_{FN} , and compute it for this bank.

More than two classes

From binary to multiclass probabilities

Binary logistic regression modeled one probability: $\hat{p}(\mathbf{x}) = \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x})$.

Now $K = 3$ classes (**Normal**, **Suspicious**, **Fraud**): we model $p_k(\mathbf{x}) = \mathbb{P}(Y = k \mid \mathbf{X} = \mathbf{x})$.

What **constraints** must $p_1(\mathbf{x}), \dots, p_K(\mathbf{x})$ satisfy ?

$$p_k(\mathbf{x}) \geq 0 \quad \text{and} \quad \sum_{k=1}^K p_k(\mathbf{x}) = 1$$

One linear score per class: $s_k(\mathbf{x}) = \beta_k^\top \mathbf{x}$, for example $s(\mathbf{x}) = (2.1, -0.3, 0.8)$

How do we turn these scores into **probabilities** ?

$$p_k(\mathbf{x}) = \frac{\exp(s_k(\mathbf{x}))}{\sum_{j=1}^K \exp(s_j(\mathbf{x}))} \quad (\text{Softmax})$$

$$(e^{2.1}, e^{-0.3}, e^{0.8}) = (8.17, 0.74, 2.23) \longrightarrow p(\mathbf{x}) = (0.73, 0.07, 0.20)$$

The raw scores $s_k(\mathbf{x})$ are called the **logits**.

From probabilities to the log loss

$$s(\mathbf{x}) = (2.1, -0.3, 0.8) \xrightarrow{\text{softmax}} p(\mathbf{x}) = (0.73, 0.07, 0.20) \text{ over (Normal, Suspicious, Fraud)}$$

Suppose true label is $y = 2$ (Suspicious). How should we score this prediction ?

Use the binary log loss with the **observed** class:

$$-\log p_2(\mathbf{x}) = -\log 0.07 \approx 2.7$$

Now suppose the true label is $y = 1$ (Normal). Its score ?

$$-\log p_1(\mathbf{x}) = -\log 0.73 \approx 0.31$$

In general: $\ell(y, p(\mathbf{x})) = -\log p_y(\mathbf{x})$, the minus log-probability given to the **observed** class.

Regularized softmax

The multiclass log loss of row i :

$$L(y_i, p(\mathbf{x}_i)) = -\log p_{y_i}(\mathbf{x}_i)$$

the **regularized** problem, exactly as before:

$$\hat{\mathbf{B}}_\lambda = \operatorname{argmin}_{\beta_1, \dots, \beta_K} \frac{1}{n} \sum_i L(y_i, p(\mathbf{x}_i)) + \lambda \sum_k \|\beta_k\|^2$$

and the **lasso variant**, when we also want few active coefficients per class:

$$\hat{\mathbf{B}}_\lambda^{\text{lasso}} = \operatorname{argmin}_{\beta_1, \dots, \beta_K} \frac{1}{n} \sum_i L(y_i, p(\mathbf{x}_i)) + \lambda \sum_k \sum_j |\beta_{kj}|$$

Evaluating K classes

True ↓ / Predicted →	Normal	Suspicious	Fraud
Normal	760	35	5
Suspicious	40	95	15
Fraud	5	10	35

Global metrics: one number for all classes.

$$\text{accuracy} = \frac{760 + 95 + 35}{1\,000} = 0.89$$

$$\text{error} = 1 - 0.89 = 0.11$$

Per class metrics: each class in turn is the "positive" one:

class	precision	recall
Normal	?	?
Suspicious	?	?
Fraud	?	?

Evaluating K classes

Global metrics: one number for all classes.

$$\text{accuracy} = \frac{760 + 95 + 35}{1\,000} = 0.89$$

$$\text{error} = 1 - 0.89 = 0.11$$

Per class metrics: each class in turn is the "positive" one:

class	precision	recall
Normal	760 / 805 = 0.94	?
Suspicious	?	?
Fraud	?	?

Evaluating K classes

Global metrics: one number for all classes.

$$\text{accuracy} = \frac{760 + 95 + 35}{1\,000} = 0.89$$

$$\text{error} = 1 - 0.89 = 0.11$$

Per class metrics: each class in turn is the "positive" one:

class	precision	recall
Normal	760 / 805 = 0.94	760 / 800 = 0.95
Suspicious	?	?
Fraud	?	?

Evaluating K classes

Global metrics: one number for all classes.

$$\text{accuracy} = \frac{760 + 95 + 35}{1\,000} = 0.89$$

$$\text{error} = 1 - 0.89 = 0.11$$

Per class metrics: each class in turn is the "positive" one:

class	precision	recall
Normal	760 / 805 = 0.94	760 / 800 = 0.95
Suspicious	95 / 140 = 0.68	?
Fraud	?	?

Evaluating K classes

Global metrics: one number for all classes.

$$\text{accuracy} = \frac{760 + 95 + 35}{1\,000} = 0.89$$

$$\text{error} = 1 - 0.89 = 0.11$$

Per class metrics: each class in turn is the "positive" one:

class	precision	recall
Normal	760 / 805 = 0.94	760 / 800 = 0.95
Suspicious	95 / 140 = 0.68	95 / 150 = 0.63
Fraud	?	?

Evaluating K classes

Global metrics: one number for all classes.

$$\text{accuracy} = \frac{760 + 95 + 35}{1\,000} = 0.89$$

$$\text{error} = 1 - 0.89 = 0.11$$

Per class metrics: each class in turn is the "positive" one:

class	precision	recall
Normal	760 / 805 = 0.94	760 / 800 = 0.95
Suspicious	95 / 140 = 0.68	95 / 150 = 0.63
Fraud	35 / 55 = 0.64	?

Evaluating K classes

Global metrics: one number for all classes.

$$\text{accuracy} = \frac{760 + 95 + 35}{1\,000} = 0.89$$

$$\text{error} = 1 - 0.89 = 0.11$$

Per class metrics: each class in turn is the "positive" one:

class	precision	recall
Normal	$760 / 805 = 0.94$	$760 / 800 = 0.95$
Suspicious	$95 / 140 = 0.68$	$95 / 150 = 0.63$
Fraud	$35 / 55 = 0.64$	$35 / 50 = 0.70$
macro average	$(0.94+0.68+0.64)/3 = 0.75$	$(0.95+0.63+0.70)/3 = 0.76$

What to remember

topic	remember
classification	a regression on the probability $\hat{p}(x) = P(Y = 1 \mid x)$, then a threshold
regularization	exactly as in regression: ridge, lasso, elastic net
comparing models	across all thresholds, not only at $t = \frac{1}{2}$
imbalanced classes	accuracy is misleading: a model that never flags is right 988 times in 1 000
ROC-AUC	the probability that a random (positive, negative) pair is ranked correctly
PR-AUC	takes the class imbalance into account

See **chapter 5** (logistic regression, penalties) and **chapter 6** (evaluating predictors) of the textbook.

Contents

1. Classification
2. Evaluating a classifier
3. More than two classes