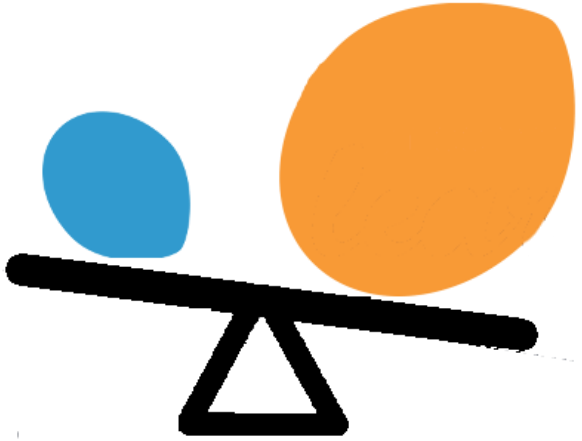
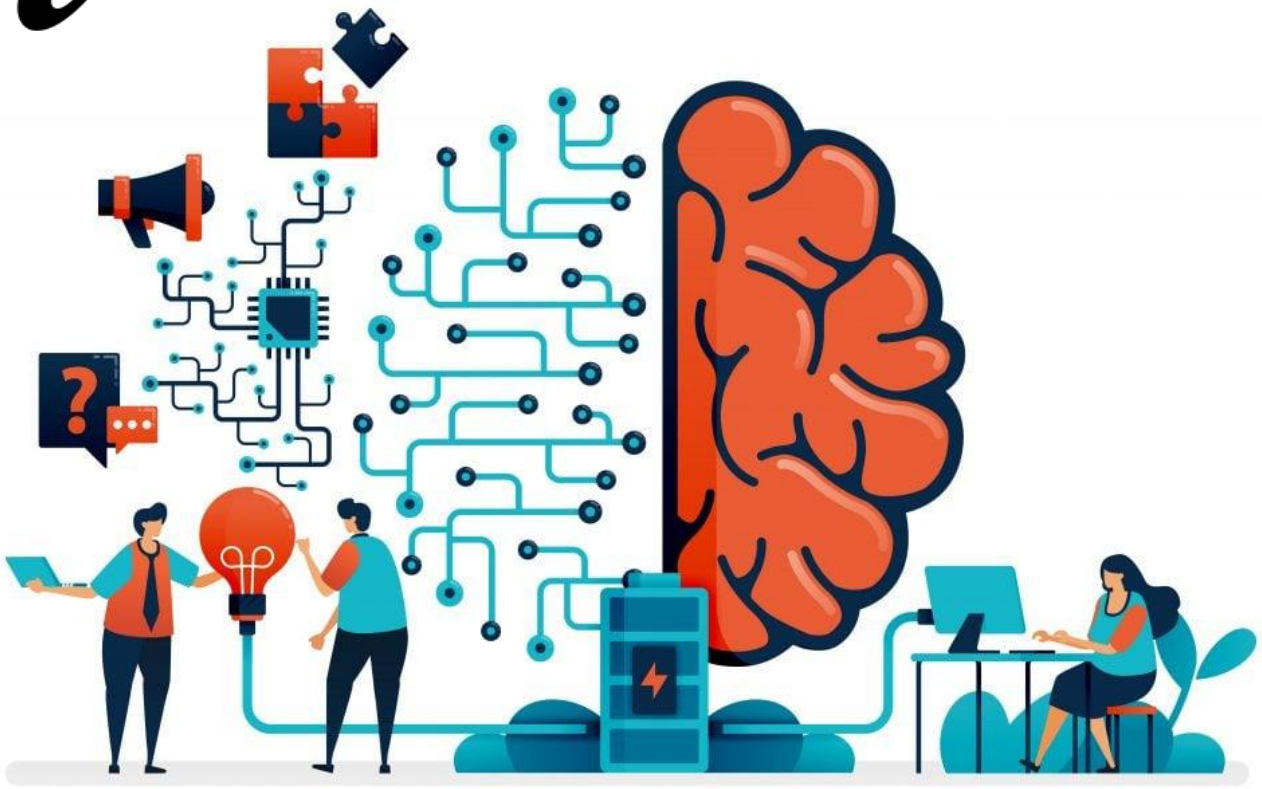


Imbalanced learn

Imbalanced Classification: Fundamentals, Solutions and More

Master in Data Science and Computer Engineering

Alberto Fernández – DASCI Institue. University of Granada



Objectives

To understand the nature of singular problems in classification, especially those related to imbalanced classes

To establish suited metrics to analyse the behavior in such scenarios

To acknowledge the actual challenges for uneven represented concepts and how those might be diminished

To know the current solutions for avoiding class bias in modeling

To consider relevant future lines of work in the topic

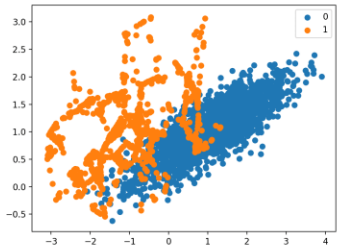
To know find tools and methods for addressing imbalanced data

Outline

1. Introduction: Definition, properties and difficulty
2. Evaluation metrics
3. Addressing imbalanced datasets
4. Software tools for classification with imbalanced data
5. Final Comments and Surveys for a deeper study

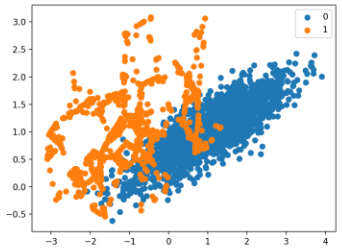
Outline

1. **Introduction: Definition, properties and difficulty**
2. Evaluation metrics
3. Addressing imbalanced datasets
4. Software tools for classification with imbalanced data
5. Final Comments and Surveys for a deeper study



Motivation: A corporation ask our Data Science team to detect fraud transactions

- We train a RandomForest model and obtain 95% accuracy
- The model is put into production, and one month later we are fired from the company
- WHAT HAPPENED?
- The system is not useful it could not predict any suspicious transaction



Motivation: Why a 95% accuracy is not worth it?

- We must take a look to the **confusion matrix** to understand what happened
- None of the “important” transactions (high cost) have been identified
- Majority class overrepresentation biased the classification model:
  - In case of “doubt” it is safer to predict the most frequent behavior

Test Set	Predicted Pos.	Predicted Neg.
Actual Positive	0	50
Actual Negative	0	950

Problem Definition

Real application areas characterised by having a **very different distribution** of examples among their classes.

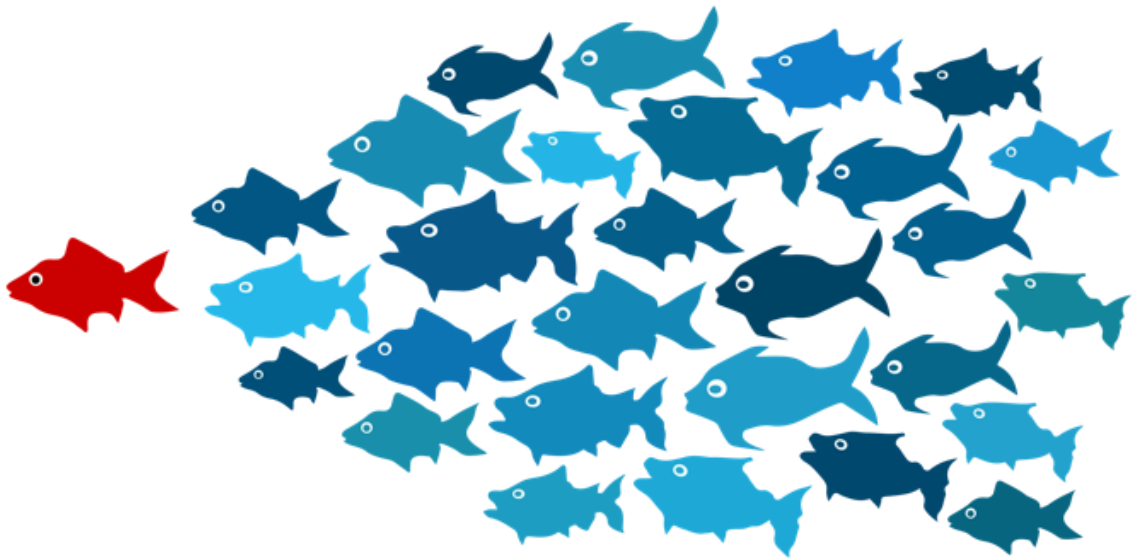
Intrinsic to the problem or due to **limitations** during data collection process. Namely: **Biased sampling** (related to representation bias)

Positive class often represents the concept of the highest interest for the problem, whereas the negative class represents counter-examples.

Problem of imbalanced data-sets: imposes a **bias** for the correct identification of the different concepts to be learnt.

Imbalanced Data: Definition and Measures

- Class Imbalance refers to a **disproportion in the number of examples belonging to each class** in a dataset and is known to bias classifiers towards the most representative concepts.



- **Ratio** (e.g., 1:100)
- **Percentage of minority class examples (%)**
- Entropy of class proportions:

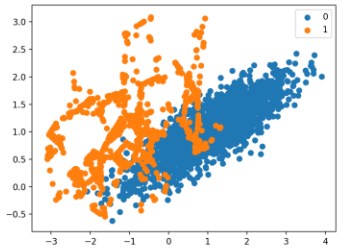
$$C1 = -\frac{1}{\log(n_c)} \sum_{i=1}^{n_c} p_{c_i} \log(p_{c_i})$$

- **Imbalance Ratio:**

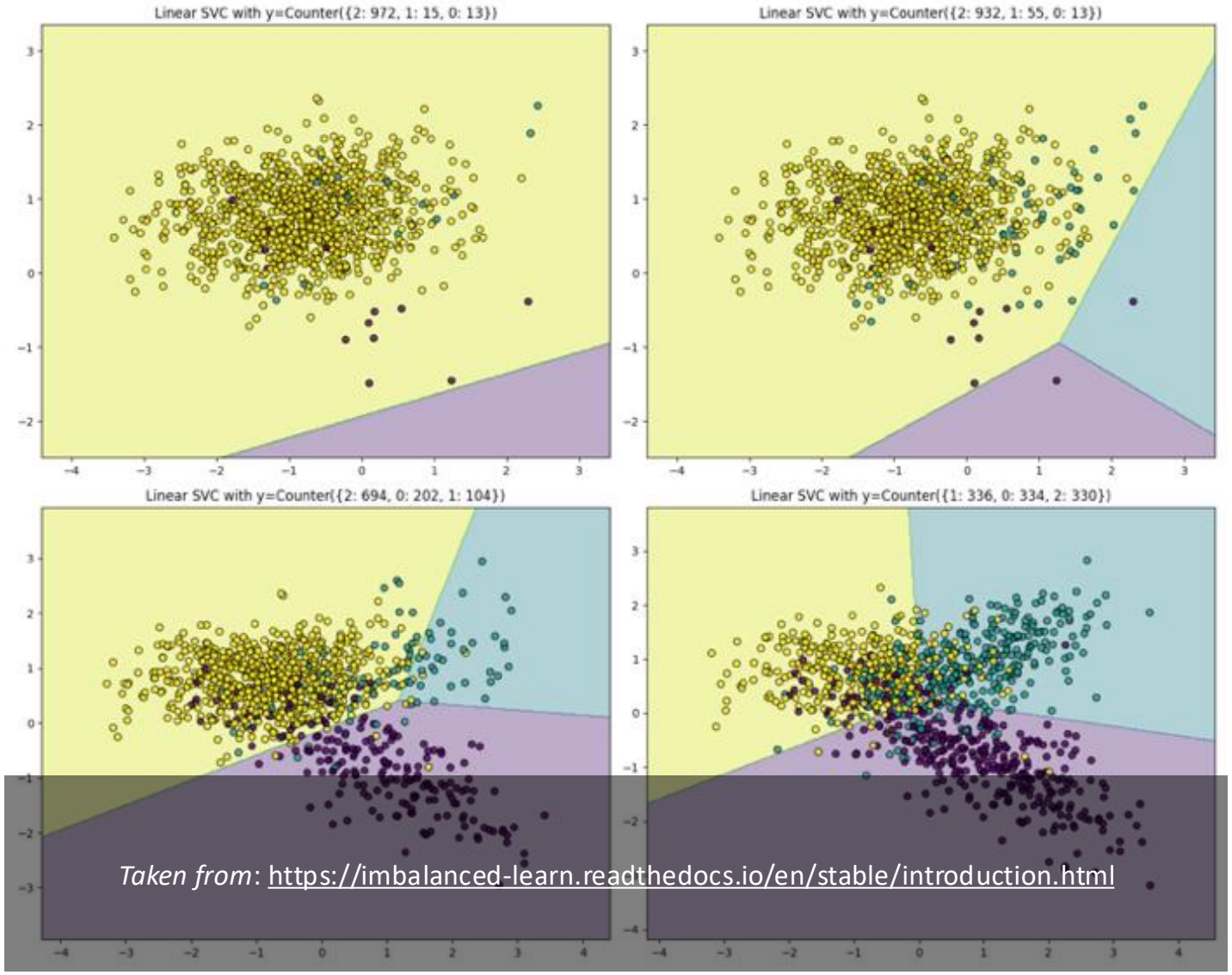
$$C2 = 1 - \frac{1}{IR} \qquad IR = \frac{n_c - 1}{n_c} \sum_{i=1}^{n_c} \frac{n_{c_i}}{n - n_{c_i}}$$

Class imbalance	Entropy of classes proportions	C1	0	1
	Imbalance ratio	C2	0	1

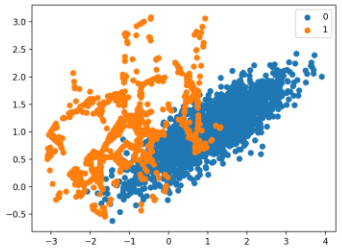
$$IR = \frac{n_{maj}}{n_{min}}$$



The importance of data representation

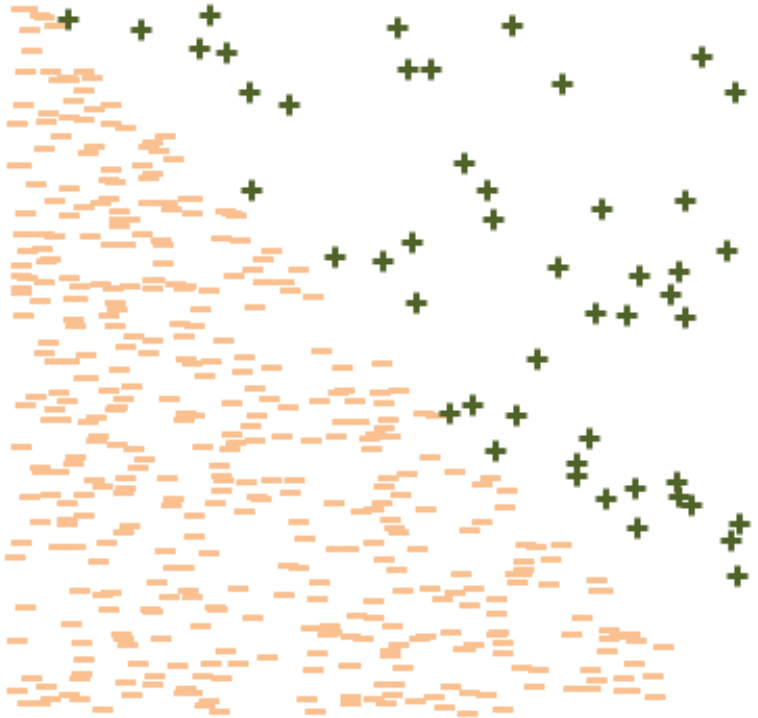


Taken from: https://imbalanced-learn.readthedocs.io/en/stable/introduction.html

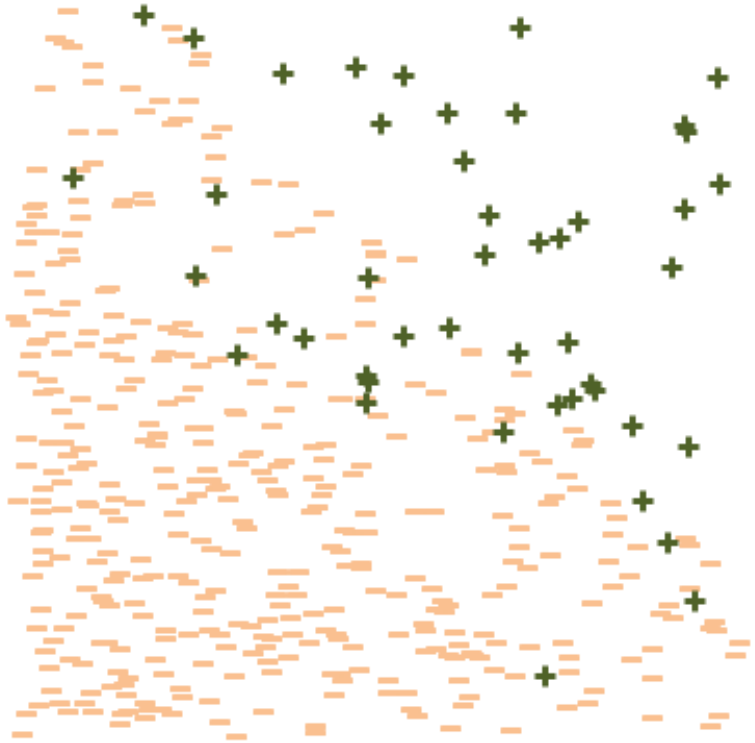


Intrinsic Data Issues: Same class ratio

Easy problem

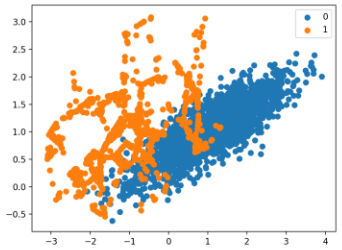


Difficult problem



UNIVERSIDAD DE GRANADA

V. López, A. Fernandez, S. García, V. Palade, F. Herrera, **An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics**. *Information Sciences* 250 (2013) 113-141



Real-areas applications for imbalanced

Medical Science and Bioinformatics

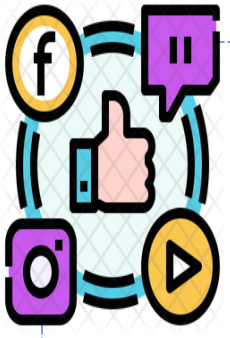
- **COVID infection:** among all patients, only 10% might have COVID
- Cancer diagnosis
- Protein classification

Information and Industry Security

- **Manufacturing defect:** different types of defects have different prevalance
- Malware detection
- Network intrusion and Spam detection

Economics and Marketing

- **Fraud Detection:** fraudulent transactions might make up 0.2% of all transactions
- Churn prediction
- Bankruptcy
- Loan Prediction

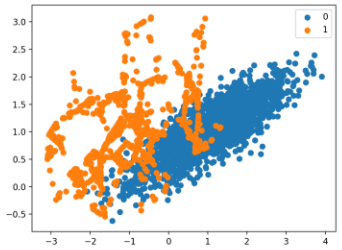


Social media and common use:

- **Self-driving cars:** objects have different prevalence (cars, trucks, pedestrians)
- Sentiment analysis
- Fake news classification

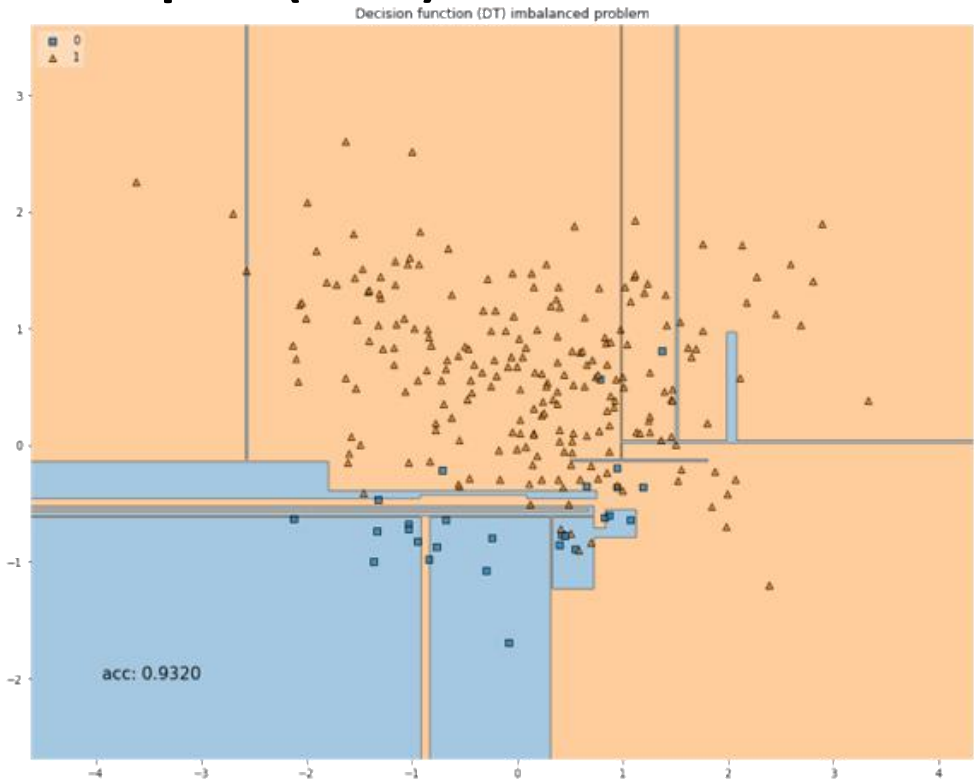
Outline

1. Introduction: Definition, properties and difficulty
2. **Evaluation metrics**
3. Addressing imbalanced datasets
4. Software tools for classification with imbalanced data
5. Final Comments and Surveys for a deeper study

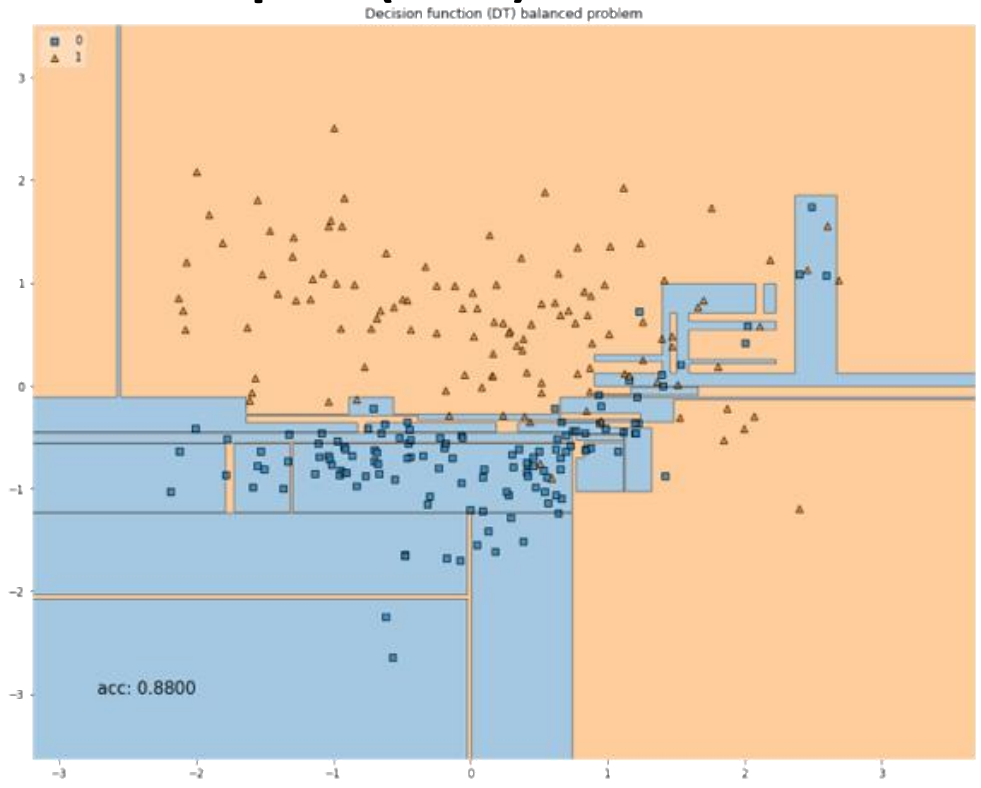


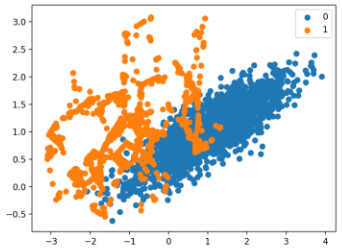
Evaluation: Common metrics (accuracy) may lead to erroneous conclusions

Imbalanced Problem: misses 7 out of 24 examples (blue): 30%



Balanced Problem: misses 15 out of 130 examples (blue): 11%





Evaluation: Measuring performance in imbalanced domains

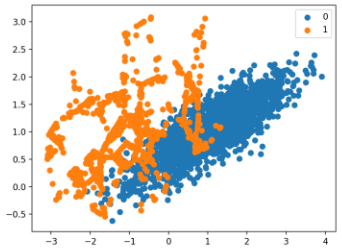
	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

Diagonal of True Hits

- Classical evaluation:

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

- Does not take into account the “Individual Rates”…
- … Very important in imbalanced problems



Sensitivity and Specificity

- *Recall* / Sensitivity (true positive ratio):
  - $TPR = \frac{TP}{TP+FN}$
- *Specificity* (true negative ratio):
  - $TNR = \frac{TN}{TN+FP}$
- *Precision*
  - $\text{Prec} = \frac{TP}{TP+FP}$
- *Geometric Mean:*
  - $GM = \sqrt{TPR \cdot TNR}$

Single class metrics provide a **unique vision** of the performance

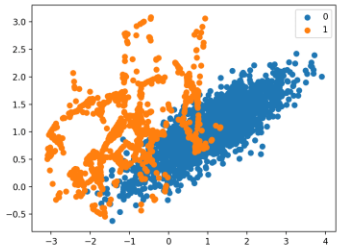
Sensitivity must be stressed

The fraction only considers same class samples

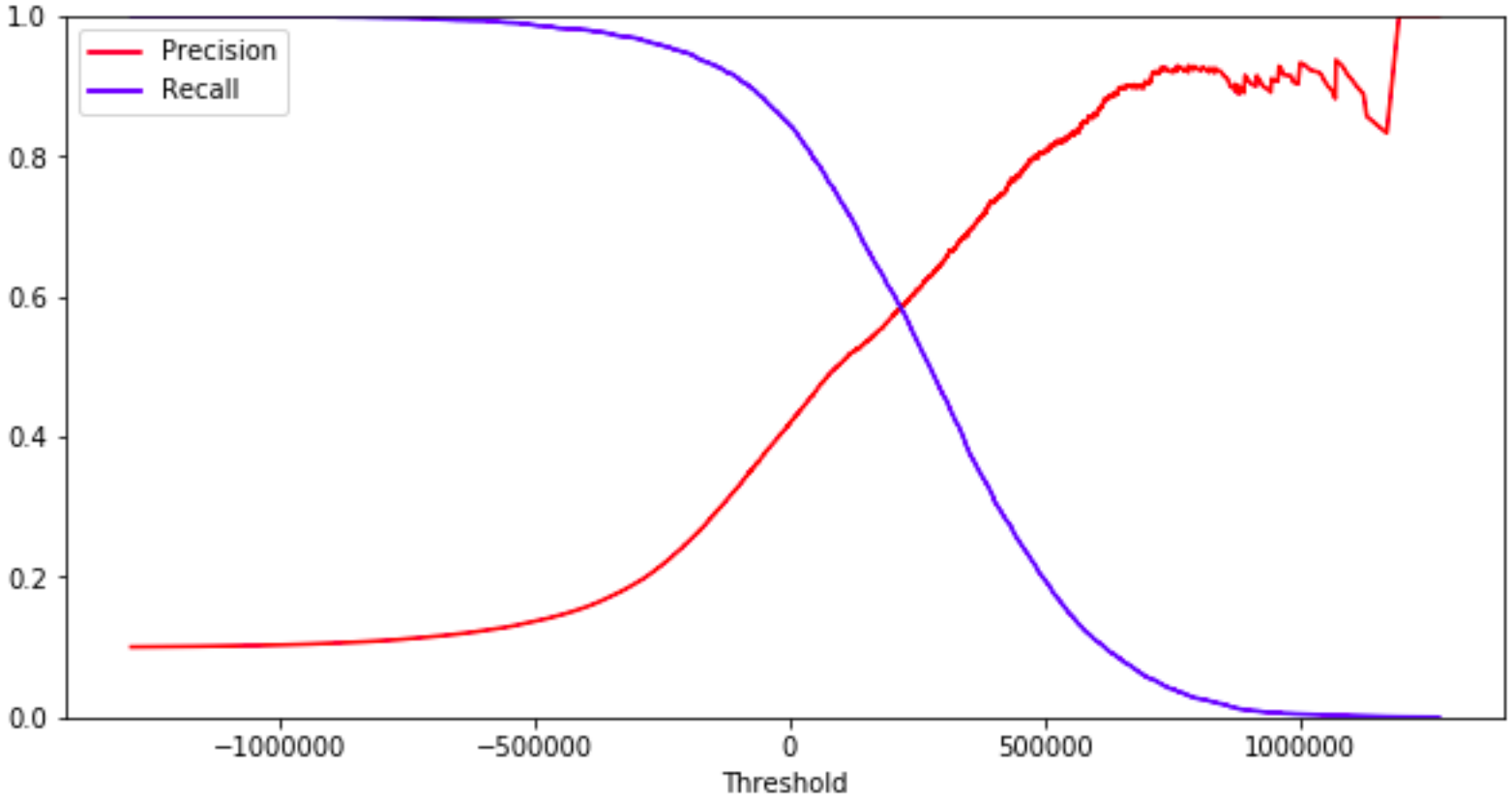
Aggregation functions are important for a global vision

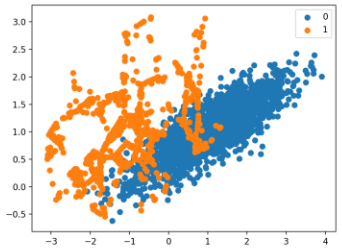
UNIVERSIDAD DE GRANADA

R. Barandela, J.S. Sánchez, V. García, E. Rangel. Strategies for learning in class imbalance problems. Pattern Recognition 36:3 (2003) 849-851



Precision vs Recall





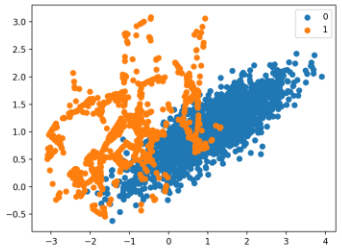
F-Measure

- F1 score is a harmonic mean between precision and recall:
  - Precision: number of correct positive results divided by the number of all positive results,
  - Recall / sensitivity: number of correct positive results divided by the number of positive results that should have been returned.

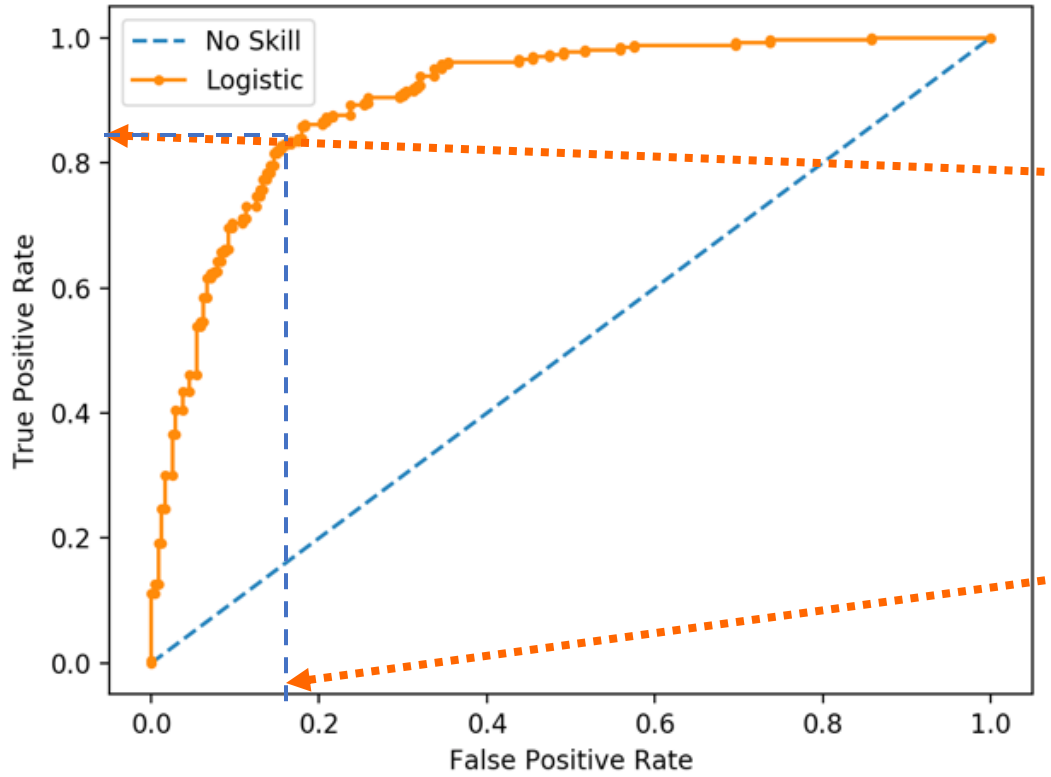
$$F_1 = 2 \cdot \frac{1}{\left(\frac{1}{recall} + \frac{1}{precision}\right)} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- General formula:

$$F_b = (1 + b^2) \cdot \frac{precision \cdot recall}{(b^2 \cdot precision) + recall}$$



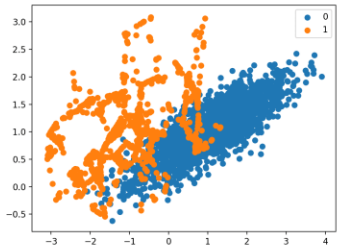
Area under ROC Curve (AUC): Scalar and graphical metric



	Positive Prediction	Negative Prediction
Positive Class	0.82	0.1
Negative Class	0.18	0.9

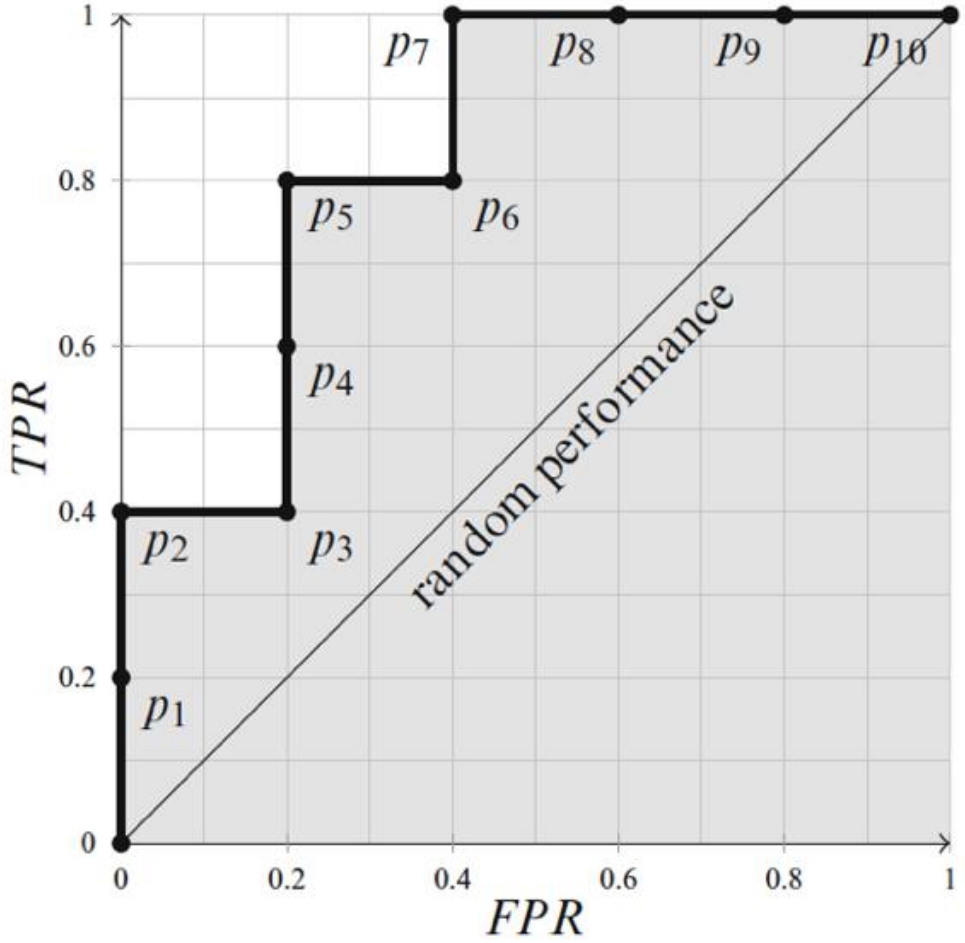
$$AUC = \frac{1 + TPR - FPR}{2}$$

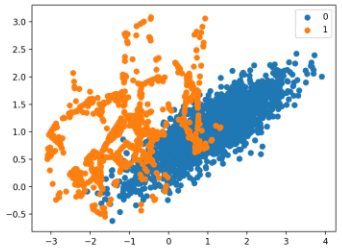
- Default Probability (0.5):



Area under ROC Curve (AUC)

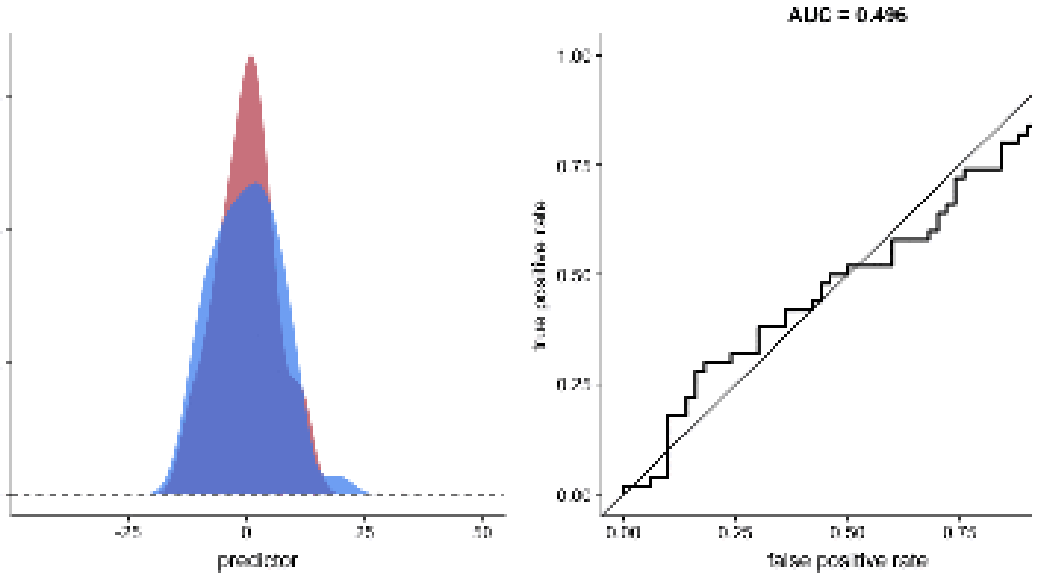
Rank	Score	Actual class	FPR	TPR	ROC point
#1	1.0	Positive	0.0	0.2	p_1
#2	0.9	Positive	0.0	0.4	p_2
#3	0.85	Negative	0.2	0.4	p_3
#4	0.7	Positive	0.2	0.6	p_4
#5	0.6	Positive	0.2	0.8	p_5
#6	0.45	Negative	0.4	0.8	p_6
#7	0.35	Positive	0.4	1.0	p_7
#8	0.3	Negative	0.6	1.0	p_8
#9	0.2	Negative	0.8	1.0	p_9
#10	0.05	Negative	1.0	1.0	p_{10}



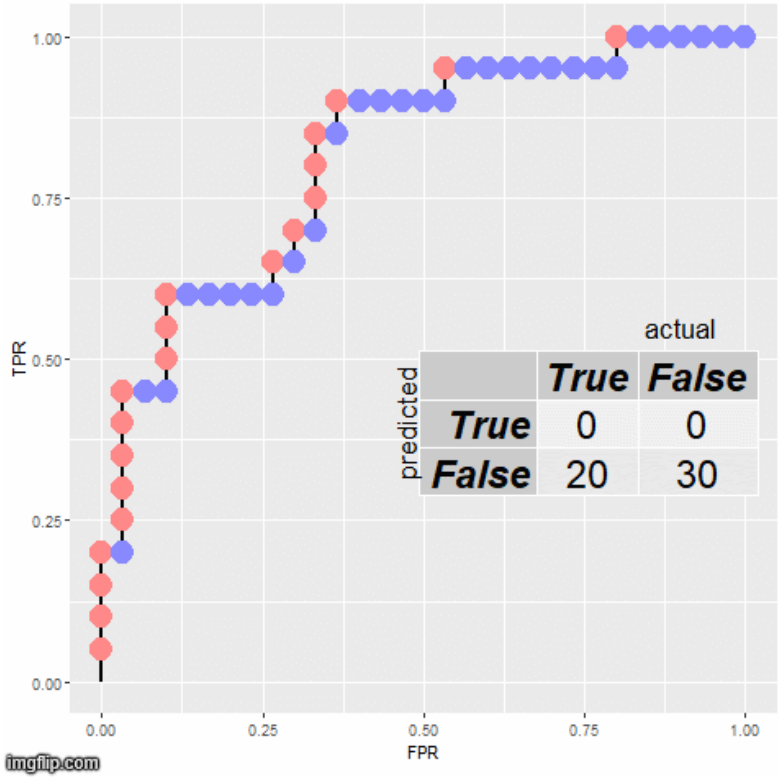


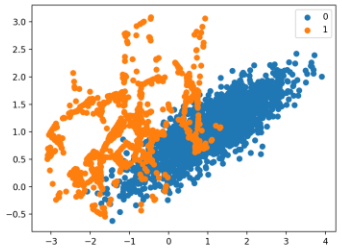
Area under ROC Curve (AUC)

AUC improves as classes separate

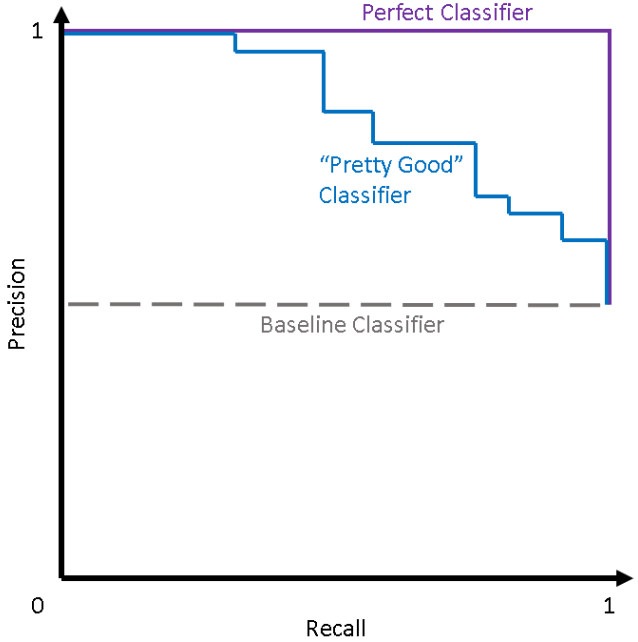


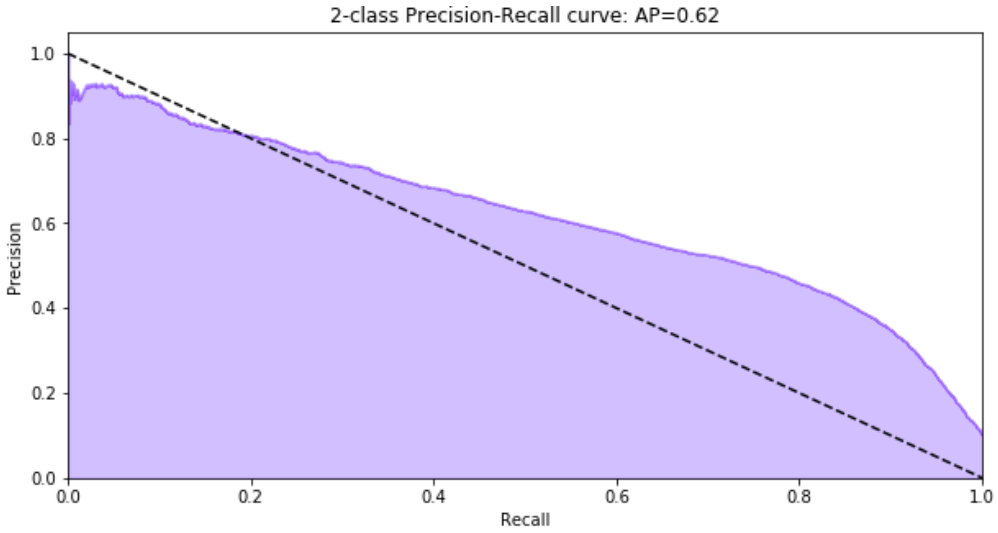
How curve is generated by threshold

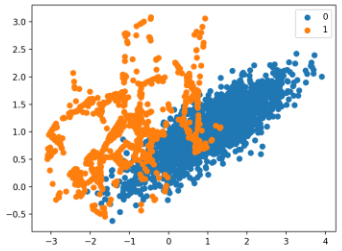




Precision-Recall Curve and average precisión (AP)







Differences between AUC and PR

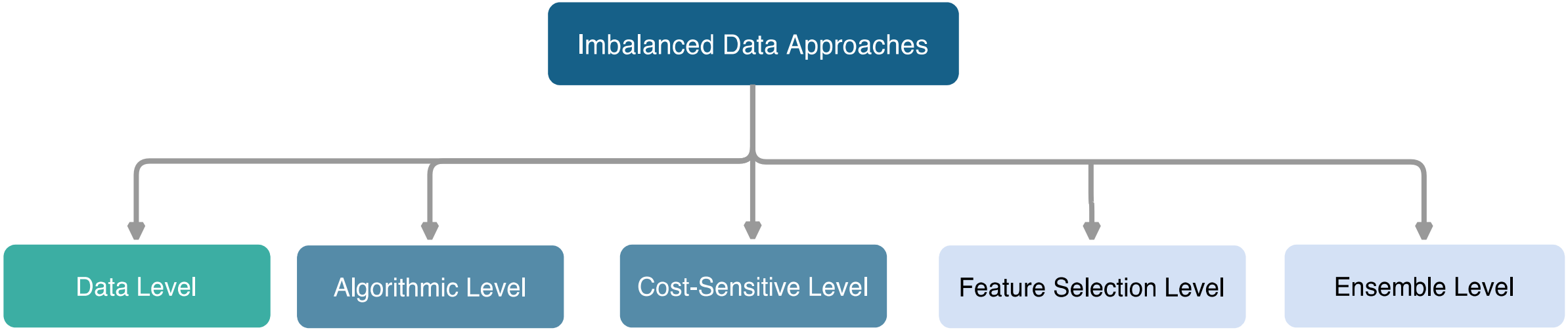
- We use AUC in both balanced and imbalanced problems
  - When there are few positive examples, AUCROC may give a high (optimistic) value
- PR curve (or Average Precision) is sought mainly for imbalanced datasets
- However, the PR curve will be far from its optimal value:
  - Revealing an accuracy indicator related to the low probability of the positive class.
- When the proportion of positive to negative instances changes, confusion matrix is the key:
  - ROC graphs are based upon TP rate and FP rate, in which each dimension is a strict columnar ratio, so do not depend on class distributions.
  - Precision, lift and F scores use values from both columns of the confusion matrix.

Outline

1. Introduction: Definition, properties and difficulty
2. Evaluation metrics
3. **Addressing imbalanced datasets**
4. Software tools for classification with imbalanced data
5. Final Comments and Surveys for a deeper study

Imbalanced Data: Data Level

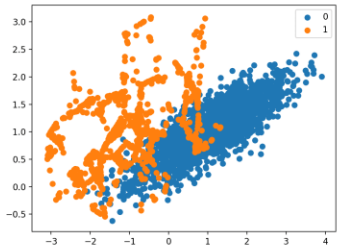
- **(Re)Sampling Methods:** Modify the prior distribution of the majority or/and the minority classes.



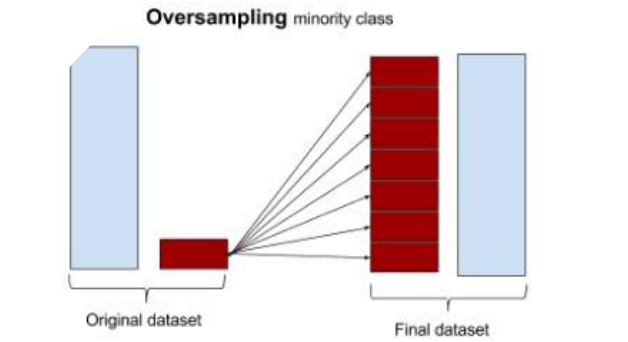
Imbalanced Data: Data-Level Approaches

Data Level

- **Data-Level approaches are the most commonly used.**
  - Have proven to be **efficient**.
  - Are rather **intuitive** and **simple** to implement.
  - **Classifier-agnostic**
  - **The most data-centric?: Can be adjusted to data intrinsic characteristics**
- **There are essentially two main categories:**
  - **Undersampling:** Removing majority examples.
  - **Oversampling:** Adding minority examples.



Preprocessing algorithms: Oversampling and Undersampling



Oversampling

- Keep influyent examples
- Reinforce clusters

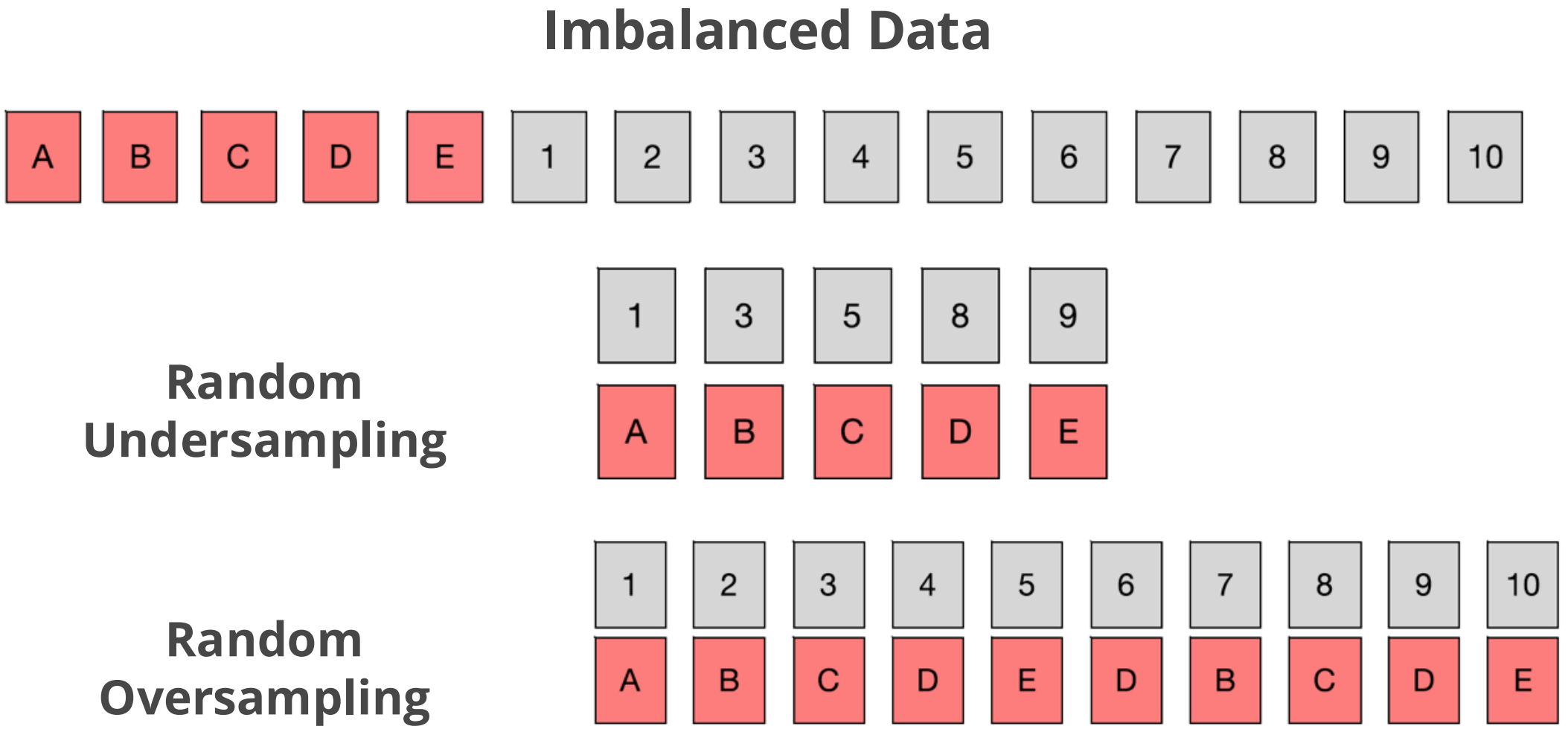
Over-Under Sampling

- Rebalance training set
- Avoid learning bias

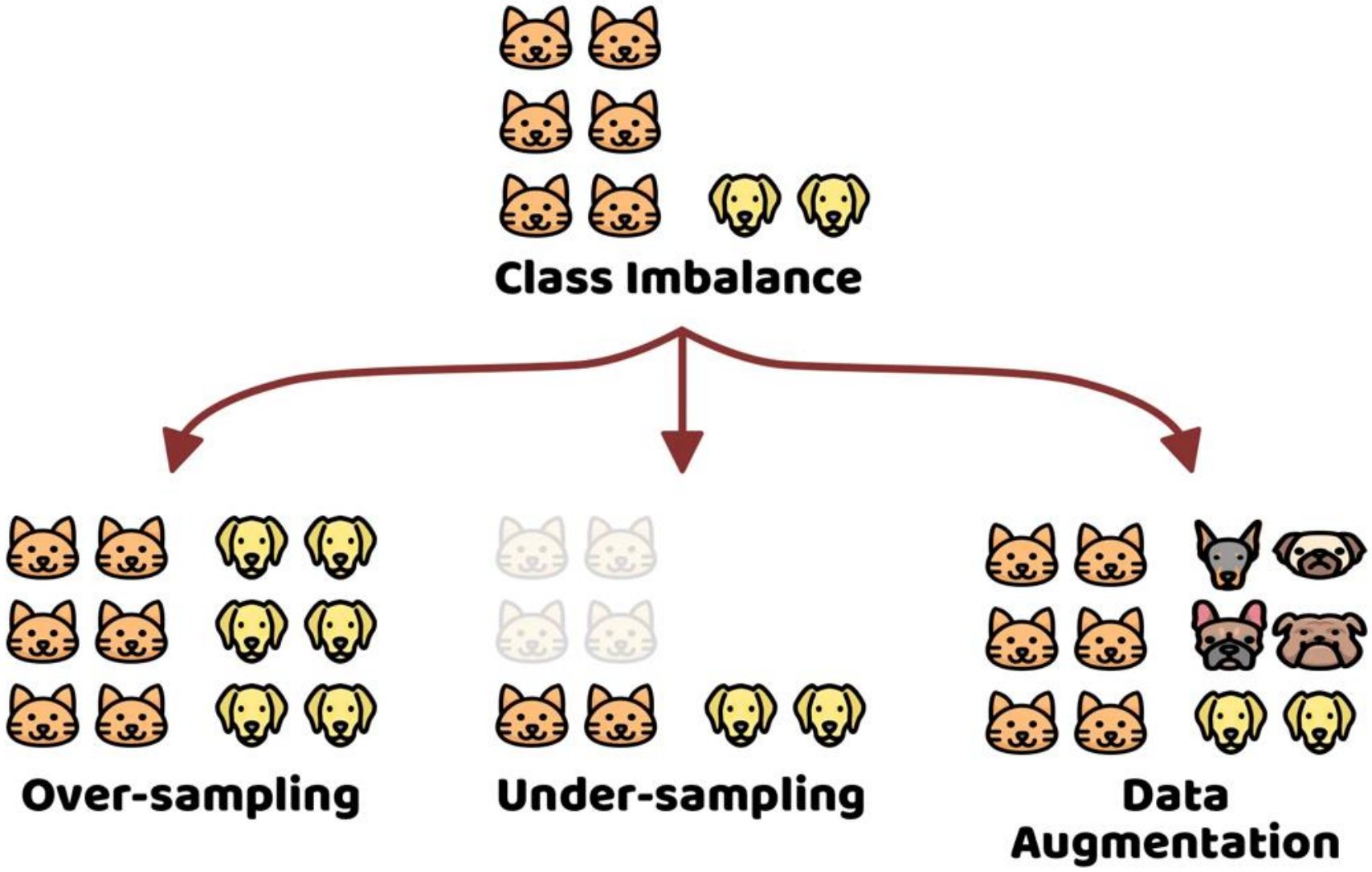
Undersampling

- Clean decision boundaries
- Reduce training set

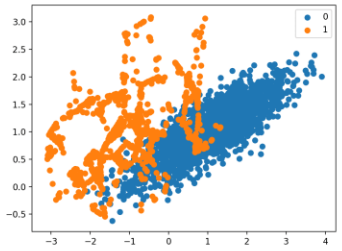
Imbalanced Data: RUS & ROS



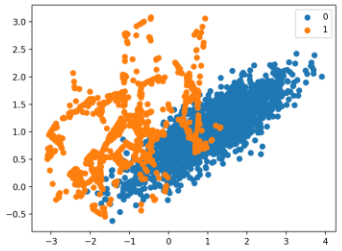
Imbalanced Data: Synthetic Oversampling



Undersampling techniques



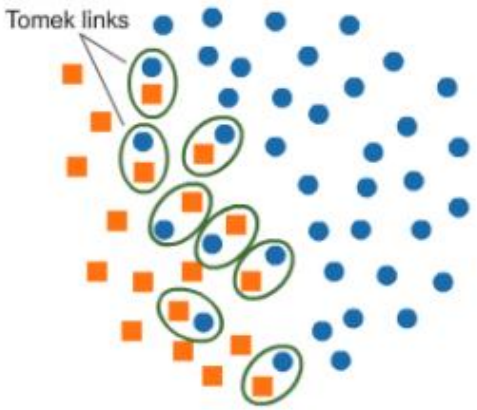
- Taxonomy
  - Basic techniques
    - TomekLinks
    - US-CNN
    - NCL
    - CPM
    - SBC
    - NearMiss
  - Advanced models
    - Undersampling by cleaning data
      - Weighted Sampling
      - IHT
      - Hybrid Undersampling
    - Clustering based undersampling
      - ClusterOSS
      - DSUS
    - Evolutionary undersampling
      - ACO sampling
      - IPADE-ID
      - CBEUS
    - Ensemble based undersampling
      - IRUS
      - OligoIS

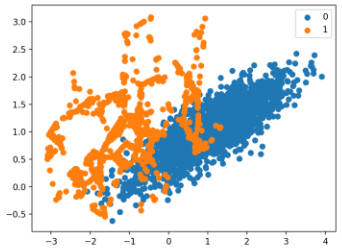


UnderSampling: Tomek Links (Cleaning)

Remove both noise and borderline examples (majority class)

- Ei, Ej belong to different classes,
- d (Ei, Ej): distance
- A (Ei, Ej) pair is called a Tomek link if there is no example El, such that
  - d(Ei, El) < d(Ei, Ej) or
  - d(Ej, El) < d(Ei, Ej).

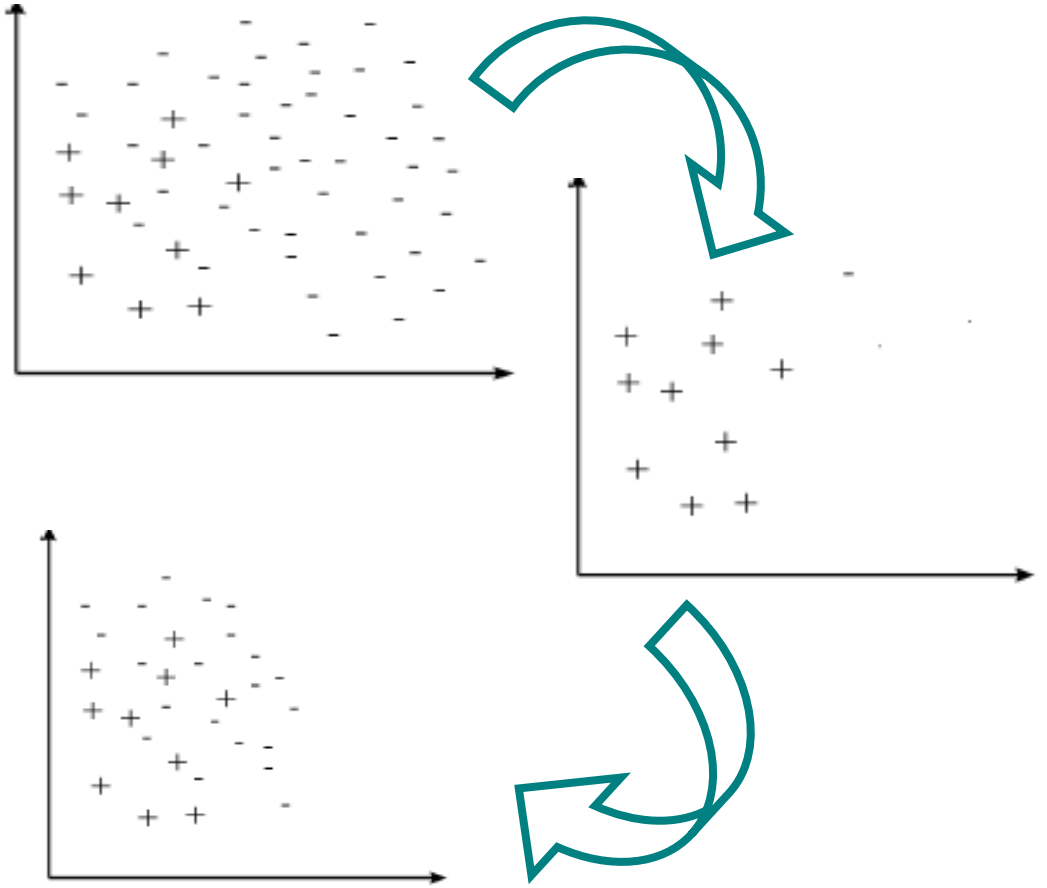


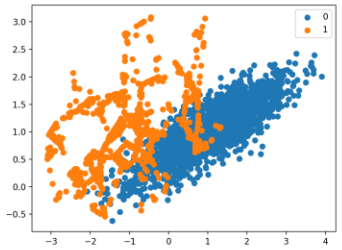


UnderSampling: CNN (Cleaning)

Remove noise and borderline

- Let E be the original training set
- Let E' contains all positive examples from S and one randomly selected negative example
- Classify E with the 1-NN rule using the examples in E'
- Move all misclassified example from E to E'





Preprocessing algorithms: SMOTE

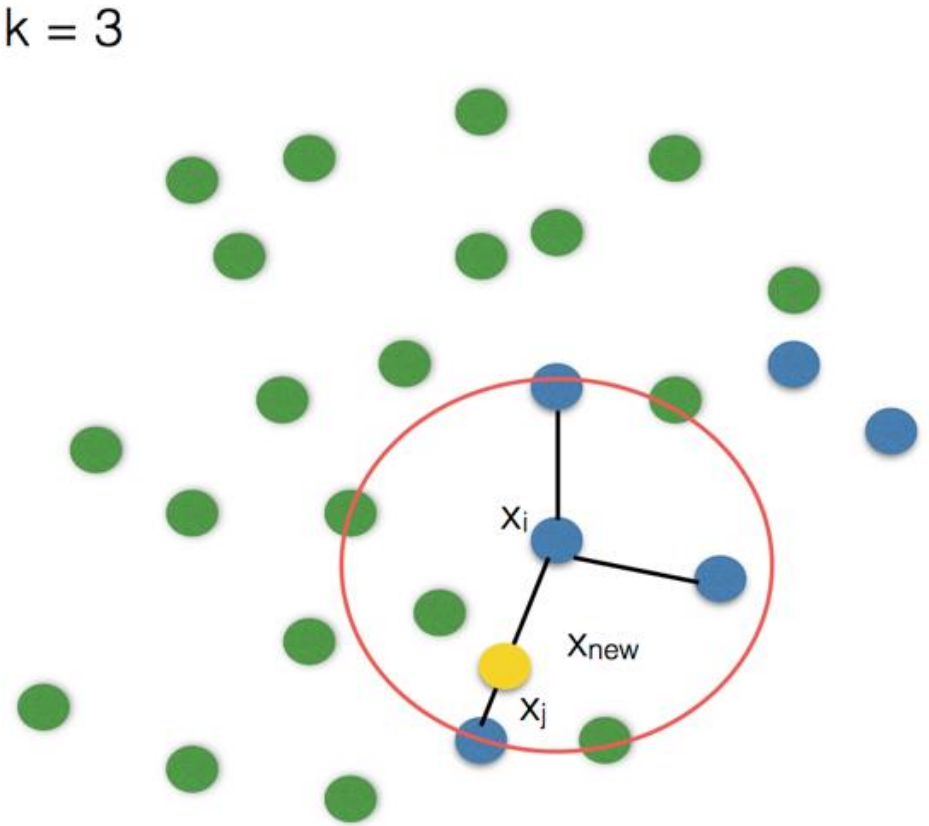
Oversampling: Simply replicating examples

Synthetic Minority Over-sampling Technique (**SMOTE**):

Generation of new minority class examples	**Interpolation** among several minority class instances that lie together

For each minority sample (j depends on the amount of oversampling desired)

Find its k-nearest minority neighbours	Randomly select j neighbours	Samples along the lines between i and {j}



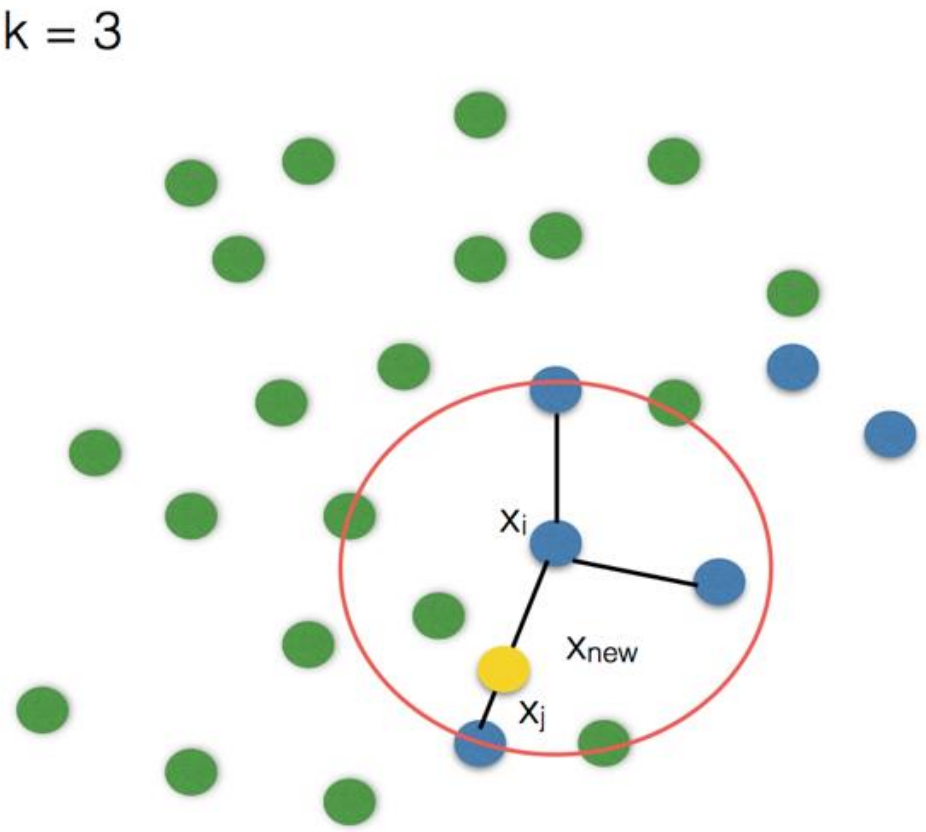
$$x_{new} = x_i + (x_j - x_i) \times \delta, \text{where } \delta \in [0, 1]$$

N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer. **SMOTE: synthetic minority over-sampling technique**. *Journal of Artificial Intelligence Research* 16 (2002) 321-357

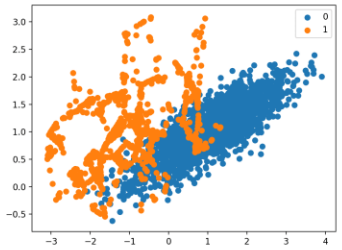
SMOTE: **S**ynthetic **M**inority **O**versampling **TE**chnique

Algorithm *SMOTE*(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE $N\%$; Number of nearest neighbors k
Output: $(N/100) * T$ synthetic minority class samples
1. (∗ *If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* ∗)
2. **if** $N < 100$
3. **then** Randomize the T minority class samples
4. $T = (N/100) * T$
5. $N = 100$
6. **endif**
7. $N = (int)(N/100)$ (∗ *The amount of SMOTE is assumed to be in integral multiples of 100.* ∗)
8. k = Number of nearest neighbors
9. *numattrs* = Number of attributes
10. *Sample*[ ][ ]: array for original minority class samples
11. *newindex*: keeps a count of number of synthetic samples generated, initialized to 0
12. *Synthetic*[ ][ ]: array for synthetic samples
(∗ *Compute k nearest neighbors for each minority class sample only.* ∗)
13. **for** $i \leftarrow 1$ **to** T
14. Compute k nearest neighbors for i, and save the indices in the *nnarray*
15. Populate(N, i, *nnarray*)
16. **endfor**

Populate(N, i, *nnarray*) (∗ *Function to generate the synthetic samples.* ∗)
17. **while** $N \neq 0$
18. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
19. **for** $attr \leftarrow 1$ **to** *numattrs*
20. Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
21. Compute: gap = random number between 0 and 1
22. $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
23. **endfor**
24. $newindex{+}{+}$
25. $N = N - 1$
26. **endwhile**
27. **return** (∗ *End of Populate.* ∗)
End of Pseudo-Code.

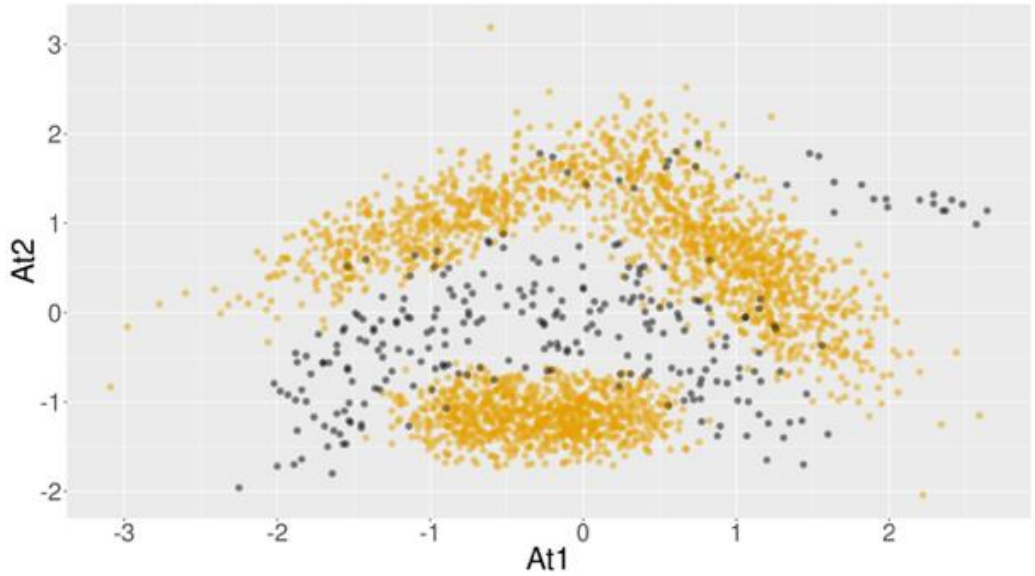


$$x_{new} = x_i + (x_j - x_i) \times \delta, \text{where } \delta \in [0, 1]$$

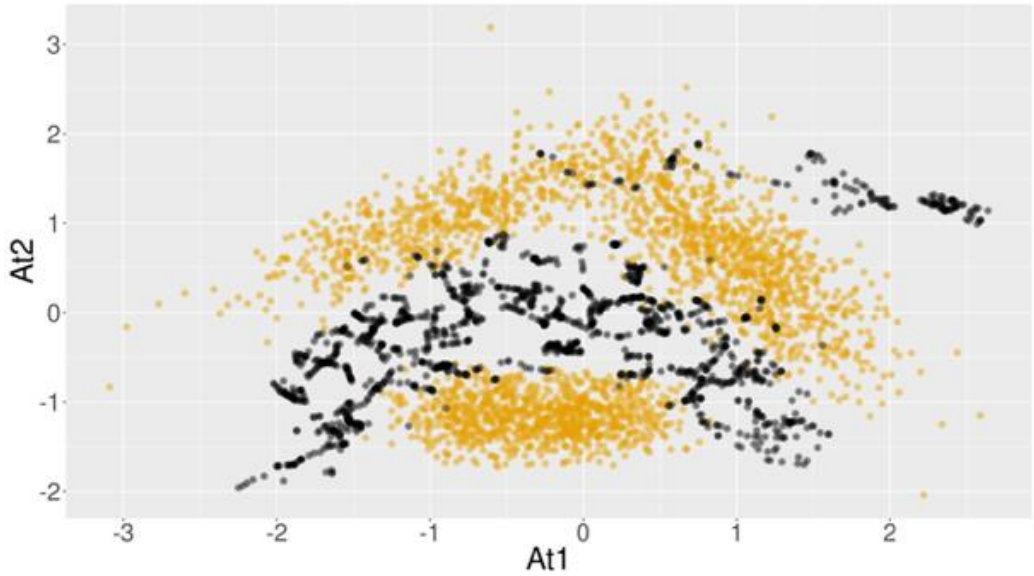


Illustrating SMOTE Preprocessing

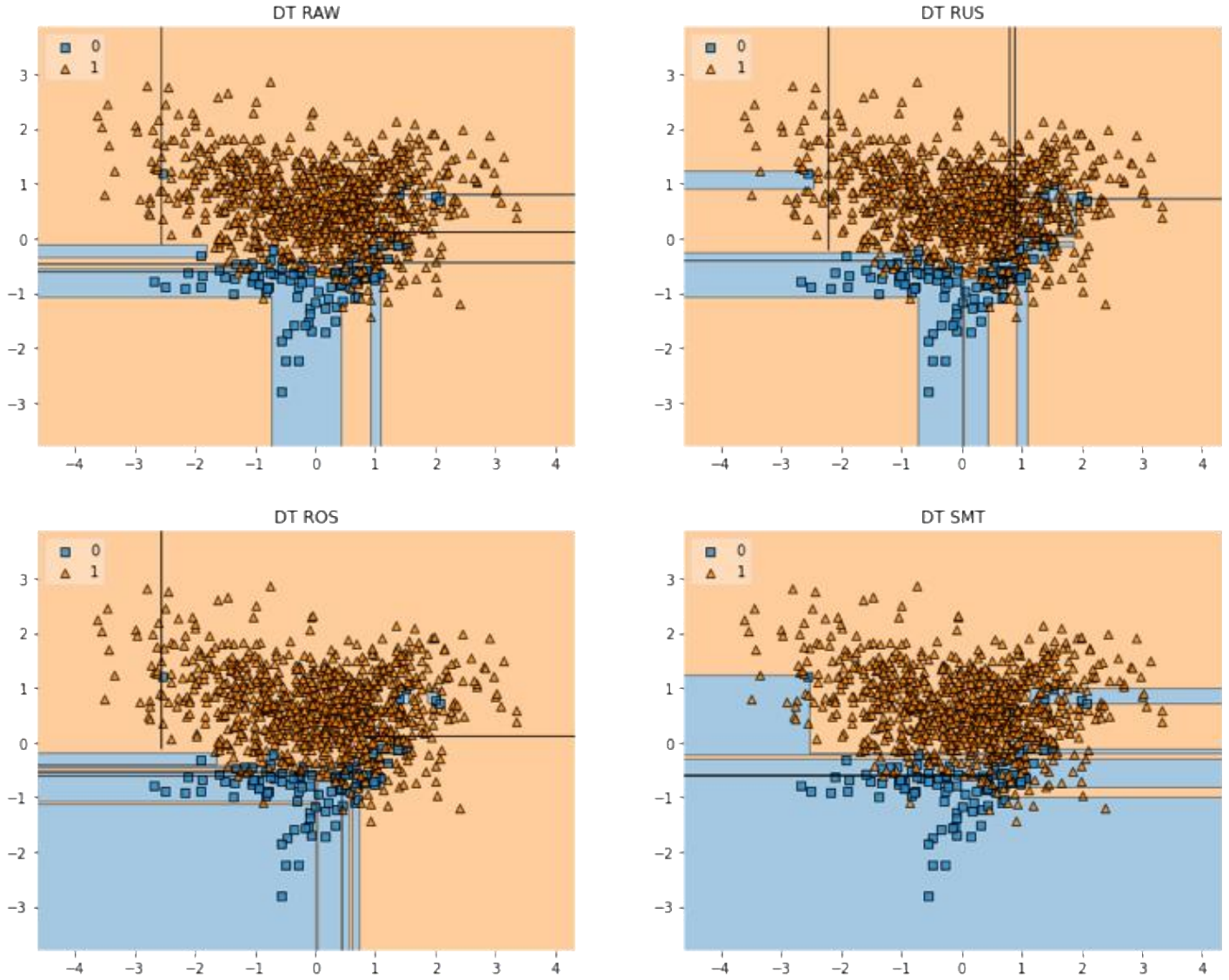
Original



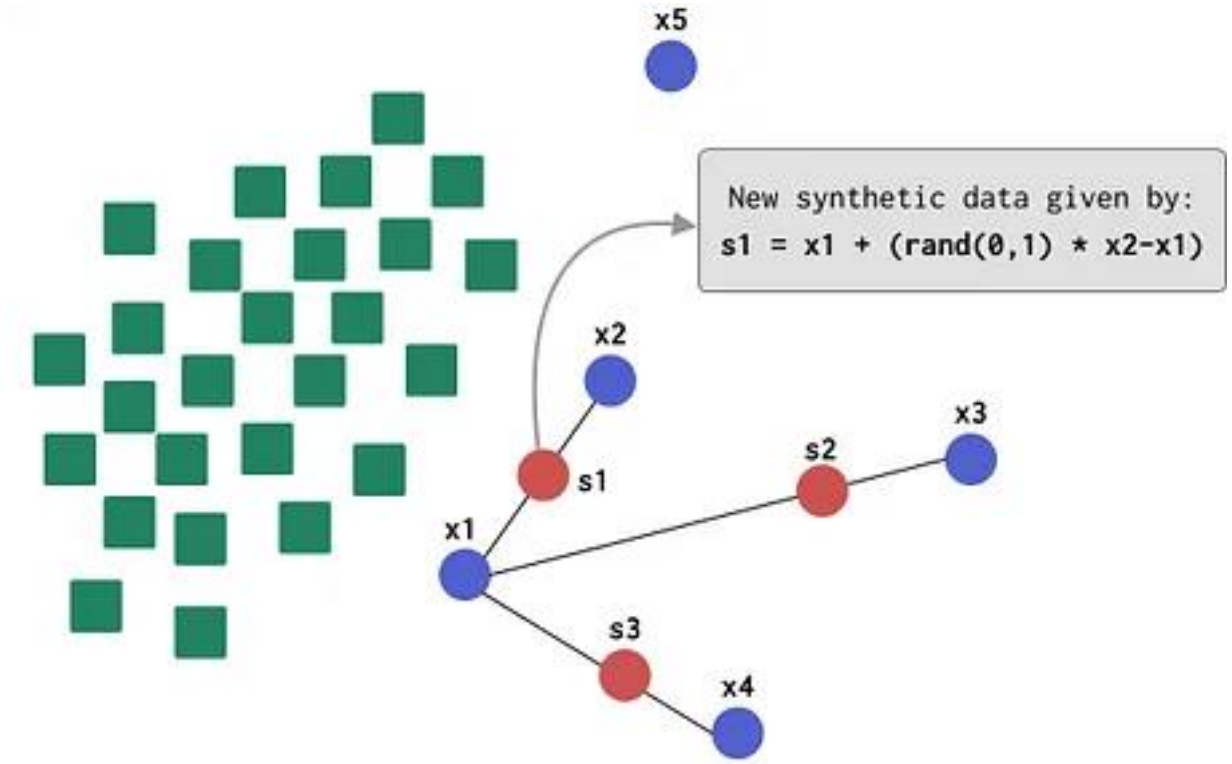
SMOTE

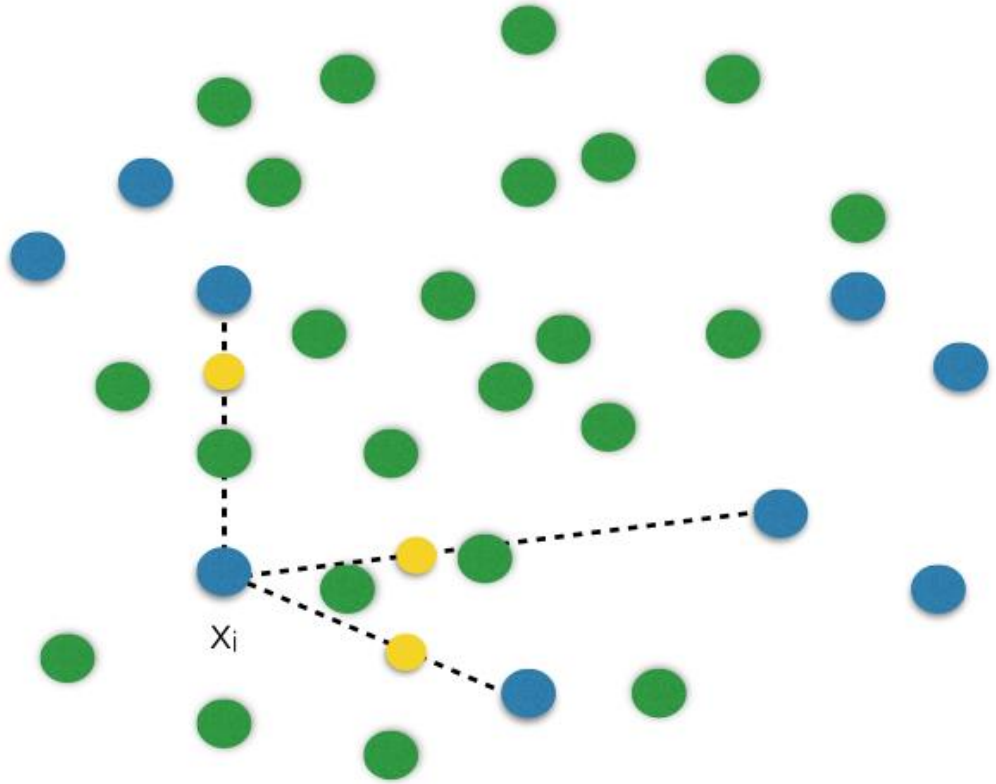


The 3 state-of-the-art resampling (DT)



SMOTE: Overgeneralization Problem

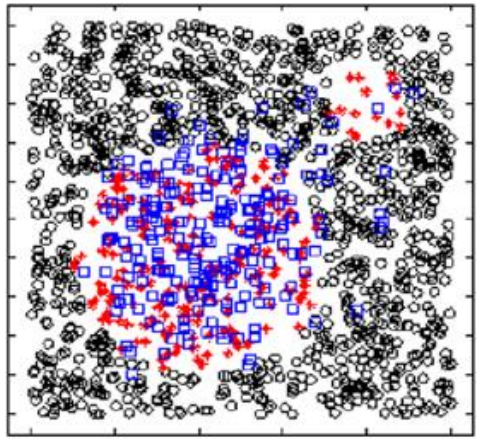




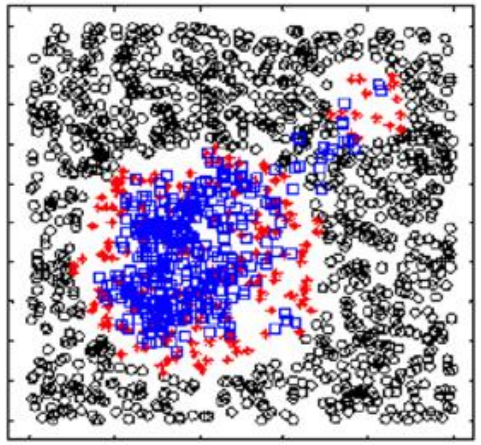
Overgeneralization

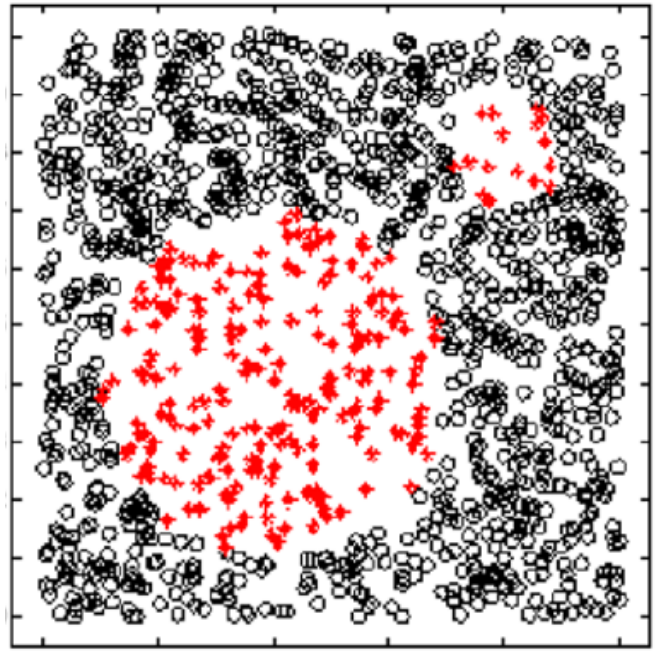
SMOTE: SMOTE variants

1 **SL-SMOTE**



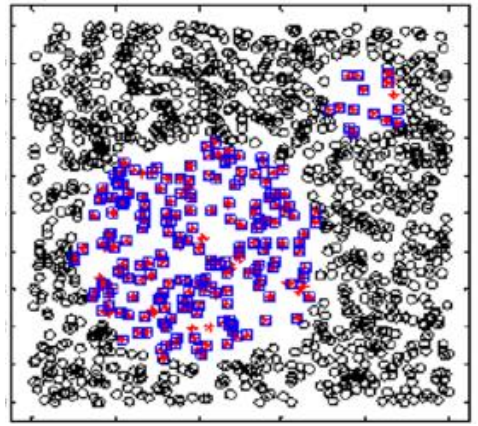
2 **SMOTE**



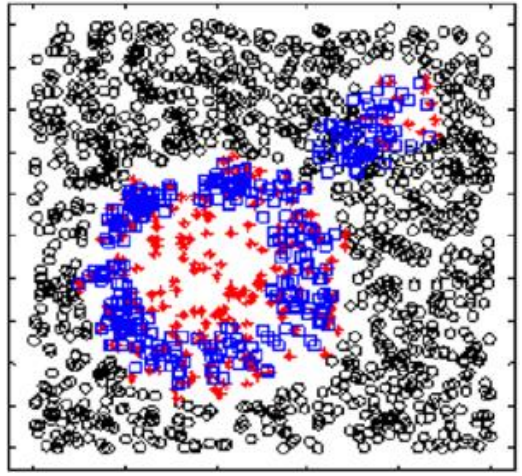


ROS
SMOTE
Safe-Level-SMOTE
Borderline-SMOTE

3 **ROS**



Borderline-SMOTE 4



SMOTE Hybridization: SMOTE + ENN

01

ENN removes any example whose class label differs from the class of at least two of their neighbors

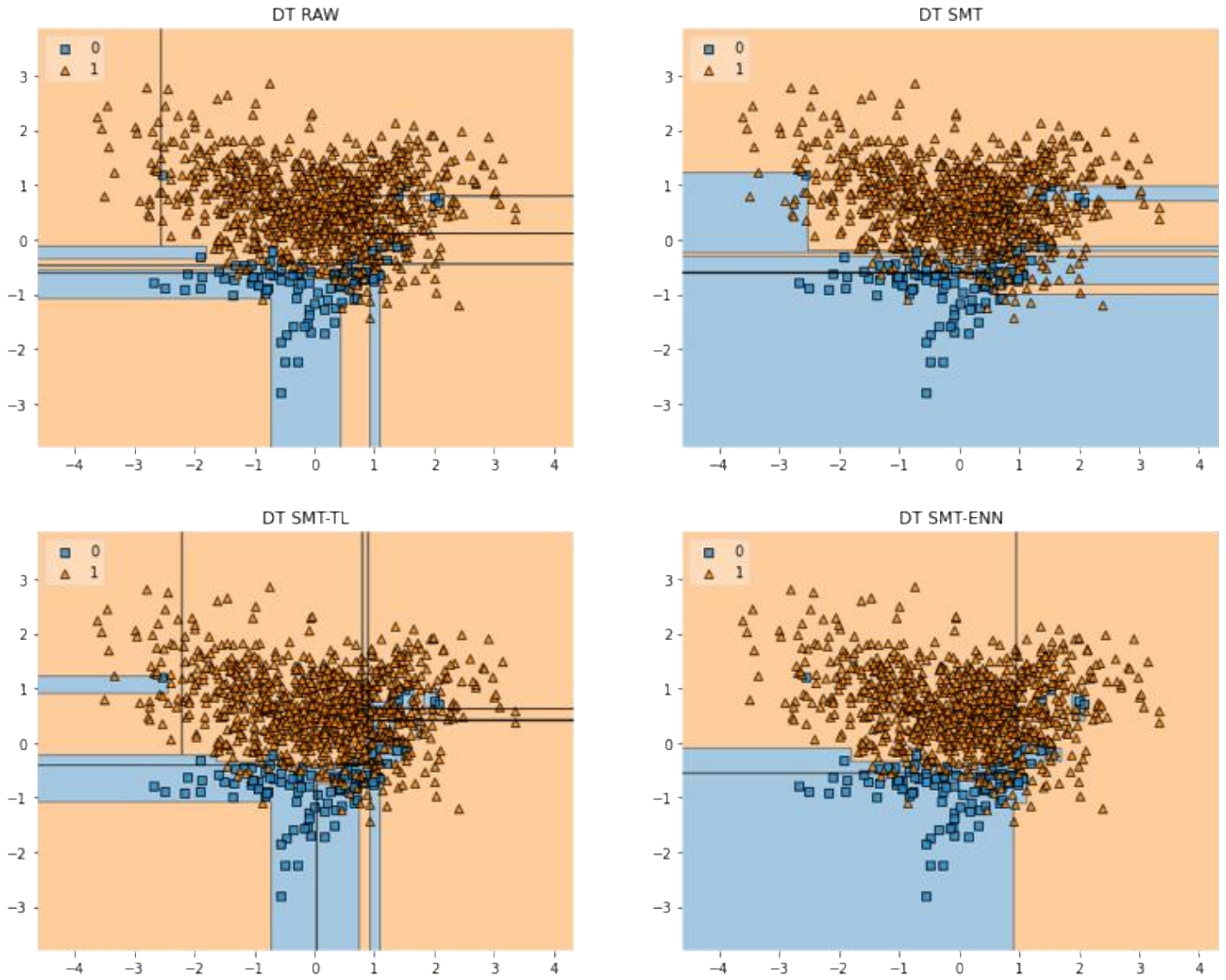
02

ENN remove more examples than the Tomek links does

03

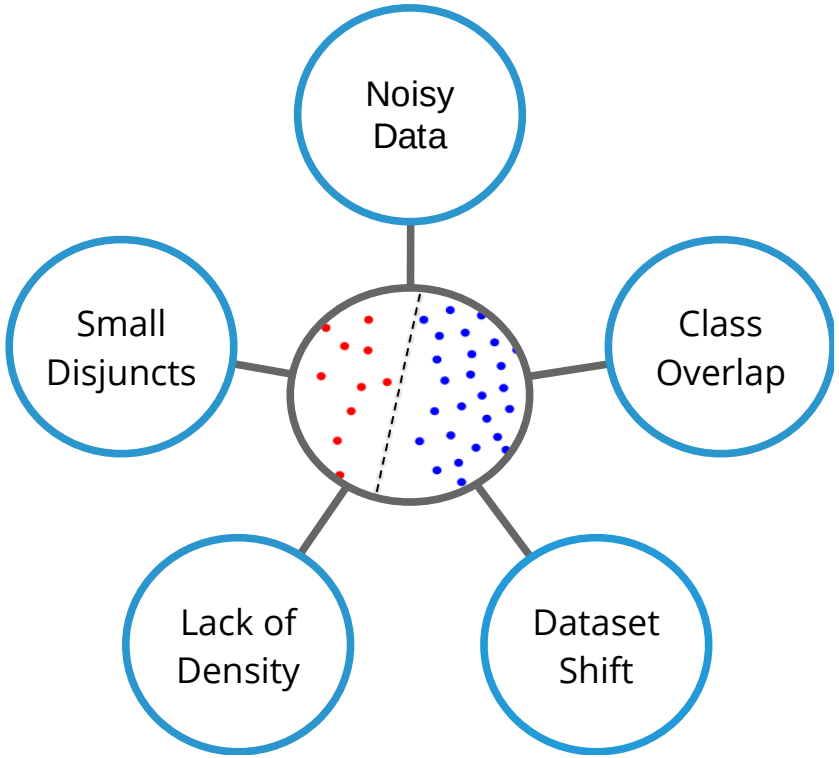
ENN remove examples from both classes

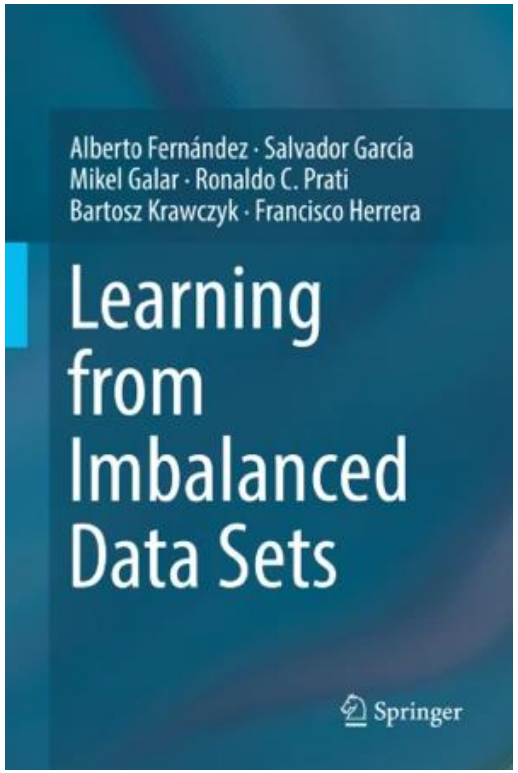
Using different SMOTE Hybridization



Imbalanced Data: interplay with other factors

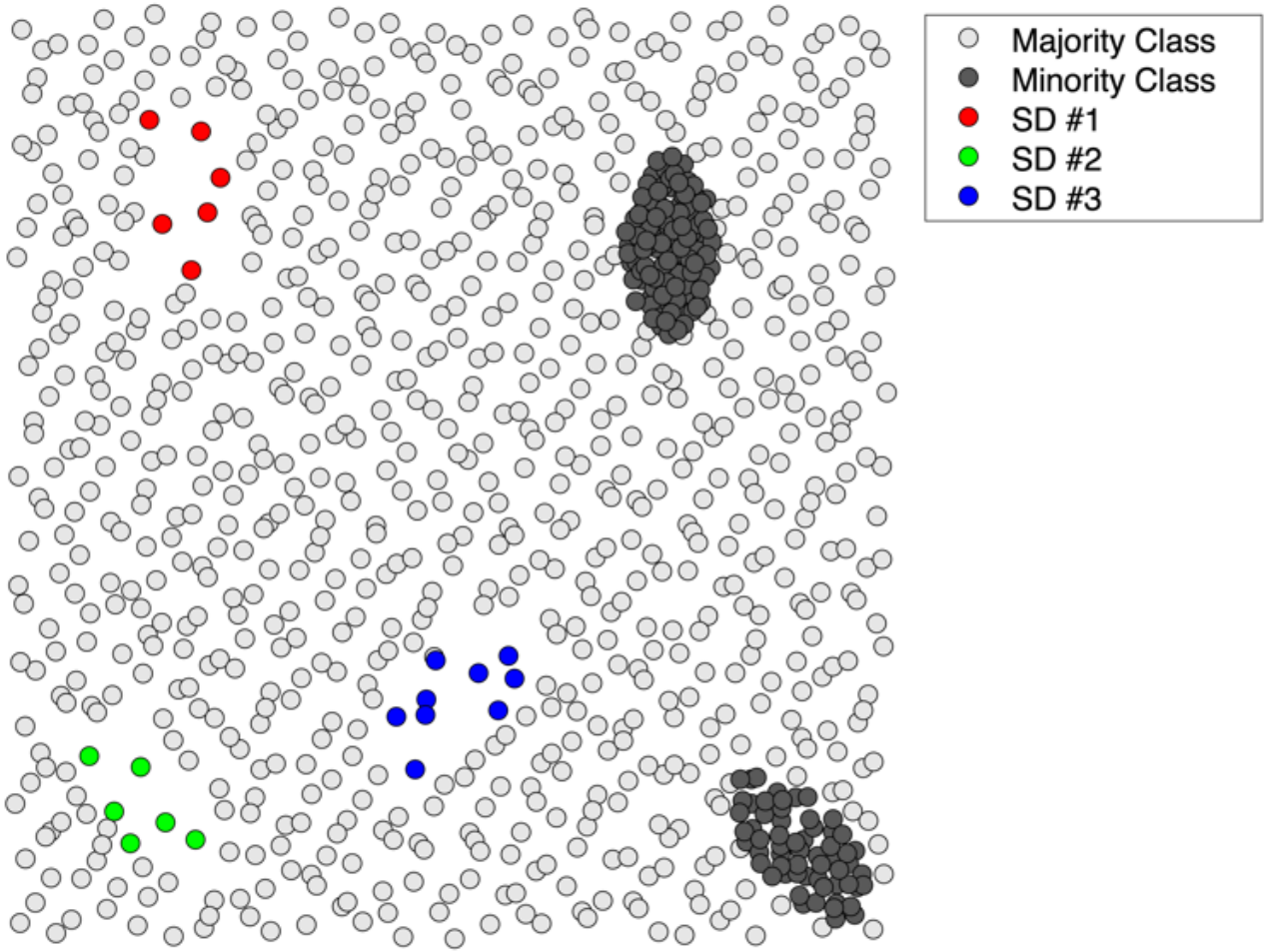
- Although class imbalance is an **important problem in isolation, its combination with other factors** creates a much more difficult setting for classifiers.





- **Its effects are exacerbated** by other *data intrinsic characteristics, irregularities, complexity factors*.

Small Disjuncts: within-class imbalance



- **Definition:** Underrepresented sub-concepts associated with within-class imbalance (*small disjuncts*).
- **Problem:** Classifiers learn by generating rules for larger disjuncts , overfitting smaller disjuncts.
- **Example:** Clusters of patients with the same outcome but distinct characteristics.
- **Ongoing Research:** Distinguishing between rare cases, core concepts, and noise.

Small Disjuncts: CBO – **C**luster-**B**ased **O**versampling

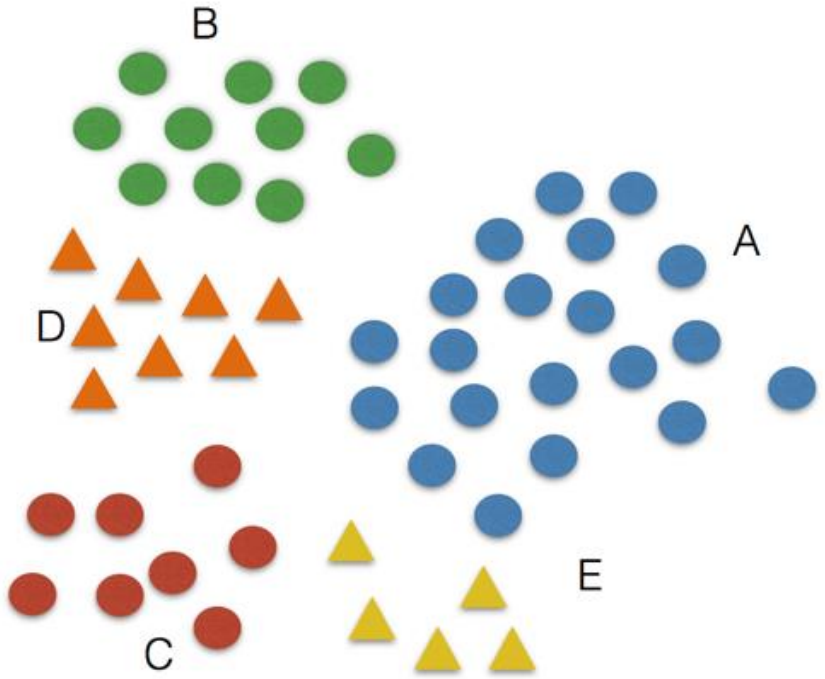
Number of examples in each cluster:

Majority class: A: 20; B: 10; C: 8

$c_{maj} = 3$

Minority class: D: 8; E: 5

$c_{min} = 2$



Number of examples in each cluster:

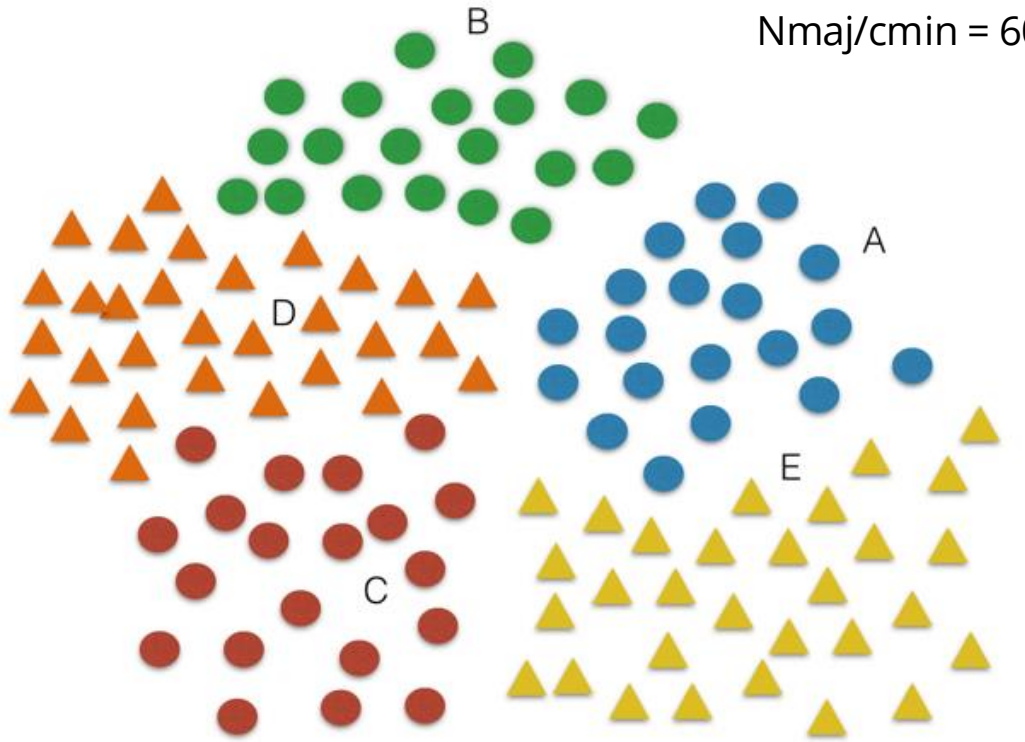
Majority class: A: 20; B: 20; C: 20

$c_{maj} = 3$

Minority class: D: 30; E: 30

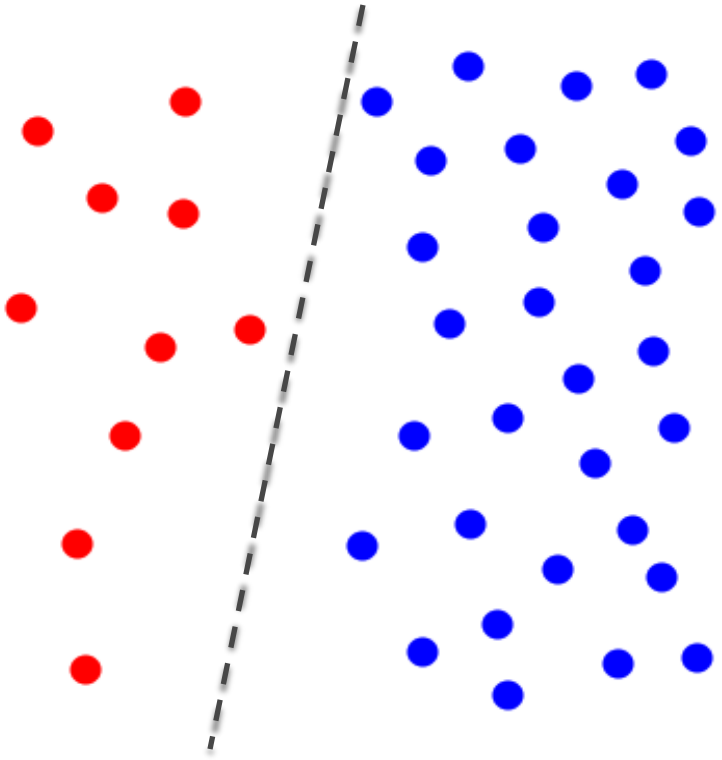
$c_{min} = 2$

Nmaj/cmin = 60/2 = 30

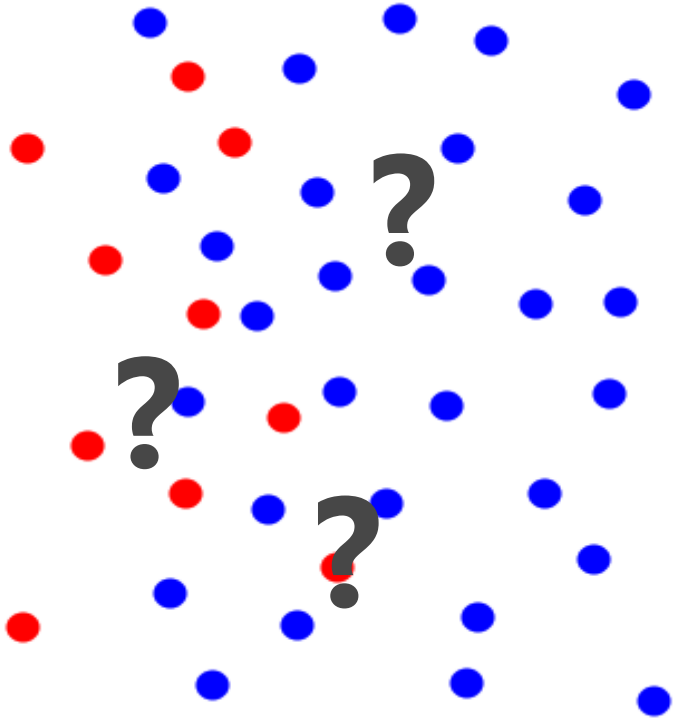


Imbalanced Data: the problem of Class Overlap

- Class Overlap is recognized as **the most harmful issue** for classification, especially in imbalanced domains.

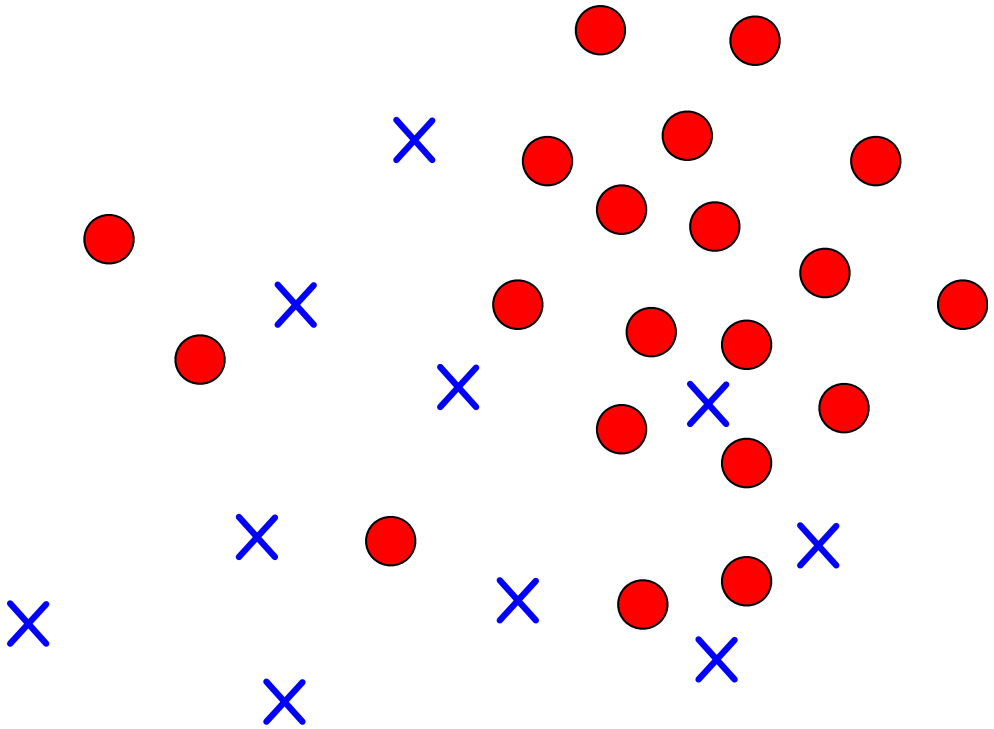


IR = 3:1

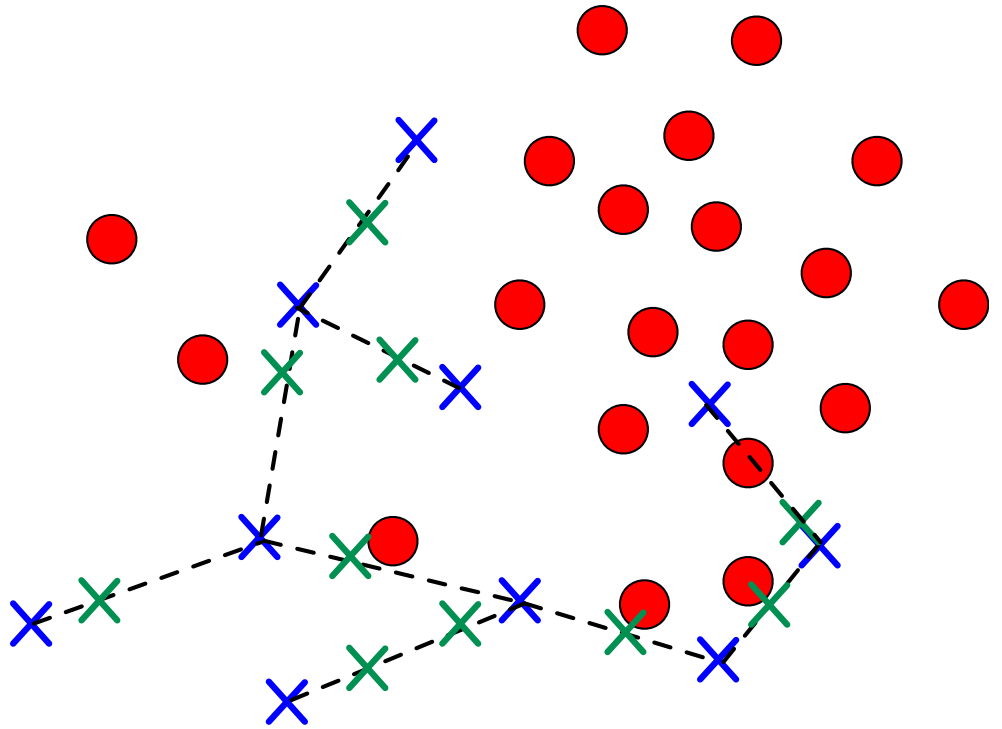


IR = 3:1

Class Overlap: SMOTE-TL and SMOTE-ENN

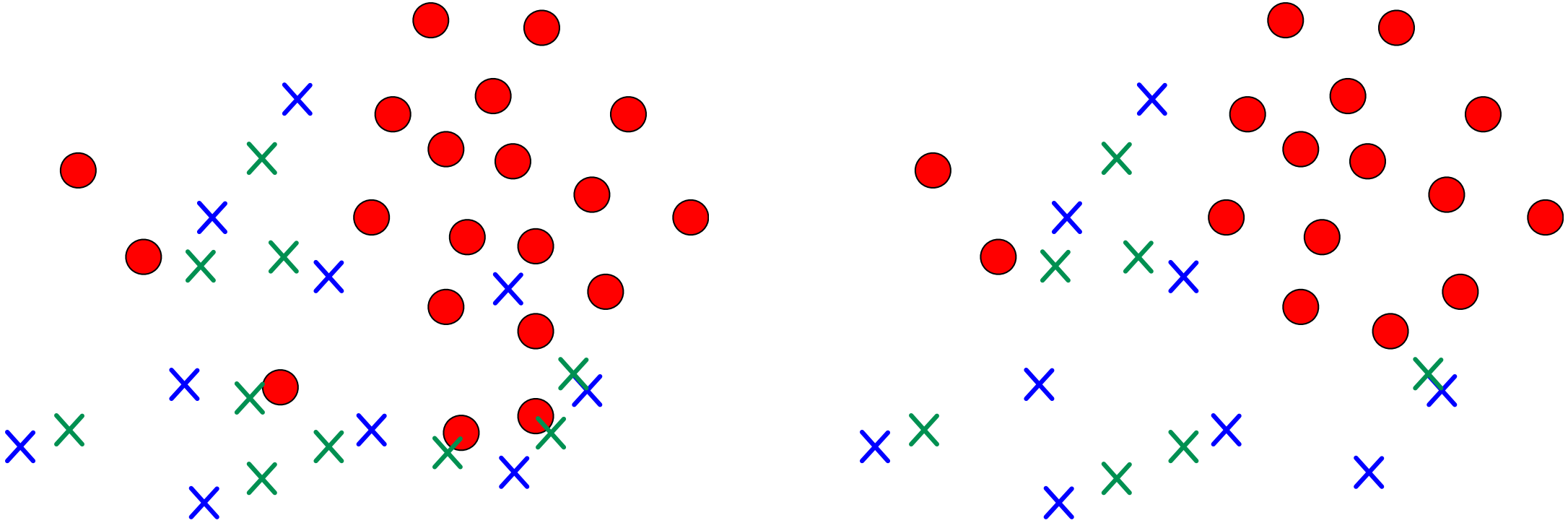


Imbalanced Data



SMOTE

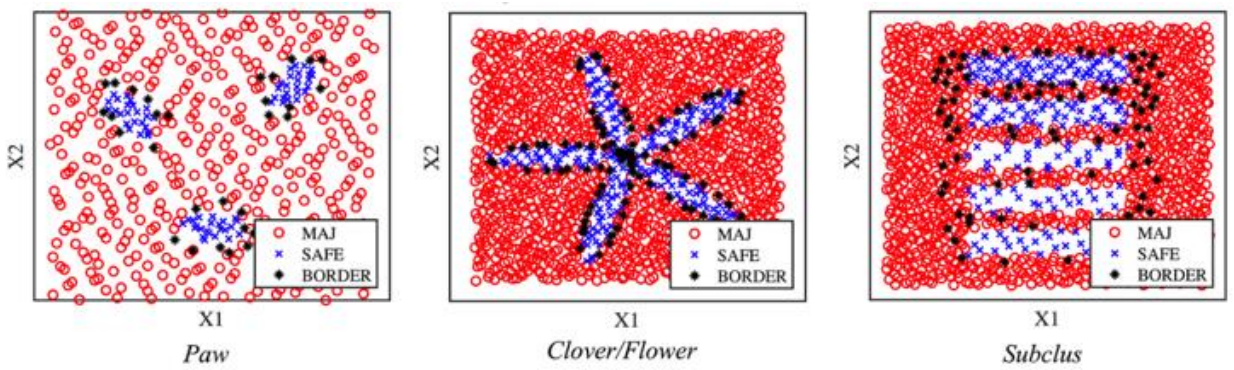
SMOTE-TL: SMOTE + Tomek Links



Tomek Links

SMOTE-TL

Imbalanced Data: Data Difficulty Factors



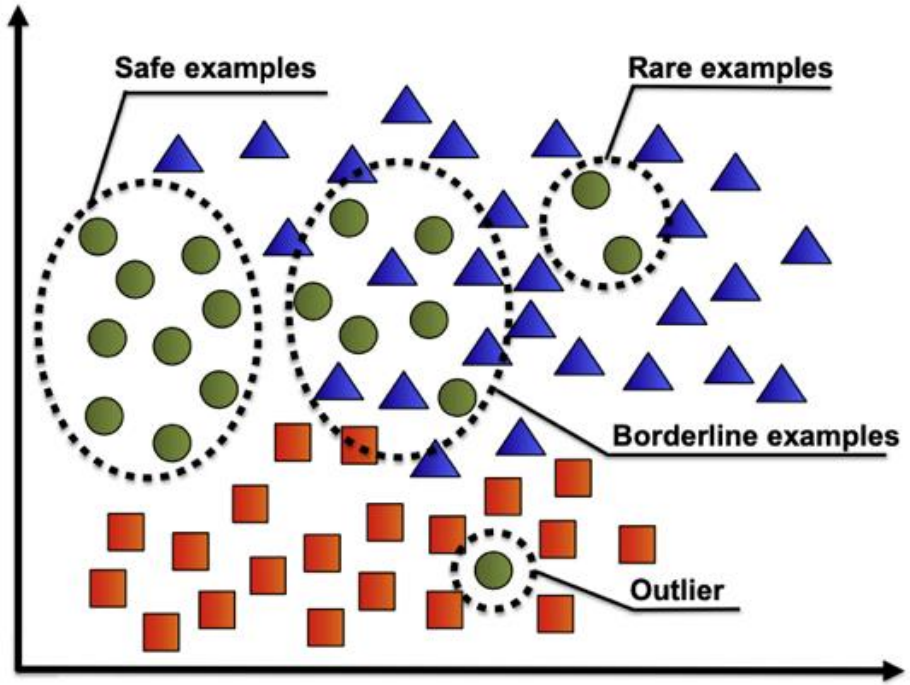


Table 3 Labelling of datasets with respect to minority class examples and k-neighbourhood

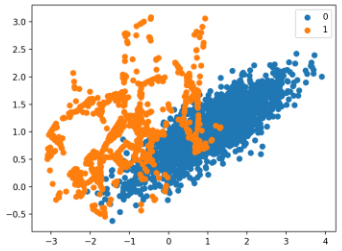
Dataset	S [%]	B [%]	R [%]	O [%]
breast-w	91.29	7.88	0.00	0.83
abdominal-pain	59.90	22.28	8.90	7.92
acl	67.50	30.00	0.00	2.50
new-thyroid	68.57	31.43	0.00	0.00
vehicle	74.37	24.62	0.00	1.01
nursery	82.00	17.00	1.00	0.00
satimage	47.47	39.76	4.58	8.19
car	47.83	39.13	8.70	4.35
scrotal-pain	38.98	45.76	10.17	5.08
ionosphere	44.44	30.95	11.90	12.70
credit-g	9.33	63.67	10.33	16.67
ecoli	28.57	54.29	2.86	14.29
hepatitis	15.63	62.50	6.25	15.63
haberman	4.94	61.73	18.52	14.81
breast-cancer	24.71	25.88	32.94	16.47
cmc	17.72	44.44	18.32	19.52
cleveland	0.00	31.43	17.14	51.43
glass	0.00	35.29	35.29	29.41
hsv	0.00	0.00	28.57	71.43
abalone	8.36	20.60	20.60	50.45
postoperative	0.00	41.67	29.17	29.17
seismic-bumps	3.52	29.41	16.47	50.58
solar-flare	0.00	48.84	11.63	39.53
transfusion	18.54	47.19	11.24	23.03
yeast	5.88	47.06	7.84	39.22
balance-scale	0.00	0.00	8.16	91.84

Algorithmic modifications in imbalanced

Concentrate on modifying existing learners to alleviate their bias towards majority class instead on altering the training set

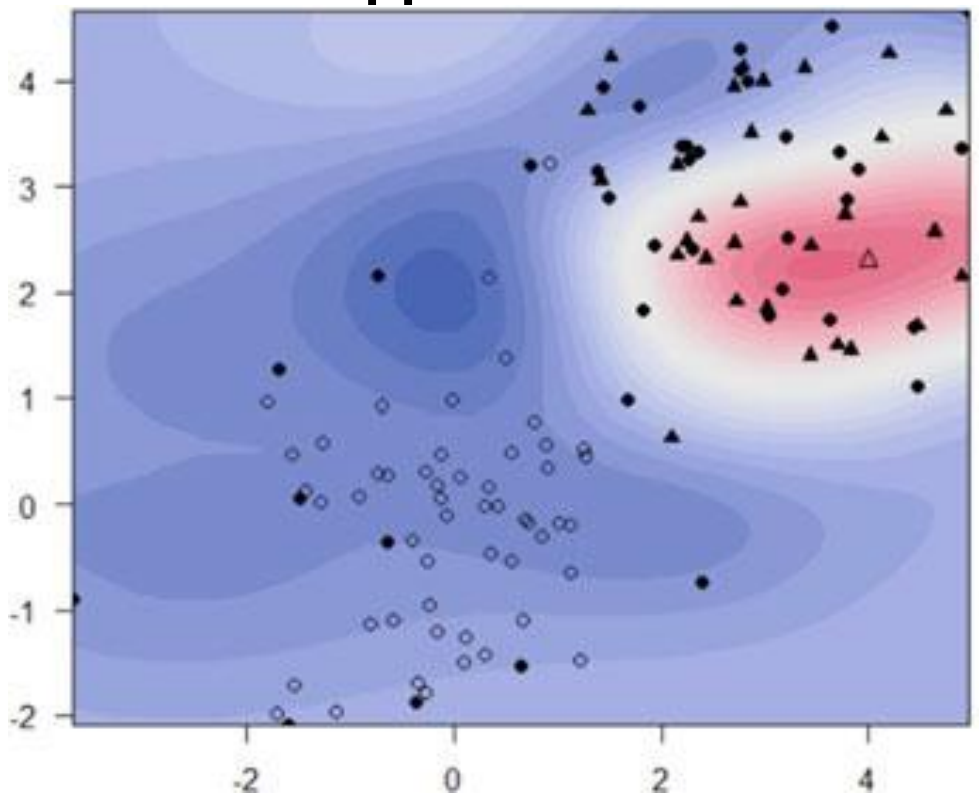
This requires a good insight into the modified learning algorithm and a precise identification of reasons for its failure in mining skewed distributions

This reduces their flexibility, but offers higher specialization potential in tuning the method to the problem at hand

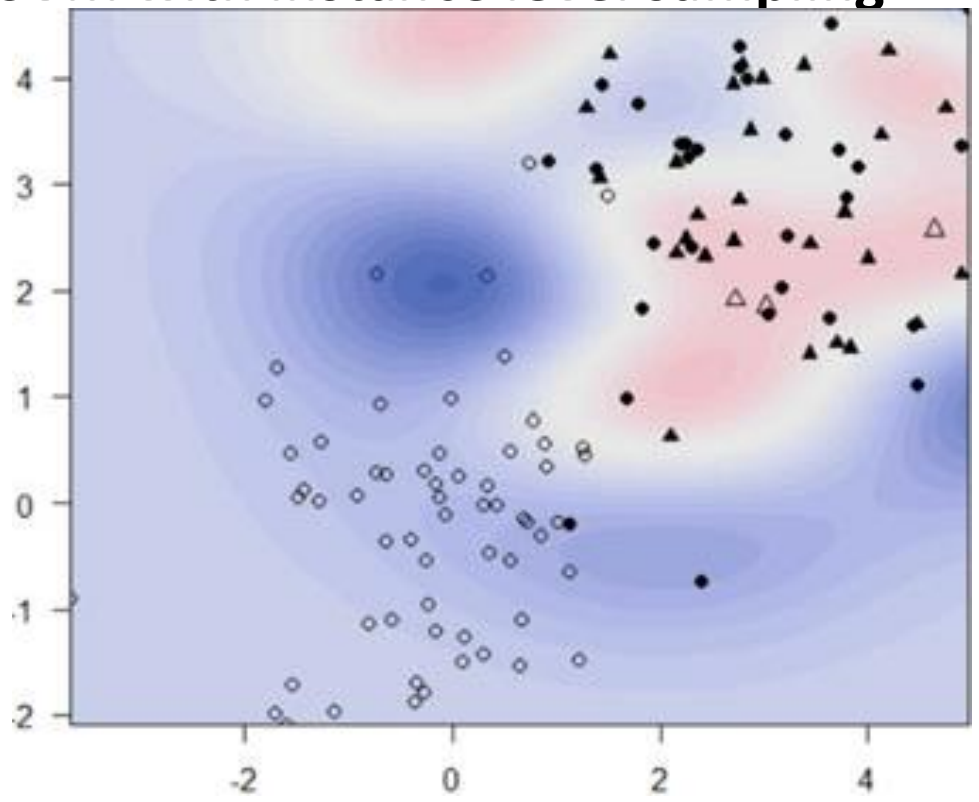


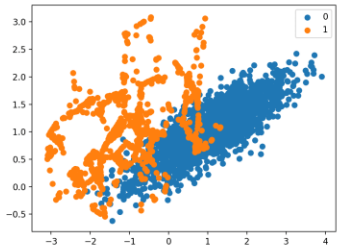
Different decision boundaries

SVM standard approach



SVM with instance level sampling





Taxonomy of algorithmic approaches

Support Vector Machines

- Kernel modification
- Weighted approaches
- Active learning

Decision Trees

- Hellinger distance for splitting

K-Nearest Neighbours

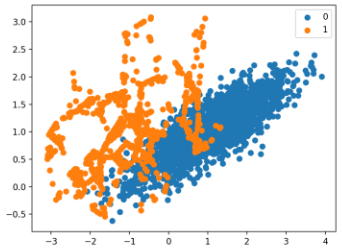
- Gravitation based computation
- Weighted prototypes
- Fuzzy OWA K-NN

Bayesian Classifiers

- Locally weighted NB

One-Class Classifiers

- Training a one-class classifier on the majority class;
- Training a well-tuned one-class classifier on the minority class;
- Training one-class classifiers on both classes and combining their outputs.



Cost-sensitive learning

- Weighting errors made on minority class examples higher than those of the majority class in computing training error:
  - C(+, -) > C(-, +)
  - C(+, +) = C(-, -) = 0
- Needs a cost matrix, which encodes misclassification penalty.
- Consider the cost-matrix throughout the building of the model for achieving the lowest cost.
- However, the cost matrix is often unavailable

	actual negative	actual positive
predict negative	$C(0,0) = c_{00}$	$C(0,1) = c_{01}$
predict positive	$C(1,0) = c_{10}$	$C(1,1) = c_{11}$

	fraudulent	legitimate
refuse	\$20	$-\$20$
approve	$-x$	$0.02x$

C. Elkan, **The foundations of cost-sensitive learning**, in: *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 2001, pp. 973–978

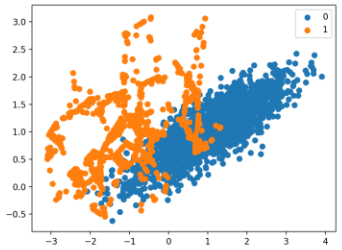
How to obtain cost-matrix

Provided by an expert.

- Supplied data is accompanied by the cost matrix that comes directly from the nature of a problem.
- This usually requires an access to a domain expert that can assess the most realistic cost values, i.e. credit card fraud detection

Estimated using training data.

- No information on cost matrix available during training:
  - Heuristic setting of cost values: IR for cost estimation
  - Learning from training data: Thresholding via validation set



Cost-sensitive learning

Direct methods:

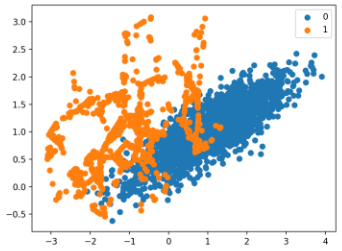
Introduce and utilize misclassification costs into the learning algorithms.

Meta-learning:

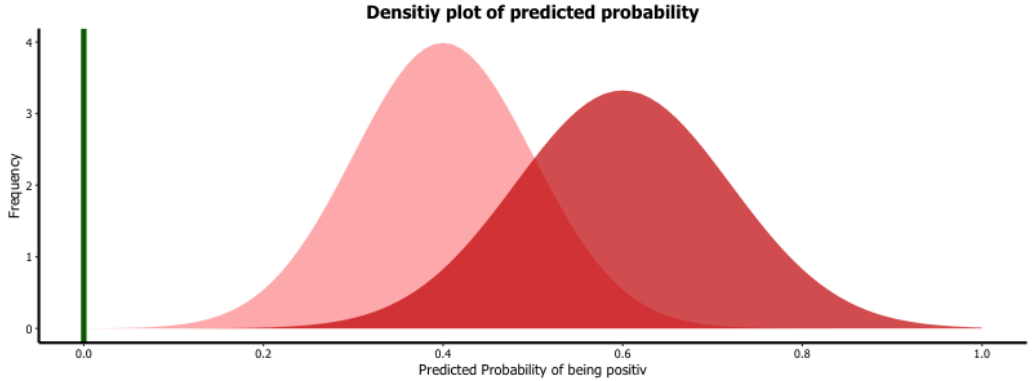
"Preprocessing" mechanism for the training data or a "post-processing" of the output.

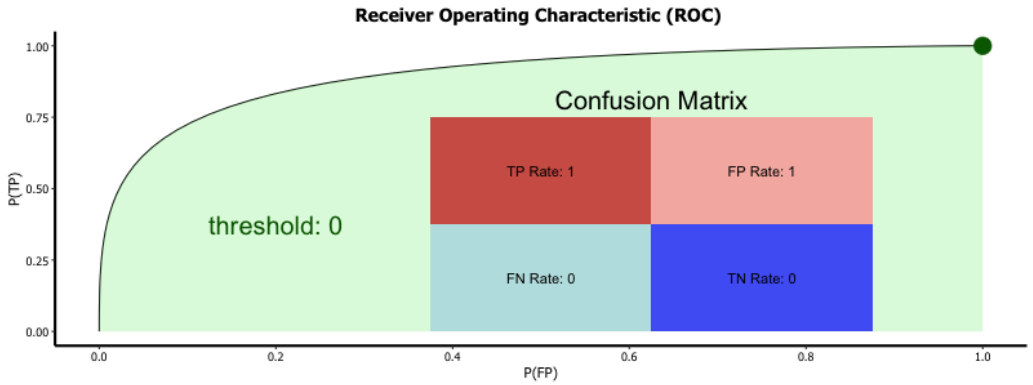
The original learning algorithm is not modified:

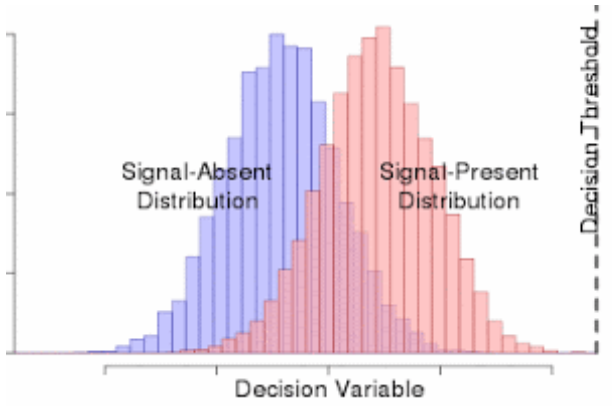
- Sampling: assigning instance weights
- Thresholding based on the Bayes decision theory: assign instances to class with minimum expected cost.

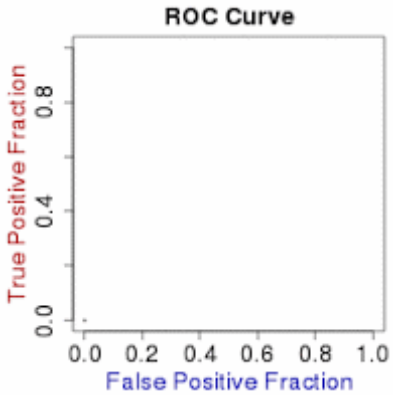


Thresholding on positive class and AUC





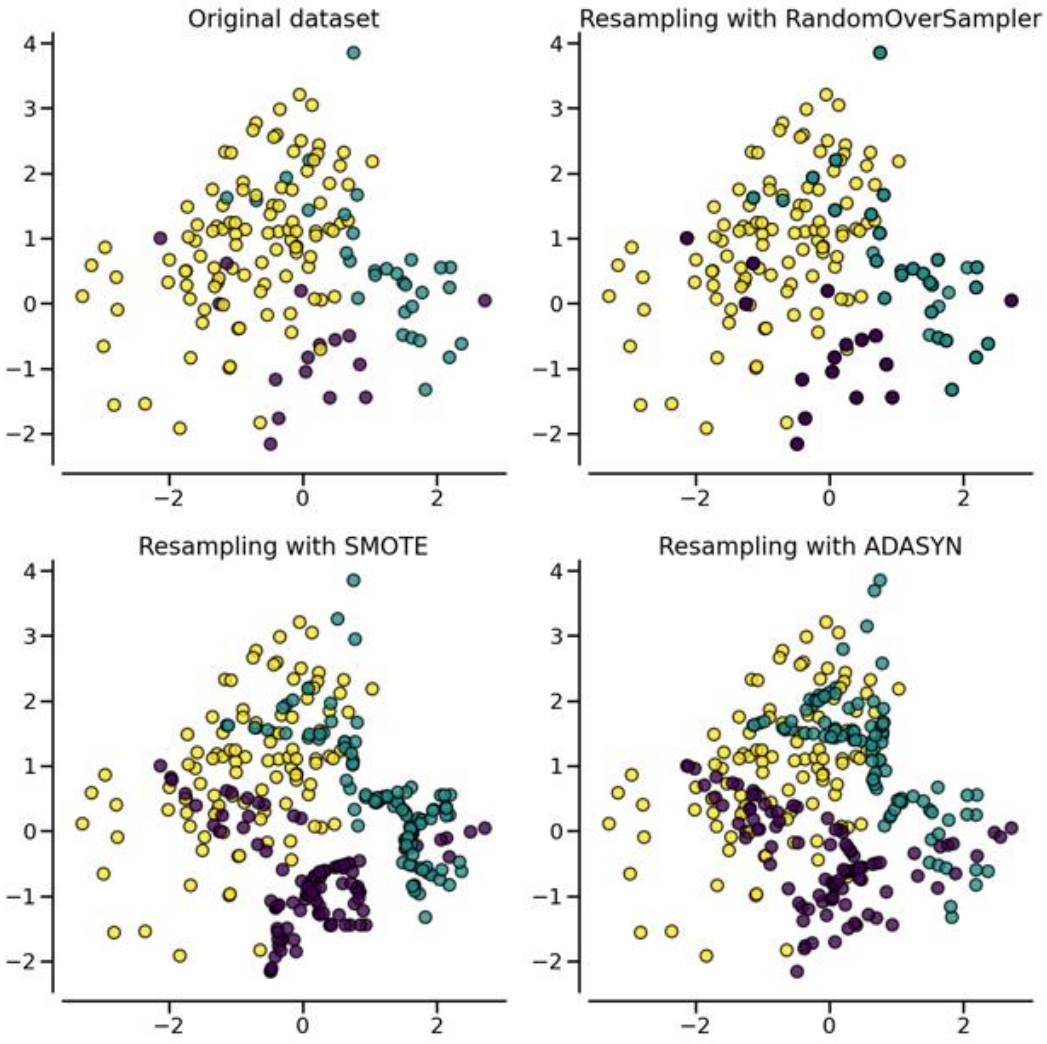




Outline

1. Introduction: Definition, properties and difficulty
2. Evaluation metrics
3. Addressing imbalanced datasets
4. **Software tools for classification with imbalanced data**
5. Final Comments and Surveys for a deeper study

imbalanced-learn: Tackle the Curse of Imbalanced Datasets in ML



- Implements several strategies to overcome the problem of imbalanced learning.

```
>>> from imblearn.over_sampling import SMOTE, ADASYN
>>> X_resampled, y_resampled = SMOTE().fit_resample(X, y)
>>> print(sorted(Counter(y_resampled).items()))
[(0, 4674), (1, 4674), (2, 4674)]
>>> clf_smote = LogisticRegression().fit(X_resampled, y_resampled)
>>> X_resampled, y_resampled = ADASYN().fit_resample(X, y)
>>> print(sorted(Counter(y_resampled).items()))
[(0, 4673), (1, 4662), (2, 4674)]
>>> clf_adasyn = LogisticRegression().fit(X_resampled, y_resampled)
```

https://imbalanced-learn.org/stable/

pip install imbalanced-learn

Python libraries: imbalanced-learn

- Dependant of Scikit-Learn
- A large number of preprocessing techniques
- Include ensemble learning
- Specific performance metrics
- Imbalanced Datasets

Preprocessing	*Technique*
Under-Sampling	Random majority under-sampling with replacement Extraction of majority-minority Tomek links Under-sampling with Cluster Centroids NearMiss-(1 & 2 & 3) Condensend Nearest Neighbour One-Sided Selection Neighboorhood Cleaning Rule Edited Nearest Neighbours Instance Hardness Threshold Repeated Edited Nearest Neighbours AllKNN
Over-Sampling	Random majority over-sampling with replacement SMOTE - Synthetic Minority Over-sampling Technique bSMOTE(1 & 2) - Borderline SMOTE of types 1 and 2 SVM SMOTE - Support Vectors SMOTE ADASYN - Adaptive synthetic sampling approach for imbalanced learning
Hybrid sampling	SMOTE + TomekLinks SMOTE + ENN
Ensemble sampling	EasyEnsemble BalanceCascade

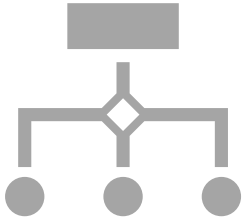
Python libraries: imbalanced-learn

Sampler class implements 3 main methods from the API:

fit computes statistics needed to resample the data;

resample performs the sampling with the desired balancing ratio;

fit_resample is equivalent to calling both methods directly.

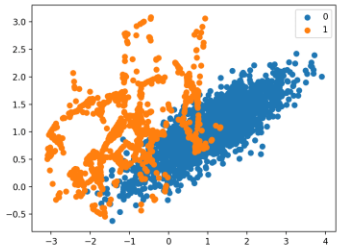


Input data must be in dataFrame or numpy structure.

from imblearn.over_sampling **import** RandomOverSampler

ros = RandomOverSampler(random_state=0)

X_resampled, y_resampled = ros.**fit_resample**(X, y)



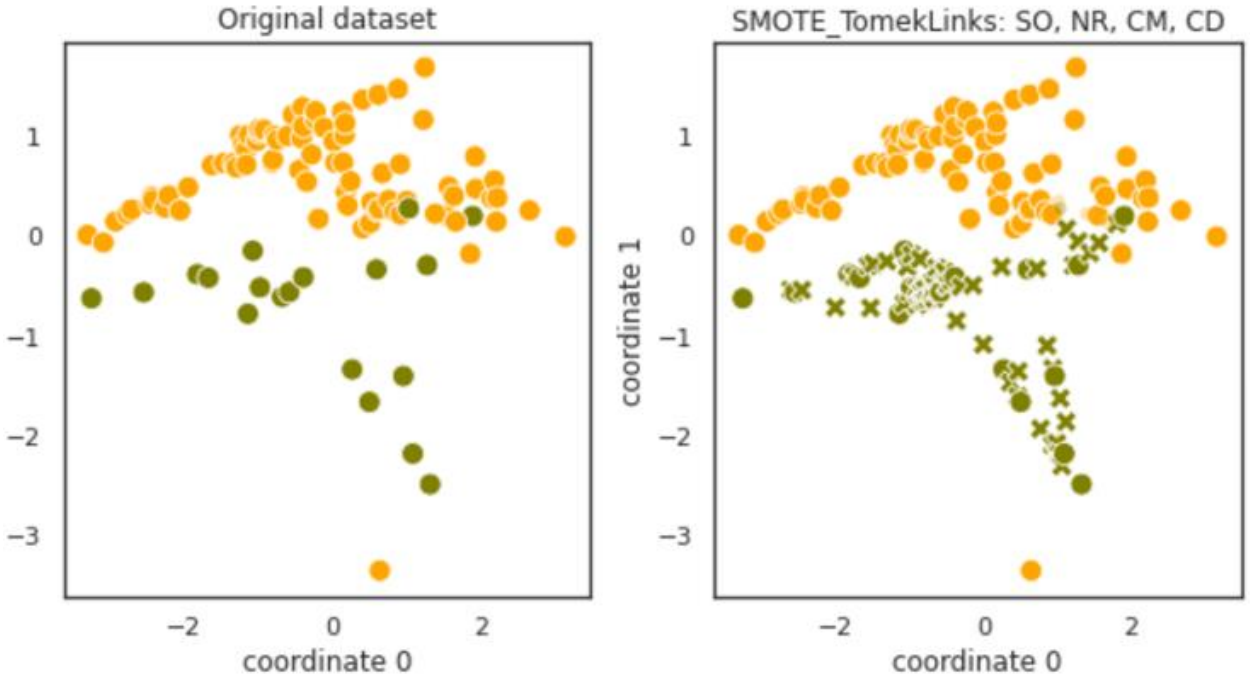
Python libraries: imbalanced-learn

- Hybridizations in preprocessing are also included:

```
from imblearn.combine import SMOTEENN
from imblearn.combine import SMOTETomek
smote_enn = SMOTEENN(random_state=0)
X_new, y_new = smote_enn.fit_resample(X, y)
```

- Class **Pipeline**:
  - Inherited from the `scikit-learn` toolbox to automatically combine samplers, transformers, and estimators.
- State-of-the-art metrics to evaluate the imbalanced learning problem: `module imblearn.metrics`
  - Recall, specificity, f-measure (F1), geometric mean, Index of Balanced Accuracy (IBA), and support.

smote-variants: A collection of 85 SMOTE variants for oversampling



- Provides a Python implementation of 85 oversampling techniques to boost the application and development in the field of imbalanced learning.

```python
import smote_variants as sv

oversampler= sv.SMOTE_ENN()

# supposing that X and y contain some the feature and target data of some dataset
X_samp, y_samp= oversampler.sample(X, y)
```

https://smote-variants.readthedocs.io

pip install smote-variants

Gyorgy Kovacs (2019), Smote-variants: A python implementation of 85 minority oversampling techniques. Neurocomputing.

Outline

1. Introduction: Definition, properties and difficulty
2. Evaluation metrics
3. Addressing imbalanced datasets
4. Software tools for classification with imbalanced data
5. **Final Comments and Surveys for a deeper study**

Imbalanced Learning: CheatSheet

1

Apply a standard battery of algorithms to raw problem, in order to know the base behaviour: kNN, DT, SVM.

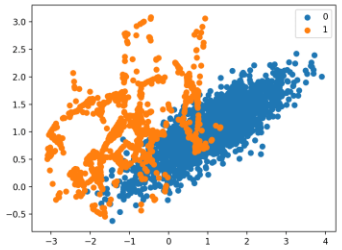
2

Observe the ROC or PR curve (AUC / AP) in case there is some threshold that allows a balance between positive and negative hits, within the requirements of the case study.

3

In case the values obtained by the quality metrics are not sufficient, apply one of the following solutions:

- Undersampling
- Oversampling
- Cost-Sensitive



CheatSheeet: Solutions

Undersampling applied in case:

- High sensitivity is desired: **RUS**
- There may be noise in the set: **RUS, CNN, TL**
- There are a high number of negative examples: **RUS**
- There is a need to reduce the learning time: **RUS, Class Weights**
- High Dimensionality: **RUS, Class Weights**

Oversampling applied in case:

- A good balance to be kept for TPR, TNR: **ROS**
- There may be subclusters of the positive class: **SMOTE**
- Positive class reinforcement needed in overlapping areas: **SMOTE**
- Few data samples: **SMOTE**
- High Dimensionality: **ROS, Class Weights**

CheatSheet: Final comments

It is quite convenient to analyze the ROC curve to find adequate probability thresholds: a posteriori approach

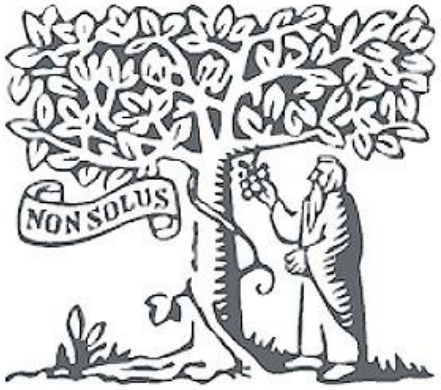
Hyperparametrization: find optimal values

the number of neighbors (k) in kNN,	the pruning in a decision tree,	the kernel values in an SVM…

Ensembles-based techniques are very powerful, best if these are used in synergy with preprocessing:

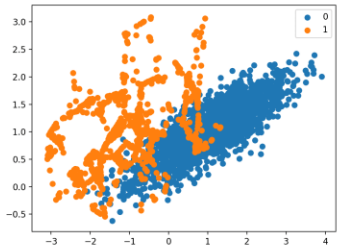
RUSBoost	SMOTEBagging.

SPECIALIZED LITERATURE

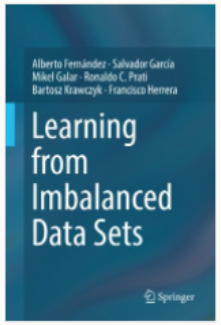


Scopus

Nice surveys to gain more insight on the topic



The Imbalanced Learning Book



Learning from Imbalanced Data Sets

Authors (view affiliations)

Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk,

Francisco Herrera

Offers a comprehensive review of imbalanced learning widely used worldwide in many real applications, such as fraud detection, disease diagnosis, etc

Provides the user with the required background and software tools needed to deal with Imbalance data

Presents the latest advances in the field of learning with imbalanced data, including Big Data applications and non-classical problems, such as semi-supervised learning, multilabel and multi instance learning, and ordinal classification and regression

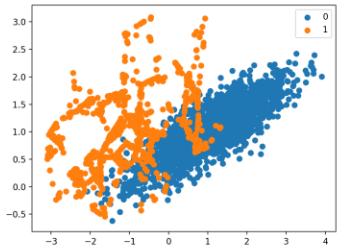
Includes case studies

Book

172 Citations | 12 Mentions | 43k Downloads

Table of contents (14 chapters) | About this book

Search within book



Reviews on Imbalanced Classification

2004:

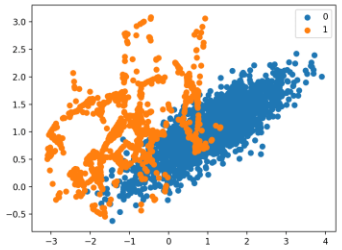
- A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data (Gustavo E. A. P. A. Batista; Ronaldo C. Prati; Maria Carolina Monard)

2009:

- Learning from Imbalanced Data (Haibo He, and Edwardo A. Garcia)

2009:

- Classification of Imbalanced Data: A Review (Yanmin Sun, Andrew K. C. Wong, Mohamed S. Kamel)



Reviews on Imbalanced Classification

2012:

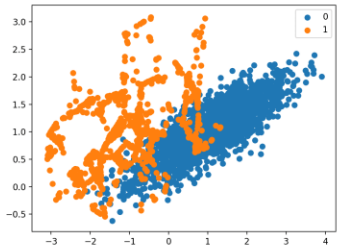
- Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics (Victoria López, Alberto Fernández, Jose G. Moreno-Torres, Francisco Herrera)

2013:

- An insight into classification with imbalanced data: Empirical results and current trends on using **data intrinsic characteristics** (Victoria López, Alberto Fernández, Salvador García, Vasile Palade, Francisco Herrera)

2015:

- Class imbalance revisited: a new experimental setup to assess the performance of treatment methods (Ronaldo C. Prati, Gustavo E. A. P. A. Batista, Diego F. Silva)



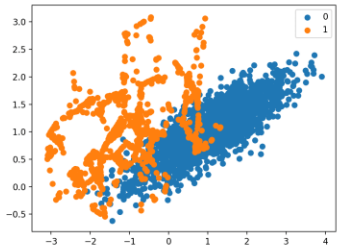
Reviews on Imbalanced Classification (2016-2019)

A **Survey** of Predictive Modeling on Imbalanced Domains (Paula Branco, Luis Torgo, Rita P. Ribeiro)

Dealing with Data Difficulty Factors While Learning from Imbalanced Data (Jerzy Stefanowski)

Learning from class-imbalanced data: **Review of methods and applications** (Guo Haixiang, Li Yijing, Jennifer Shang , Gu Mingyun, Huang Yuanyue, Gong Bing)

Learning from imbalanced data: **open challenges and future directions** (Bartosz Krawczyk)



Recent Works (2020-2022)

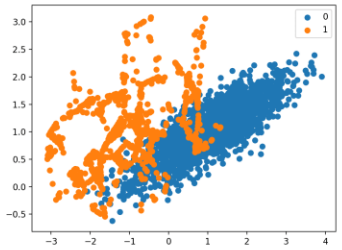
A review on classification of imbalanced data for wireless sensor networks (Patel, H.; Singh Rajput, D.;…;Kashif Bashir, A.; Jo, O.)

A review of methods for imbalanced multi-label classification(Tarekegn, A.N.; Giacobini, M.; Michalak, K.)

The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art (Susan, S. ; Kumar, A.)

An Ensemble Model for Fake Online Review Detection Based on Data Resampling, Feature Pruning, and Parameter Optimization

(Yao, J.; Zheng, Y.; Jiang, H.)



Latest Works (2022-2024)

Granular Ball Sampling for Noisy Label Classification or Imbalanced Classification (Xia, S., Zheng, S., Wang, G., Gao, X., Wang, B..)

Imbalanced data classification: Using transfer learning and active sampling (Liu, Y., Yang, G., Qiao, S., ...Yuan, G., Peng, Y.)

Noise-robust oversampling for imbalanced data classification (Liu, Y., Liu, Y., Yu, B.X.B., Zhong, S., Hu, Z.)

A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research (Santos, M.S., Abreu, P.H., Japkowicz, N., Fernández, A., Santos, J.)

THANK YOU VERY MUCH FOR YOUR ATTENTION

TO CONTACT ME ON FURTHER QUESTIONS:

Main office

ETSIIT, Dpt. CCIA, D16
C/Periodista Daniel Saucedo
18014. Granada

Alternative office

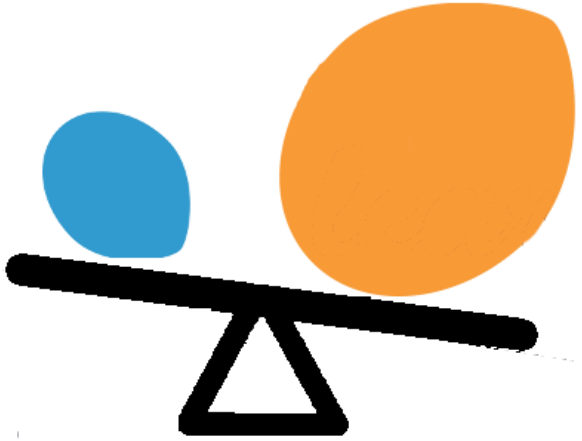
Edificio Mecenas
Facultad de Ciencias
18003. Granada

Work-phone

(+34) 958 240 079

Email / Web

alberto@decsai.ugr.es
https://www.dasci.es



Imbalanced learn

Imbalanced Classification: Fundamentals, Solutions and More

Master in Data Science and Computer Engineering

Alberto Fernández – DASCI Institue. University of Granada