# Choose the good metric

**TASK**: Classification of skin cancer tumors (based on images)

- Decide between two models A and B to predict $y \in \{\text{malignant}, \text{ benign}\}$

| | *Precision* | *Recall* |
|---|---|---|
| *Model A* | 93% | 85% |
| *Model B* | 86% | 95% |

- **Precision:** Of examples classified as malignant, what % are actually malignant?
- **Recall:** What % of malignant examples are actually classified as malignant?
- F1-score: **Avarage** of P and R : $\frac{2}{1/P+1/R}$

What a good evaluation metrics is depends on your problem

# Changing evaluation metrics

- **Metric**: Classification error = 1- classification accuracy
  - Model A: 5%
  - Model B: 8%
- **Model A** A seems to do better but you prefer model B since it is doing better on malignant melanoma tumors which you are especially interested in.
- What to do ?
  - Change the metrics

$$\frac{1}{\sum_{i=1}^{m} w_i} \sum_{i=1}^{m} w_i \mathbf{1}_{\{\widehat{y_i} \neq y_i\}}, \quad w_i = \begin{cases} 10 & \text{if malignant melanoma} \\ 1 & \text{otherwise} \end{cases}$$

- and/or your validation/test dataset distribution (by including more malignant melanoma examples).

# Changing evaluation metrics

- How do we make sure that we train our model towards this metric?
- If we have reweighed the metric, we can do the same for our training objective.

$$\text{Metric} := \frac{1}{\sum_{i=1}^{m} w_i} \sum_{i=1}^{m} w_i \mathbf{1}_{\{\widehat{y_i} \neq y_i\}}, \quad \text{Loss Function} := \frac{1}{\sum_{i=1}^{m} w_i} \sum_{i=1}^{m} w_i L(\widehat{y_i} \neq y_i)$$

**Key point:**

-1 Define your problem by choosing metric and validation/test dataset.

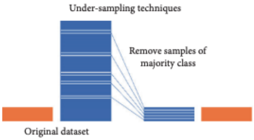
-2 Adapt your learning algorithm to do well on your metric

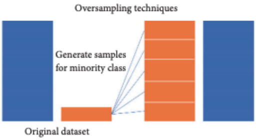
# Imbalanced Dataset

**Class imbalance problem** in multiple areas: telecommunication managements, bioinformatics, fraud detection, medical diagnosis, . . .

Accuracy is not the good metric to look at: why ?

- A survey on Resampling approach



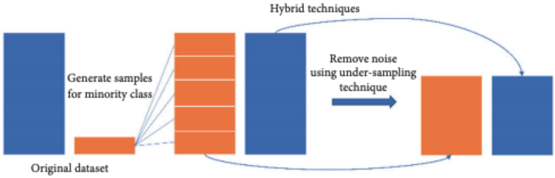




**Drawback**
leave out important instances that provide important differences between the two classes.

**Drawback**
lead to model overfitting by introducing duplicate instances, drawing from a pool of instances that is already small.

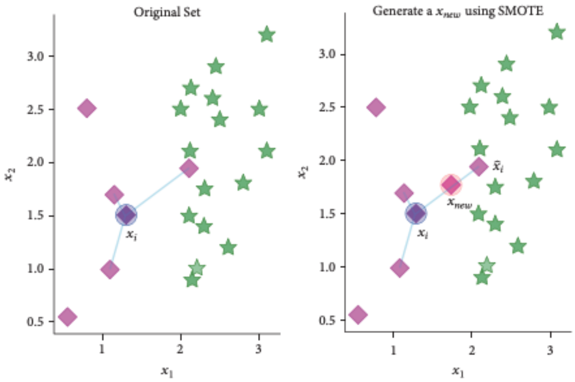
# Hybrid approach

- Combining Oversampling and Undersampling



# SMOTE (Synthetic Minority Oversampling Technique)

- Creates new instances of the **minority class** by creating convex combinations of neighboring instances.



# Edited Nearest Neighbor (ENN)

**Edited Nearest Neighbor** undersampling of the majority class is done by removing points whose class label differs from a majority of its *k* nearest neighbor.

- Combine SMOTE and ENN: **SMOTE-ENN**