# Introduction to Machine Learning

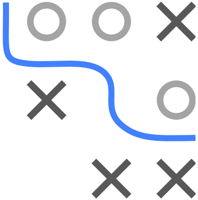
# Evaluation
# Simple Measures for Classification

| | | **True Class** $y$ | |
|---|---|---|---|
| | | $+$ | $-$ |
| **Pred.** | $+$ | True Positive (TP) | False Positive (FP) |
| $\hat{y}$ | $-$ | False Negative (FN) | True Negative (TN) |

## Learning goals

- Know the definitions of misclassification error rate (MCE) and accuracy (ACC)
- Understand the entries of a confusion matrix
- Understand the idea of costs
- Know definitions of Brier score and log loss

# LABELS VS PROBABILITIES



In classification we predict:

1. Class labels:

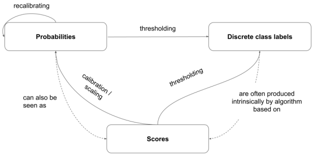
$$\boldsymbol{F} = \left(\hat{o}_k^{(i)}\right)_{i\in\{1,\dots,m\},k\in\{1,\dots,g\}} \quad \in \mathbb{R}^{m\times g},$$

   where $\hat{o}_k^{(i)} = [\hat{y}^{(i)} = k], k = 1, \dots, g$ is the one-hot-encoded class label prediction.

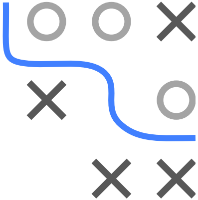
2. Class probabilities:

$$\boldsymbol{F} = \left(\hat{\pi}_k^{(i)}\right)_{i\in\{1,\dots,m\},k\in\{1,\dots,g\}} \quad \in [0,1]^{m\times g}$$

$\rightarrow$ These form the basis for evaluation.

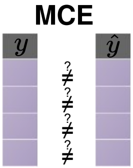


# LABELS: MCE & ACC



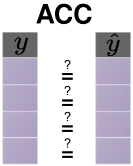
The **misclassification error rate (MCE)** counts the number of incorrect predictions and presents them as a rate:

$$\rho_{MCE} = \frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \neq \hat{y}^{(i)}] \in [0, 1].$$



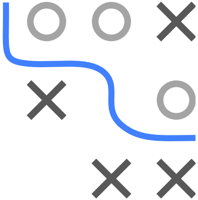

**Accuracy (ACC)** is defined in a similar fashion for correct classifications:

$$\rho_{ACC} = \frac{1}{m}\sum_{i=1}^{m}[y^{(i)} = \hat{y}^{(i)}] \in [0, 1].$$




- If the data set is small this can be brittle.
- MCE says nothing about how good/skewed predicted probabilities are.
- Errors on all classes are weighted equally, which is often inappropriate.

# LABELS: CONFUSION MATRIX



Much better than reducing prediction errors to a simple number is tabulating them in a **confusion matrix**:

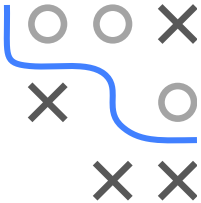
- true classes in columns,
- predicted classes in rows.

We can nicely see class sizes (predicted/true) and where errors occur.

True classes

| | | **setosa** | **versicolor** | **virginica** | **error** | *n* |
|---|---|---|---|---|---|---|
| Predicted classes | **setosa** | 50 | 0 | 0 | 0 | 50 |
| | **versicolor** | 0 | 46 | 4 | 4 | 50 |
| | **virginica** | 0 | 4 | 46 | 4 | 50 |
| | **error** | 0 | 4 | 4 | 8 | - |
| | *n* | 50 | 50 | 50 | - | 150 |

# LABELS: CONFUSION MATRIX



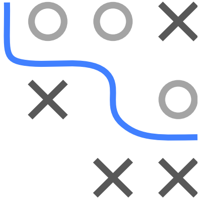
- In binary classification, we typically call one class "positive" and the other "negative".
- The positive class is the more important, often smaller one.

| | | **True Class** $y$ | |
|---|---|---|---|
| | | $+$ | $-$ |
| **Pred.** | $+$ | True Positive (TP) | False Positive (FP) |
| $\hat{y}$ | $-$ | False Negative (FN) | True Negative (TN) |

e.g.,

- **True Positive** (TP) means that an instance is classified as positive that is really positive (correct prediction).
- **False Negative** (FN) means that an instance is classified as negative that is actually positive (incorrect prediction).

# LABELS: COSTS



We can also assign different costs to different errors via a **cost matrix**.

$$Costs = \frac{1}{n}\sum_{i=1}^{n} C[y^{(i)}, \hat{y}^{(i)}]$$

Example: Depending on certain features (age, income, profession, ...) a bank wants to decide whether to grant a 10,000 EUR loan.

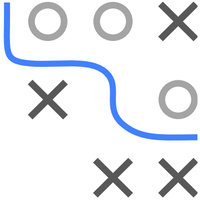
Predict if a person is solvent (yes / no).
Should the bank lend them the money?

**Examplary costs:**

| | |
|---|---|
| Loss in event of default: | 10,000 EUR |
| Income through interest paid: | 100 EUR |

| | | True classes | |
|---|---|---|---|
| | | **solvent** | **not solvent** |
| Predicted | **solvent** | 0 | 10,000 |
| classes | **not solvent** | 100 | 0 |

# LABELS: COSTS



### Cost matrix

| | | True classes | |
|---|---|---|---|
| | | **solvent** | **not solvent** |
| Predicted classes | **solvent** | 0 | 10,000 |
| | **not solvent** | 100 | 0 |

### Confusion matrix

| | | True classes | |
|---|---|---|---|
| | | **solvent** | **not solvent** |
| Predicted classes | **solvent** | 70 | 3 |
| | **not solvent** | 7 | 20 |

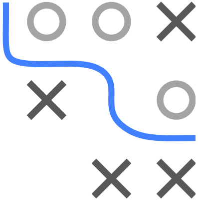
- If the bank gives everyone a credit, who was predicted as *solvent*, the costs are at:

$$\begin{aligned} Costs &= \frac{1}{n}\sum_{i=1}^{n} C[y^{(i)}, \hat{y}^{(i)}] \\ &= \frac{1}{100}(100 \cdot 7 + 0 \cdot 70 + 10.000 \cdot 3 + 0 \cdot 20) = 307 \end{aligned}$$

- If the bank gives everyone a credit, the costs are at:

$$Costs = \frac{1}{100}(100 \cdot 0 + 0 \cdot 77 + 10.000 \cdot 23 + 0 \cdot 0) = 2.300$$

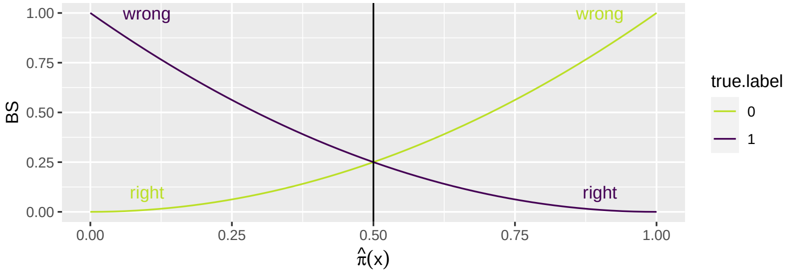
# PROBABILITIES: BRIER SCORE



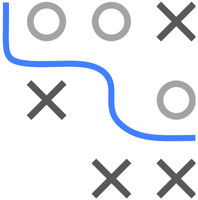
Measures squared distances of probabilities from the true class labels:

$$\rho_{BS} = \frac{1}{m} \sum_{i=1}^{m} \left( \hat{\pi}^{(i)} - y^{(i)} \right)^2$$

- Fancy name for MSE on probabilities.
- Usual definition for binary case; $y^{(i)}$ must be encoded as 0 and 1.



# PROBABILITIES: BRIER SCORE / 2



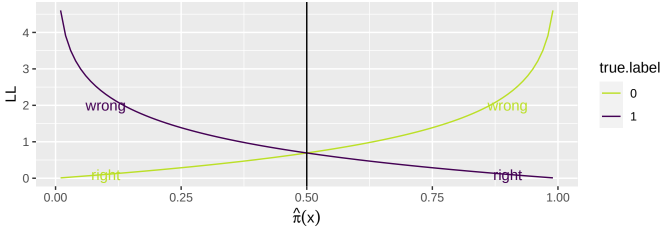
$$\rho_{BS,MC} = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{g}\left(\hat{\pi}_k^{(i)} - o_k^{(i)}\right)^2$$

- Original by Brier, works also for multiple classes.
- $o_k^{(i)} = [y^{(i)} = k]$ marks the one-hot-encoded class label.
- For the binary case, $\rho_{BS,MC}$ is twice as large as $\rho_{BS}$: in $\rho_{BS,MC}$, we sum the squared difference for each observation regarding both class 0 **and** class 1, not only the true class.

# PROBABILITIES: LOG-LOSS

Logistic regression loss function, a.k.a. Bernoulli or binomial loss, $y^{(i)}$ encoded as 0 and 1.

$$\rho_{LL} = \frac{1}{m} \sum_{i=1}^{m} \left( -y^{(i)} \log\left(\hat{\pi}^{(i)}\right) - \left(1 - y^{(i)}\right) \log\left(1 - \hat{\pi}^{(i)}\right)\right).$$




- Optimal value is 0, “confidently wrong” is penalized heavily.
- Multi-class version: $\rho_{LL,MC} = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{g} o_k^{(i)} \log\left(\hat{\pi}_k^{(i)}\right).$