

Artificial Intelligence and Society

Module 03: Imbalanced Data

Miriam Seoane Santos

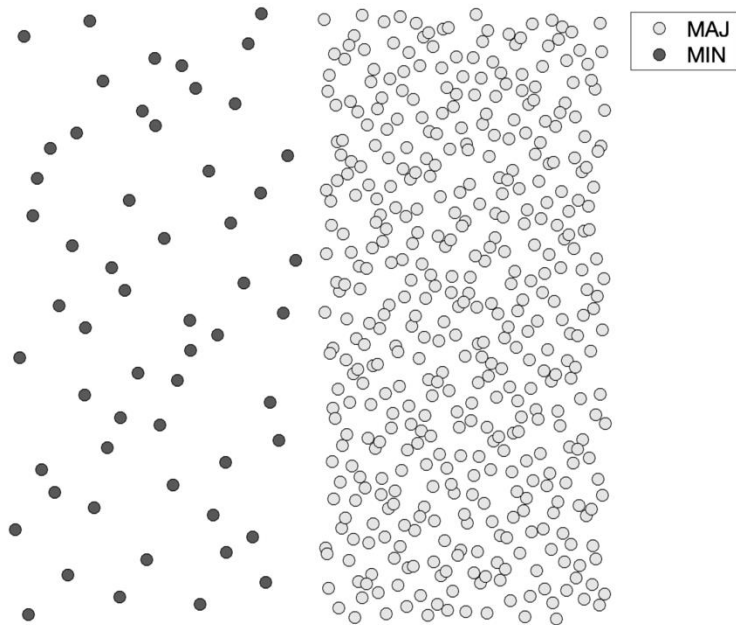
LIAAD, INESC TEC, FCUP, University of Porto

miriam.santos@fc.up.pt

Previously...

Data Complexity Measures: Class Imbalance Measures

- **Class Imbalance Measures:** Consider the ratio of the number of examples between classes.



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

$$C2 = 1 - \frac{1}{IR}$$

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

Class Imbalance

Definition, Applications, Impact, and Mitigation Strategies

Class Imbalance “in the wild”

- Due to **biased sampling** (e.g., representation bias) or the **intrinsic nature** of the domain.
- Beyond research challenges, class imbalance raises many critical questions in real-world applications, from **fraud detection, churn prediction, disease diagnosis, spam detection, sentiment analysis...**

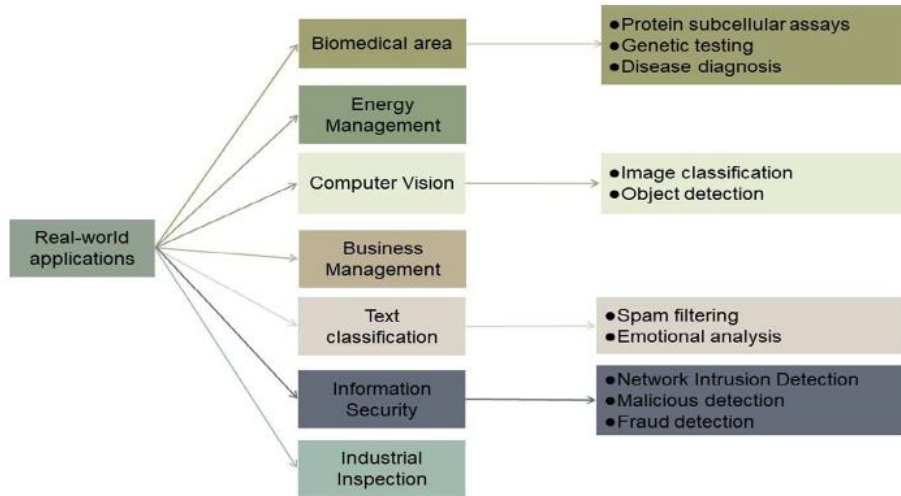
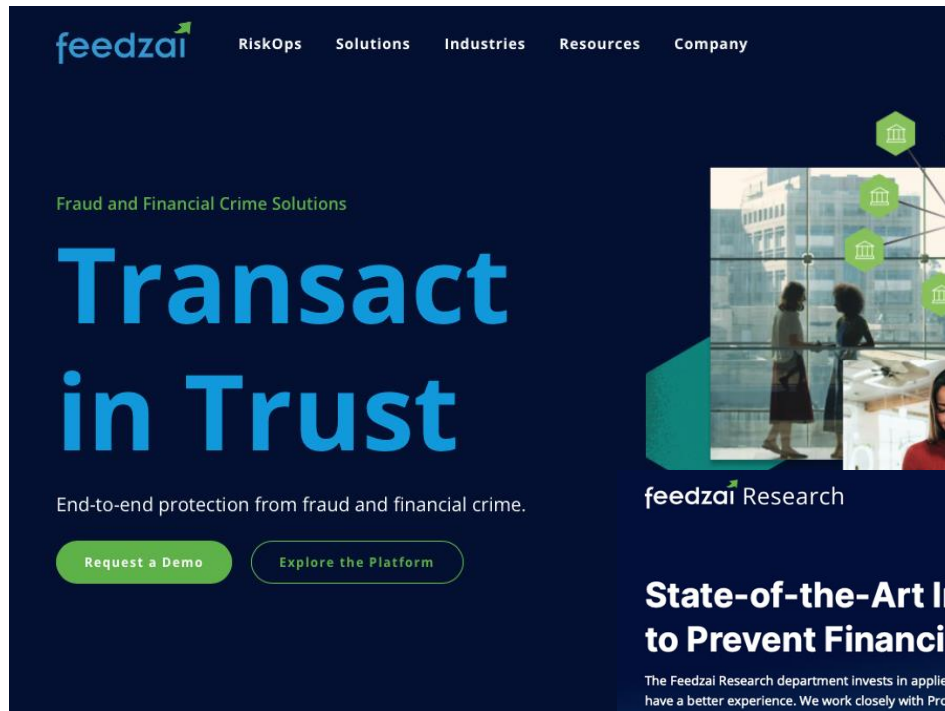


Fig.7 Real-world Application classification

- **COVID infection:** among all patients, only 10% might have COVID
- **Fraud Detection:** fraudulent transactions might make up 0.2% of all transactions
- **Manufacturing defect:** different types of defects have different prevalence
- **Self-driving cars:** objects have different prevalence (cars, trucks, pedestrians)

Class Imbalance in the industry landscape



feedzai

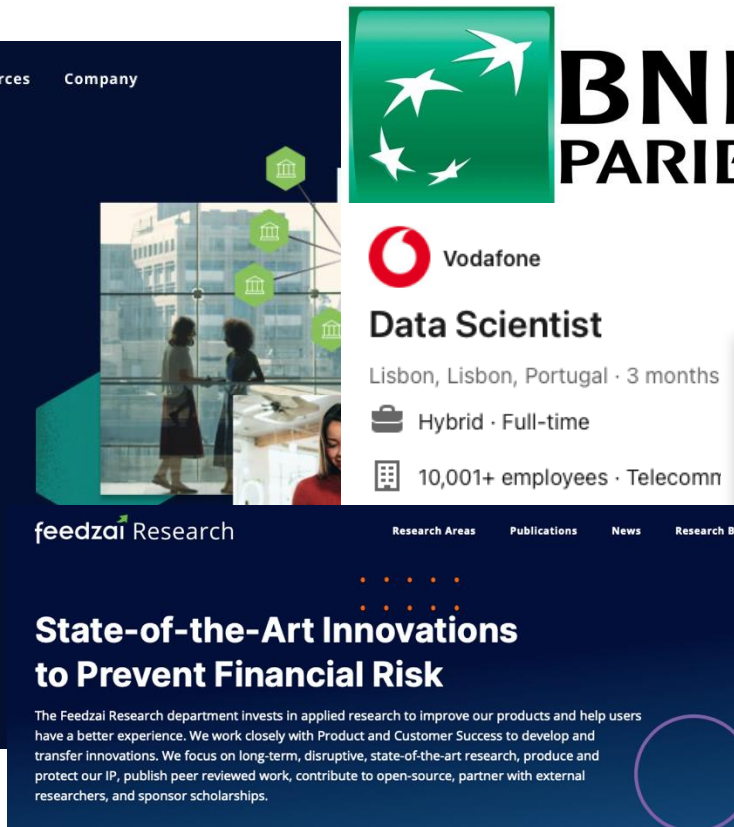
RiskOps Solutions Industries Resources Company

Fraud and Financial Crime Solutions

Transact in Trust

End-to-end protection from fraud and financial crime.

[Request a Demo](#) [Explore the Platform](#)



feedzai Research

Research Areas Publications News Research B

State-of-the-Art Innovations to Prevent Financial Risk

The Feedzai Research department invests in applied research to improve our products and help users have a better experience. We work closely with Product and Customer Success to develop and transfer innovations. We focus on long-term, disruptive, state-of-the-art research, produce and protect our IP, publish peer reviewed work, contribute to open-source, partner with external researchers, and sponsor scholarships.



**BNP
PARIBAS**



Data Scientist

Lisbon, Lisbon, Portugal · 3 months

Hybrid · Full-time

10,001+ employees · Telecomm

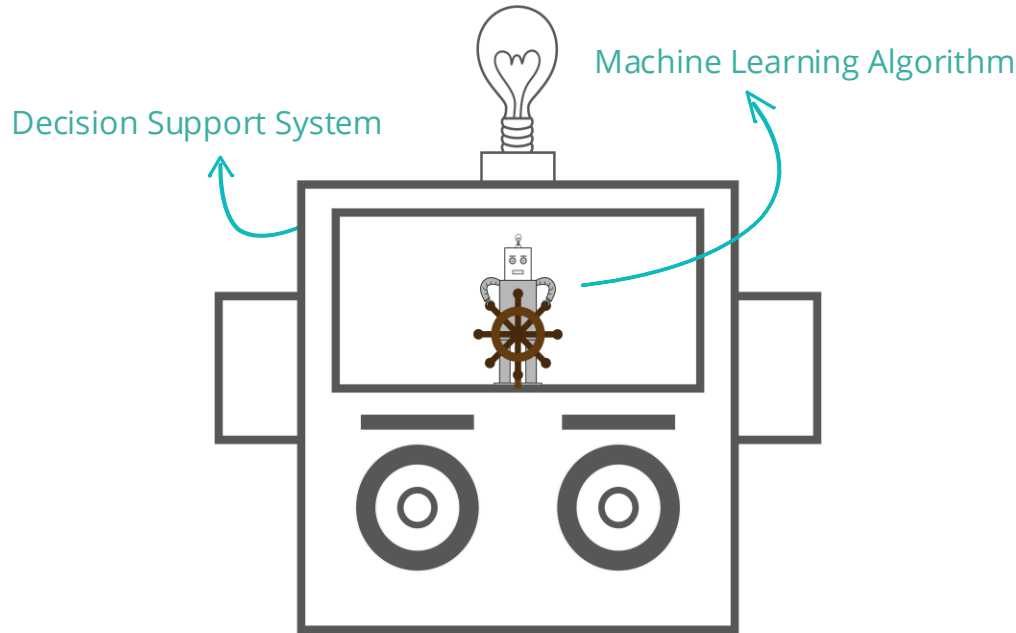


Data Science Trainee (M/F) - Porto

[Estágios](#)

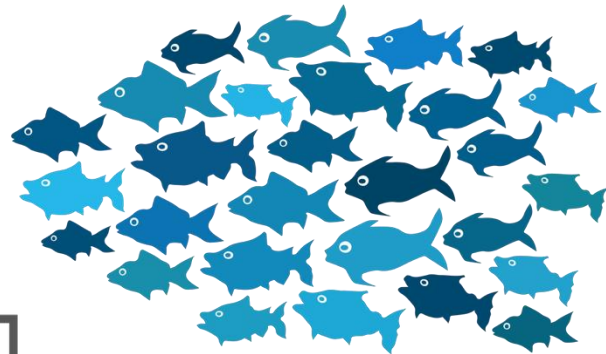
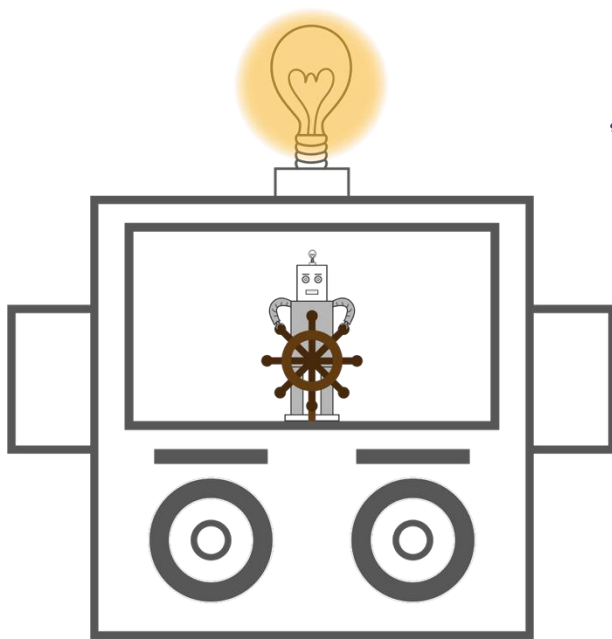
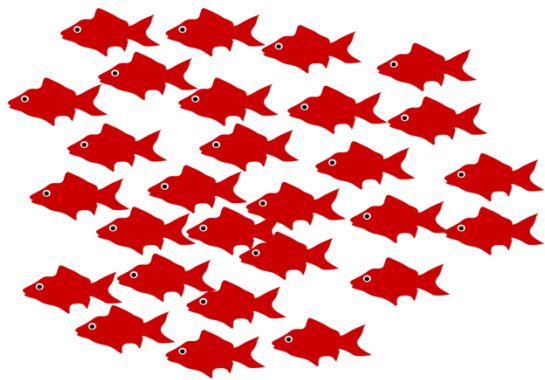
Imbalanced Data: Problem

- Most of the traditional pattern classifiers assume their **input data to be well-behaved** across a set of characteristics. Practical datasets, however, show up with **various forms of irregularities**.



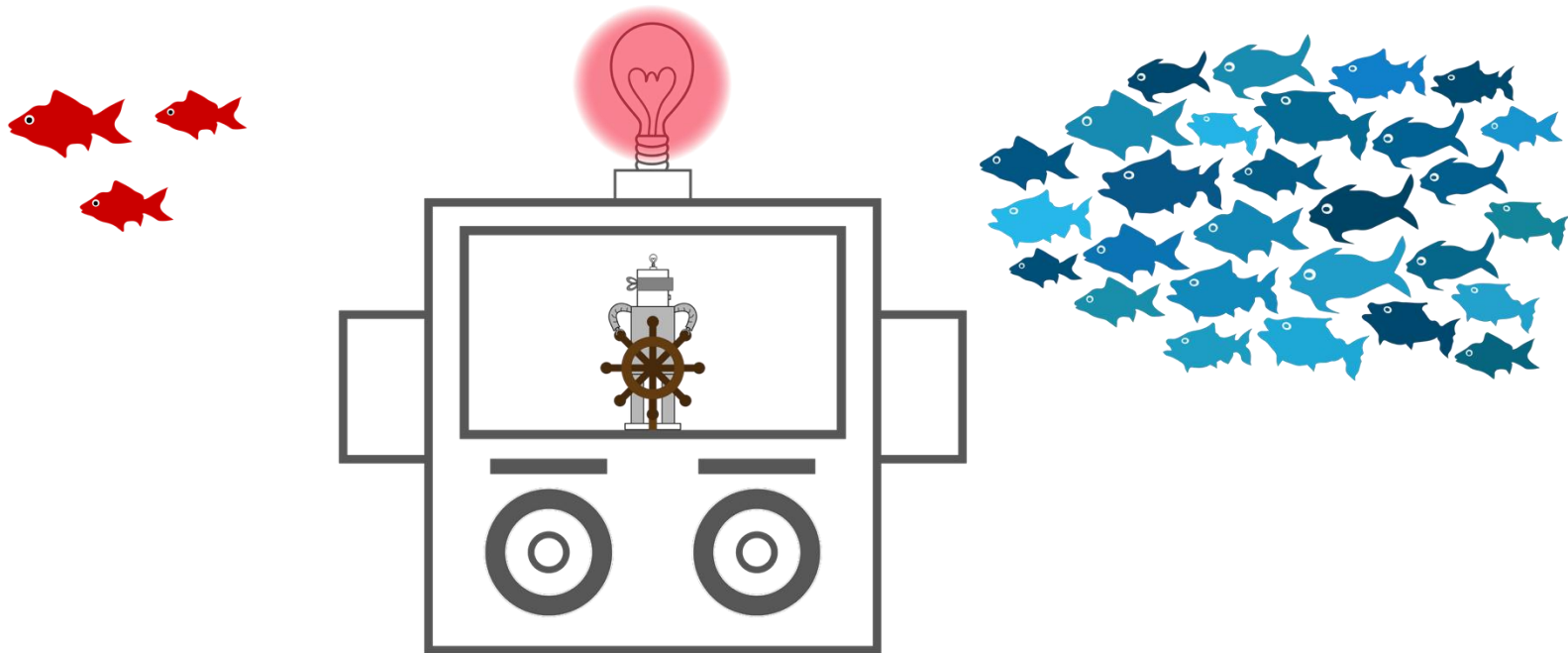
Imbalanced Data: Problem

- Standard classifiers assume **balanced class distributions** or **equal misclassification costs**.



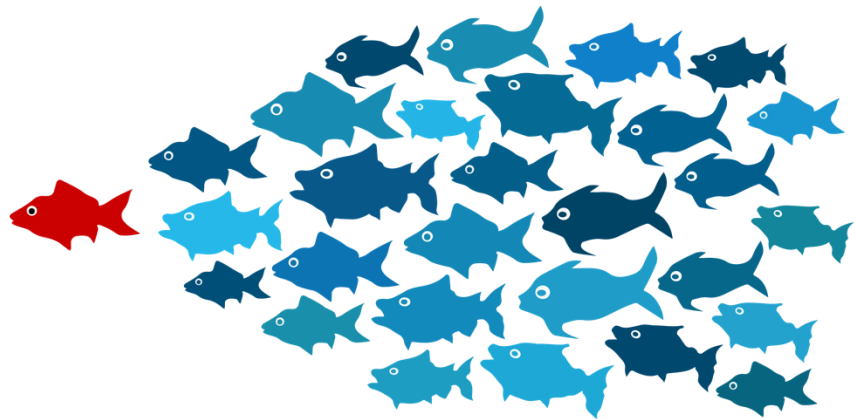
Imbalanced Data: Problem

- Violations of such ideal conditions, hinder the normal learning process of a classifier. Typically, the learning process of most classifiers is **biased towards majority class examples**.



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:**

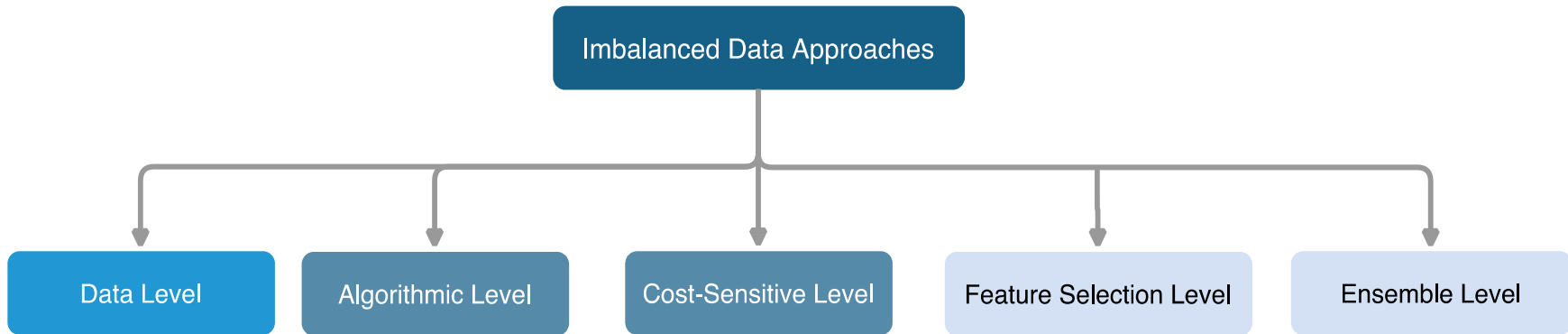
$$C2 = 1 - \frac{1}{IR}, \quad IR = \frac{n_c - 1}{n_c} \sum_{i=1}^{n_c} \frac{n_{c_i}}{n - n_{c_i}}$$

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

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

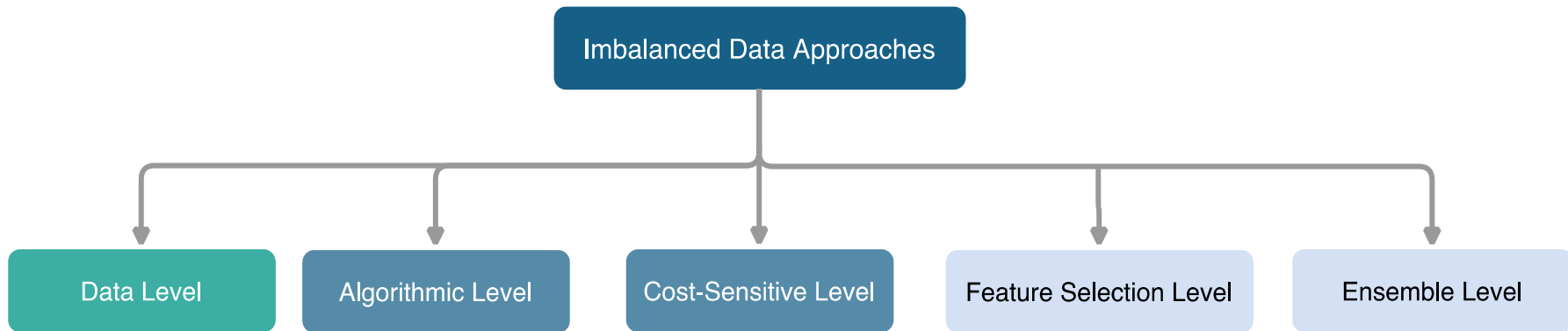
Imbalanced Data: Standard Approaches

- There are several approaches to **handle imbalanced data**:



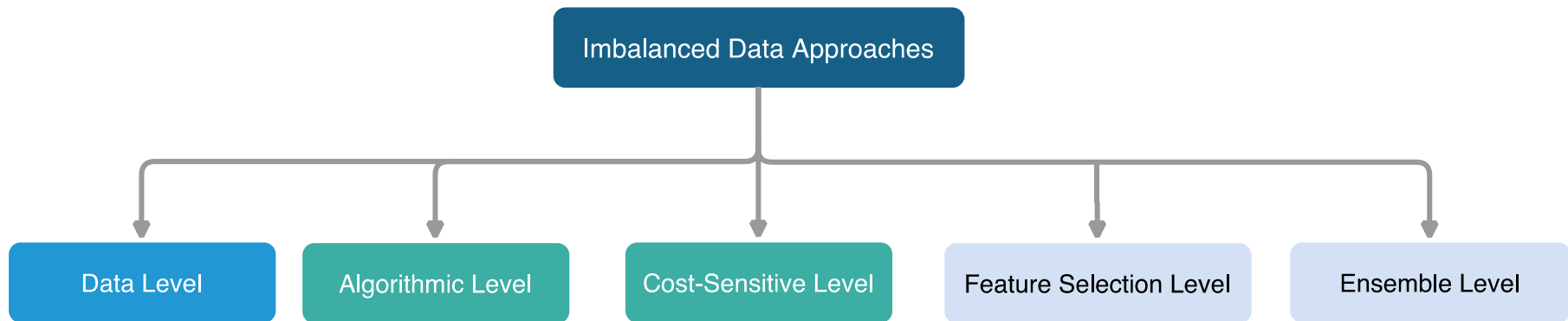
Imbalanced Data: Data Level

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



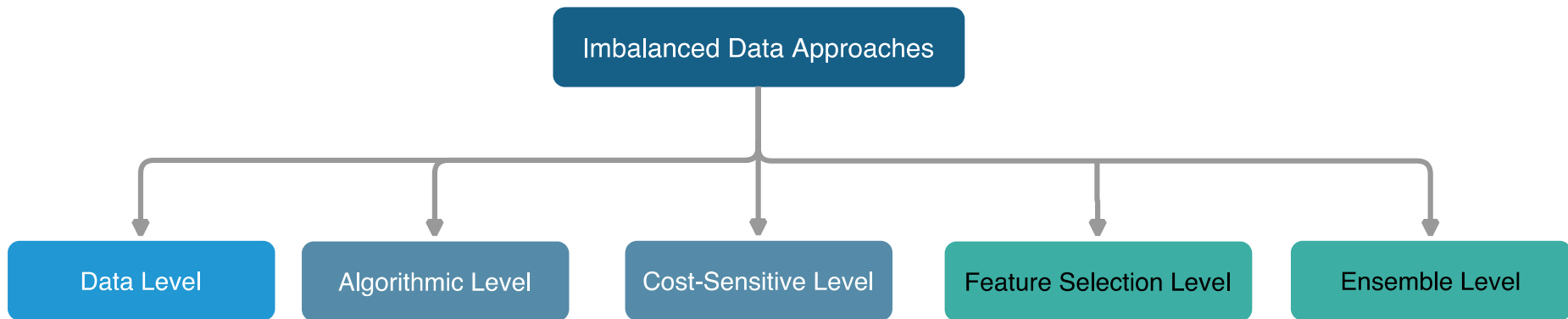
Imbalanced Data: Algorithmic Level and Cost-Sensitive

- **Algorithmic Modification:** Learning methods are adapted to be more attuned to the class imbalance issue (e.g., weighting schemes, one-class classifiers).
- **Cost-Sensitive Classification:** Considers different misclassification costs for different classes.



Imbalanced Data: Feature Selection and Ensembles





- **Feature Selection:** Select an informative subset of features (*how? perhaps using complexity measures?*)
- **Ensembles:** Aggregate the predictions of several classifiers (*how? perhaps studying classifier footprints?*)



Imbalanced Data: Data-Level Approaches

Data Level

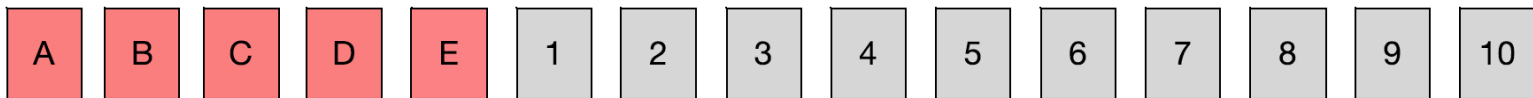
- **Data-Level approaches are perhaps the most commonly used.**

-  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**

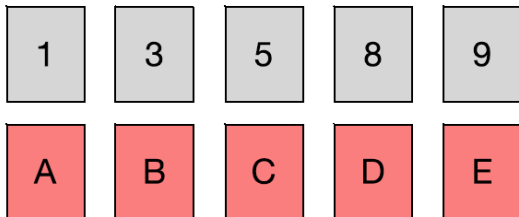
- **There are essentially two main categories:**
 - **Undersampling:** Removing majority examples.
 - **Oversampling:** Adding minority examples.

Imbalanced Data: RUS & ROS

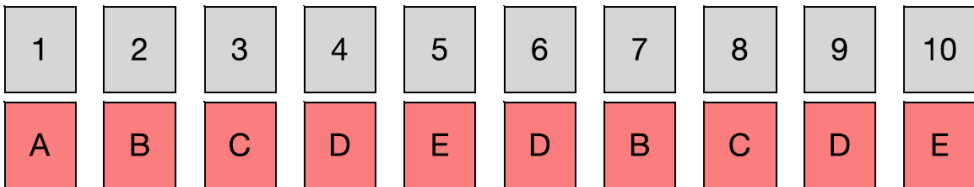
Imbalanced Data



Random Undersampling

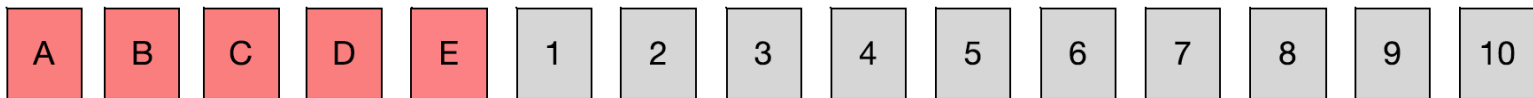


Random Oversampling

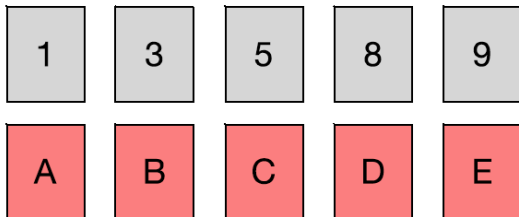


Imbalanced Data: RUS & ROS

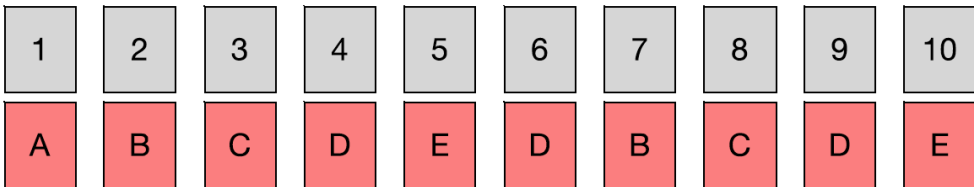
Imbalanced Data



Random Undersampling

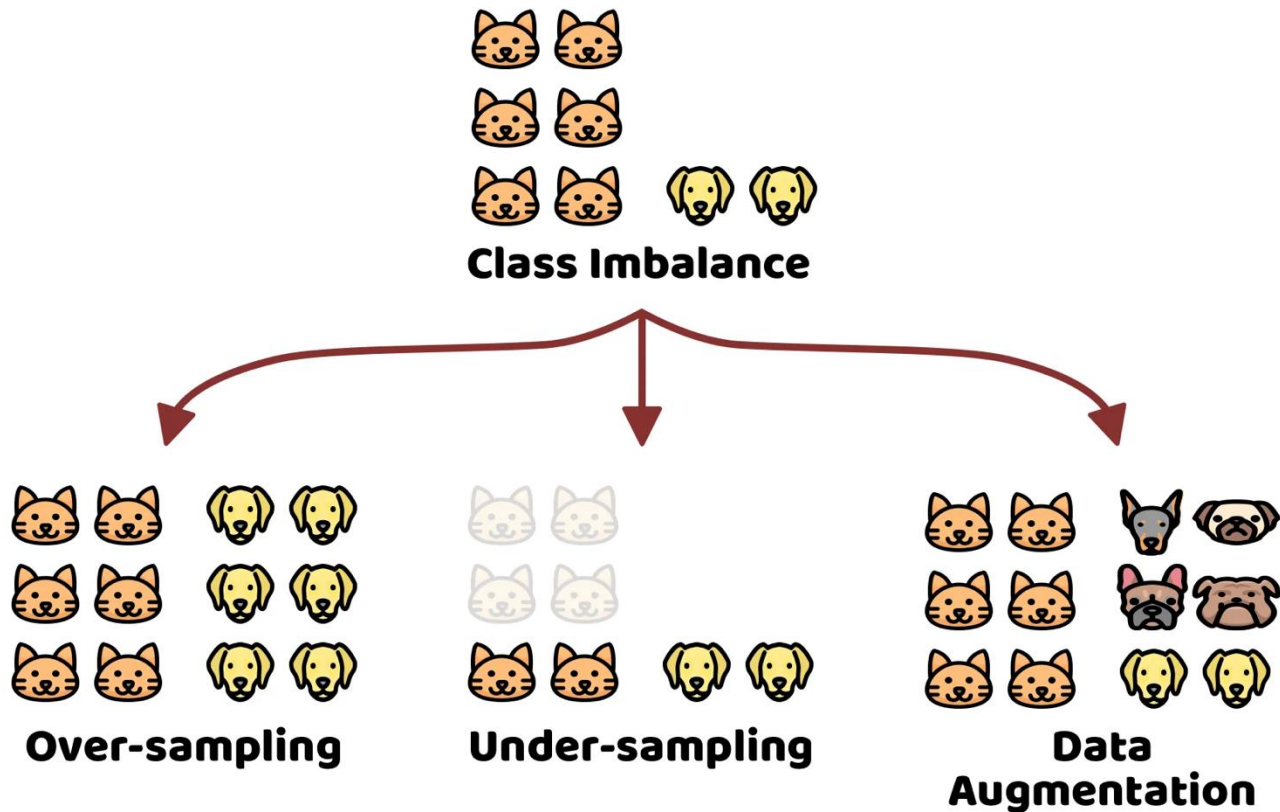


Random Oversampling



Problems?

Imbalanced Data: Synthetic Oversampling



SMOTE: Synthetic Minority Oversampling Technique

Algorithm *SMOTE*(T , N , k)

Input: Number of minority class samples T ; Amount of SMOTE $N\%$; Number of nearest neighbors k

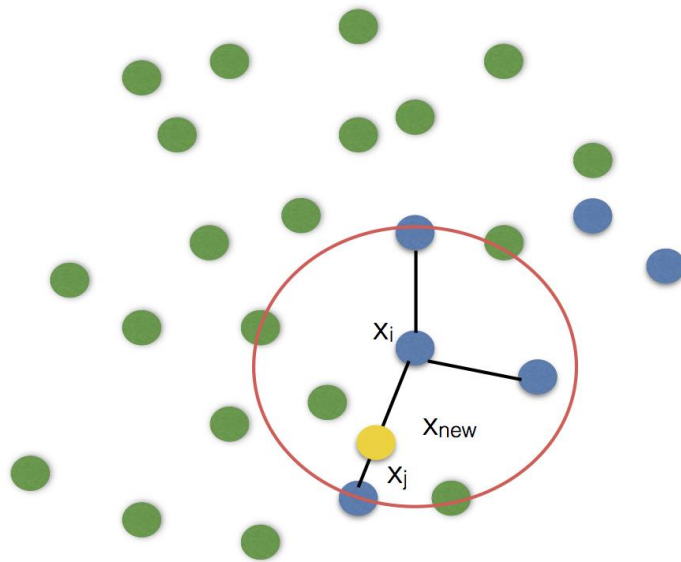
Output: $(N/100) * T$ synthetic minority class samples

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
13. (* 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. *)

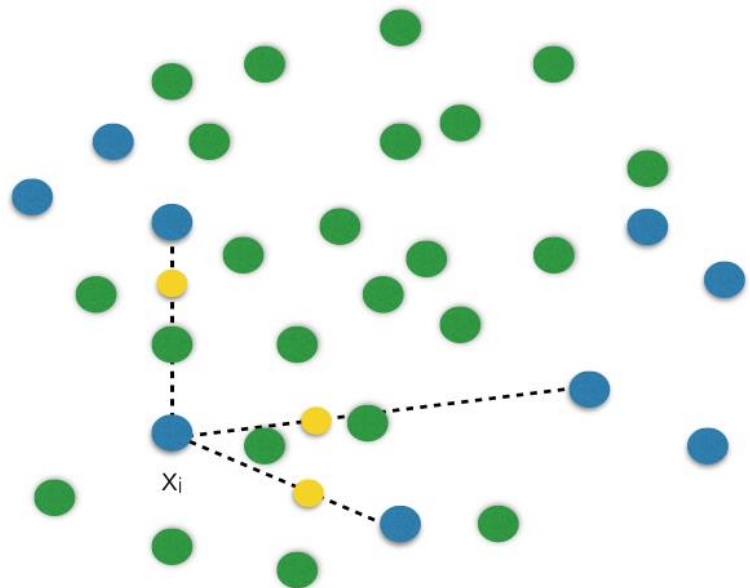
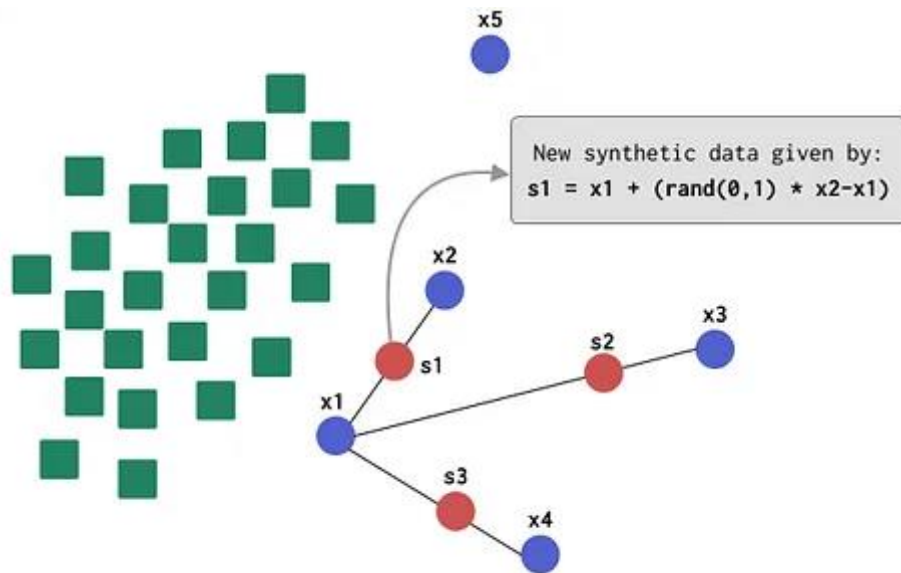
17. while $N \neq 0$
 18. Choose a random number between 1 and k , call it nn . This step chooses one of the k nearest neighbors of i .
 19. for $attr \leftarrow 1$ to $numattrs$
 20. Compute: $diff = Sample[nnarray[nn]][attr] - Sample[i][attr]$
 21. Compute: gap = random number between 0 and 1
 22.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

$k = 3$



$$X_{new} = X_i + (X_j - X_i) \times \delta, \text{ where } \delta \in [0, 1]$$

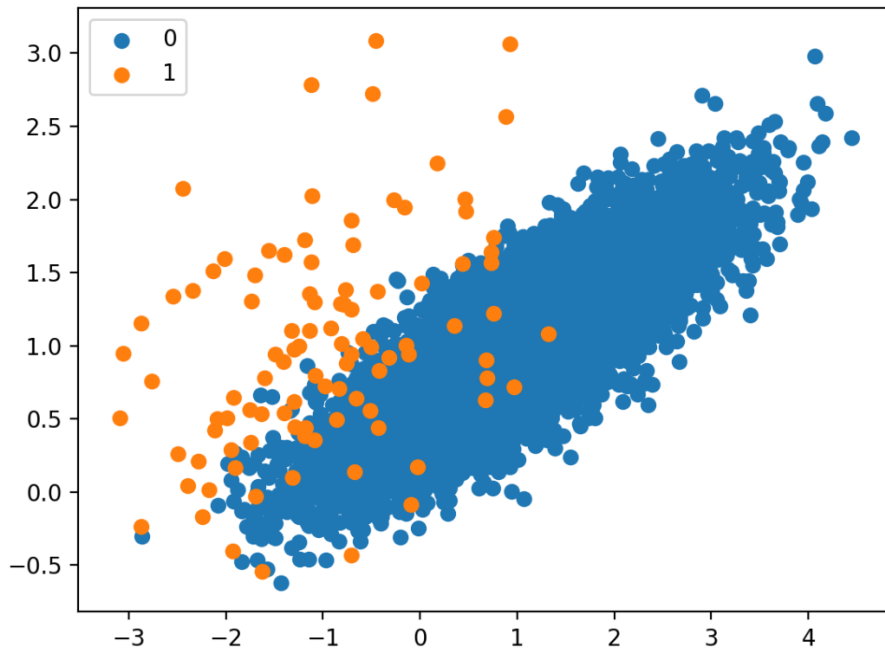
SMOTE: Overgeneralization Problem



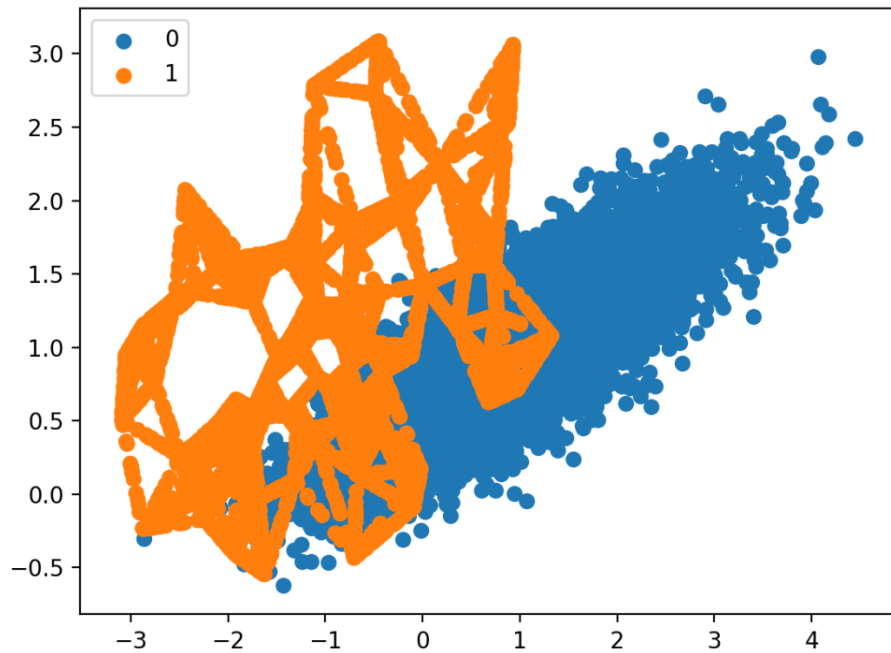
Overgeneralization

SMOTE: Overgeneralization Problem

Imbalanced Data



SMOTE



SMOTE: Other SMOTE variants

SMOTE

ADASYN

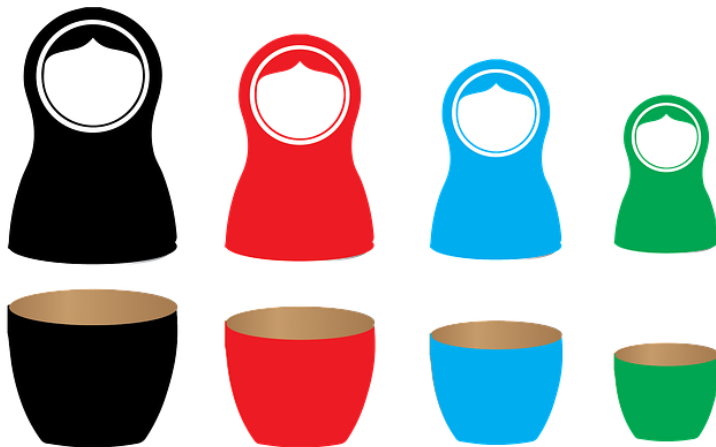
ADOMS

SMOTE-TL

Borderline-SMOTE

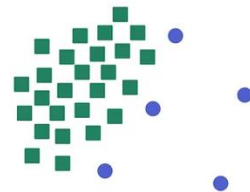
SMOTE-ENN

SL-SMOTE



SMOTE: Borderline-SMOTE

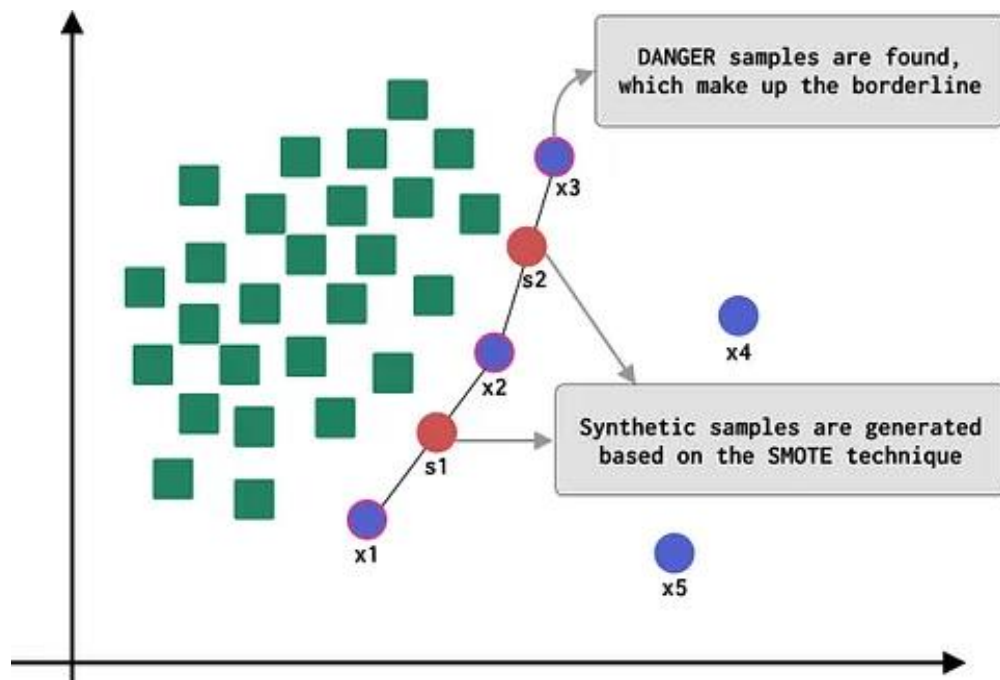
- Borderline-SMOTE considers **only examples on the border** that divides classes. It detects which minority examples are on the border and applies the SMOTE to oversample them.



4) Borderline-SMOTE

Based on the same idea of providing a more clear decision boundary, Han et al. [17] suggested two new variations of SMOTE – Borderline-SMOTE1 and Borderline-SMOTE2 – in which only the minority examples near the borderline are considered for oversampling. Borderline-SMOTE first considers the division of the minority examples into three mutually exclusive sets: noise, safe and danger. This division is made by considering the number of majority examples m' found among each minority example's k nearest neighbors. Thus being, if $m' = k$, all the nearest neighbors of a minority data point p_i are majority examples, and p_i is considered noise; conversely, if $k/2 > m' \geq 0$, p_i is considered safe while if $k > m' \geq k/2$, p_i is surrounded by more majority examples than minority ones (or surrounded by exactly the same number), and therefore is considered danger. The “dan-

ger” data points are considered the minority borderline examples, and only them are oversampled, following a SMOTE-like procedure. For Borderline-SMOTE1 new synthetic examples are created along the line between the danger examples and their minority nearest neighbors; Borderline-SMOTE2 uses the same procedure as Borderline-SMOTE1, but further considers the nearest majority example of each danger data point to produce one more synthetic example: the distance between each danger point and its nearest majority neighbour is multiplied by a *gap* between 0 and 0.5 so that the new point falls closer to the minority class, thus strengthening the minority borderline examples.



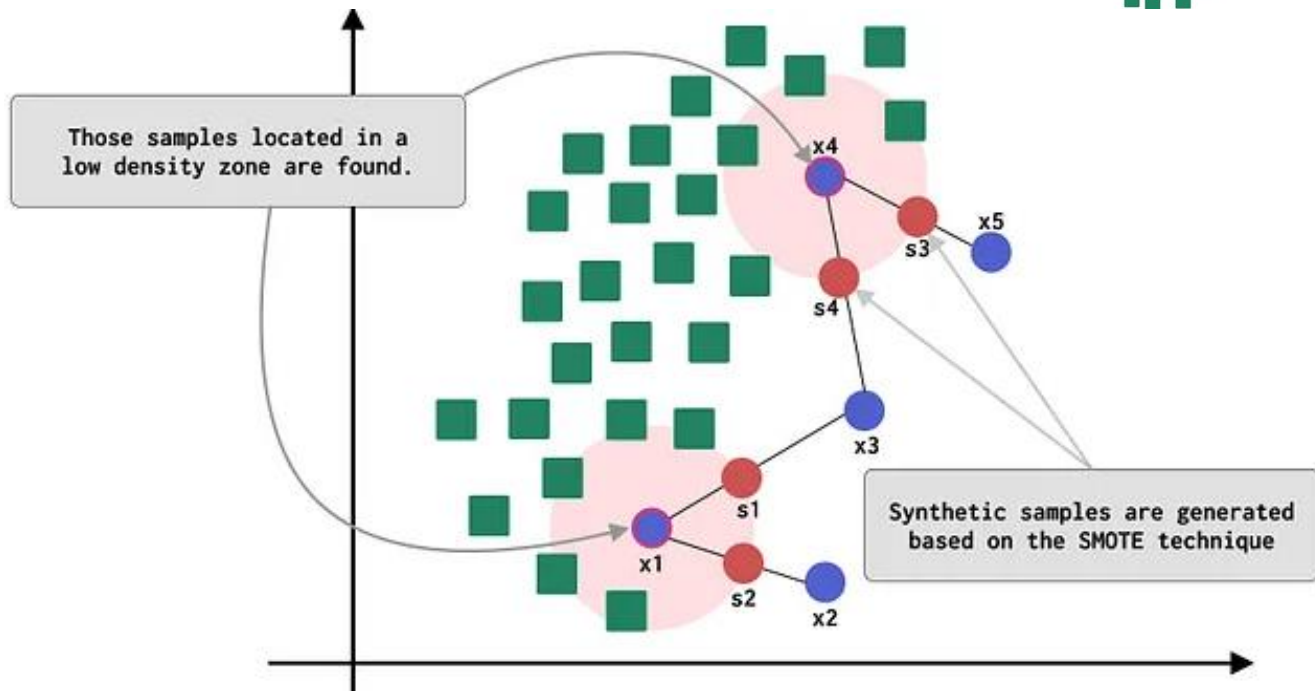
ADASYN: Adaptive Synthetic Sampling Approach

- ADASYN **focuses on “harder to learn”** minority examples, i.e., those surrounded by majority class examples.



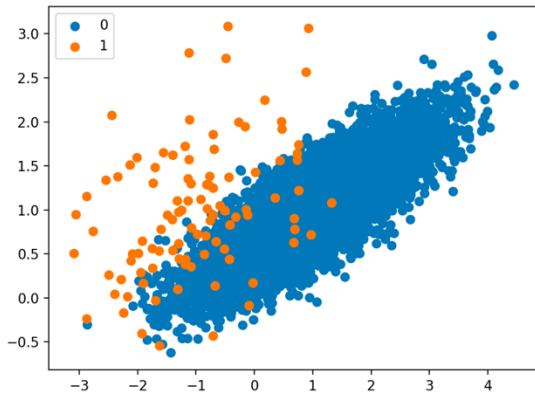
3) ADASYN

Instead of producing an equal number of synthetic minority instances for each minority example, the Adaptive Synthetic Sampling Approach (ADASYN) algorithm, proposed by He et al. [16], specifies that minority examples harder to learn are given a greater importance, being oversampled more often. ADASYN determines a weight (w_i) for each minority example, defined as the normalized ratio of majority examples N_i among its k nearest neighbors: $w_i = N_i / k \times z$ where z is a normalization constant. Then, the number of synthetic data points to generate for each minority example is specified as $g_i = w_i \times G$, being G the total necessary number of synthetic minority samples to produce according to the required amount of oversampling. The oversampling procedure is the same as SMOTE; the only difference is that harder minority examples are replicated more often.

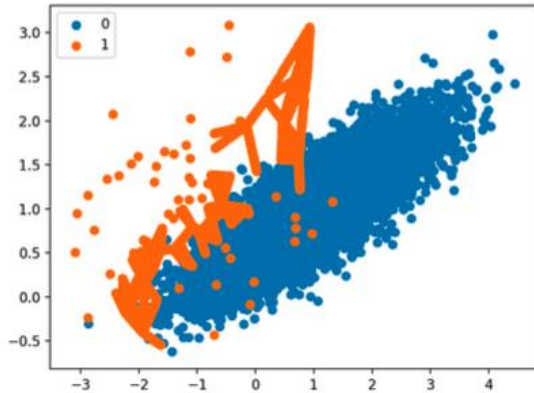


SMOTE, Borderline-SMOTE, and ADASYN

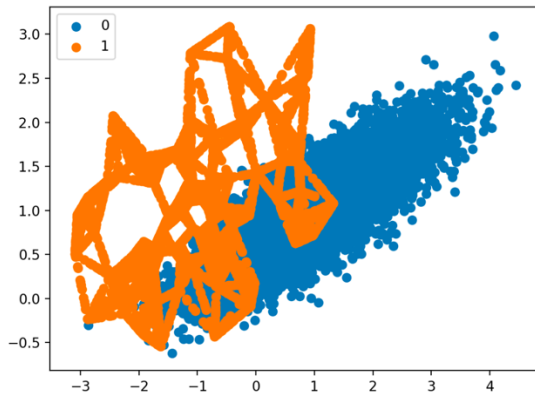
Imbalanced Data



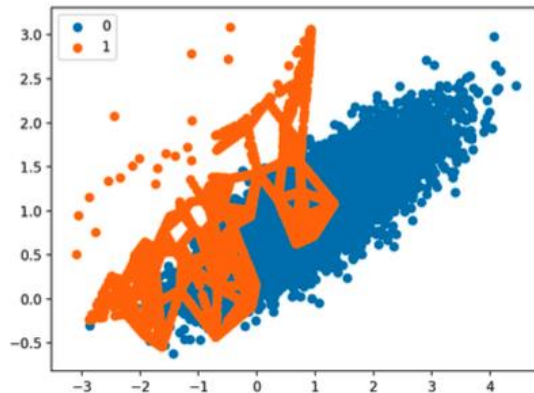
Borderline SMOTE



SMOTE

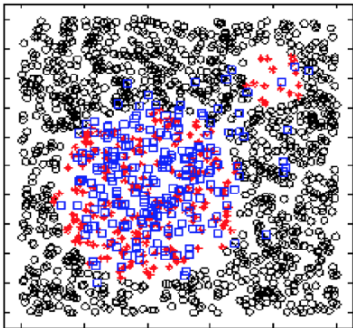


ADASYN

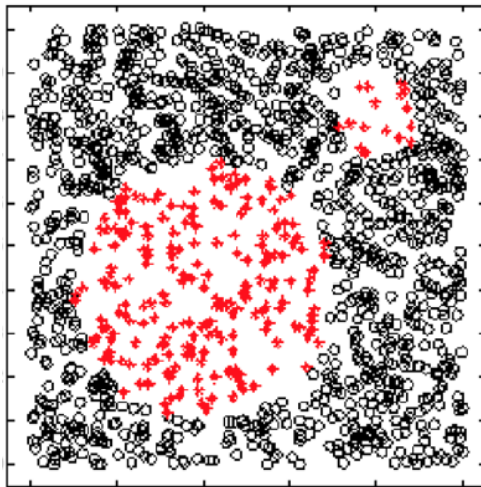
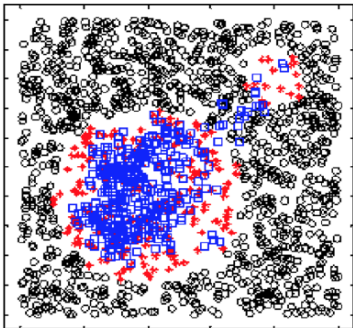


SMOTE: SMOTE variants

1

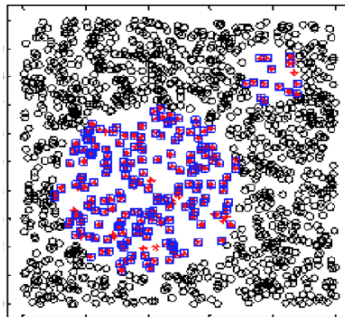


2

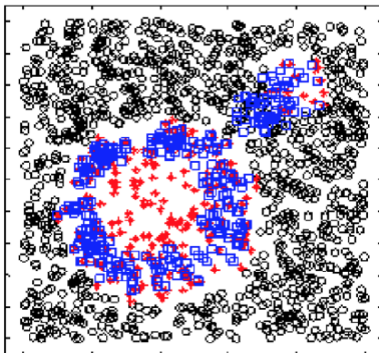


ROS
SMOTE
Safe-Level-SMOTE
Borderline-SMOTE

3



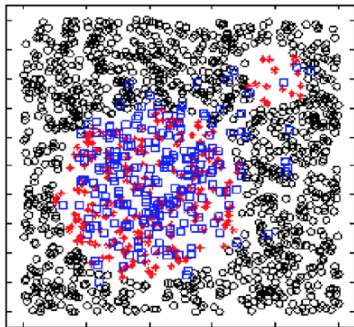
4



SMOTE: SMOTE variants

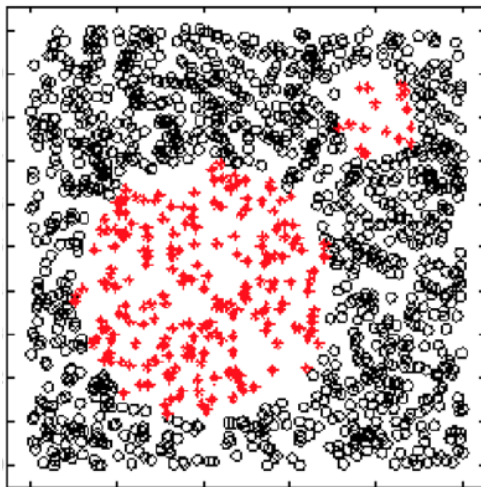
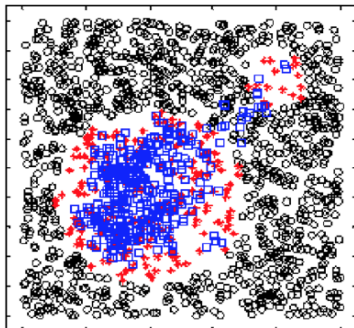
1

SL-SMOTE



2

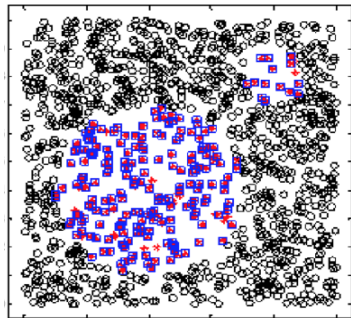
SMOTE



ROS
SMOTE
Safe-Level-SMOTE
Borderline-SMOTE

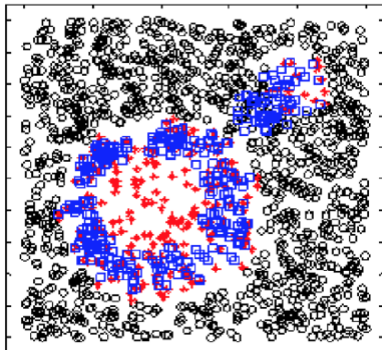
3

ROS



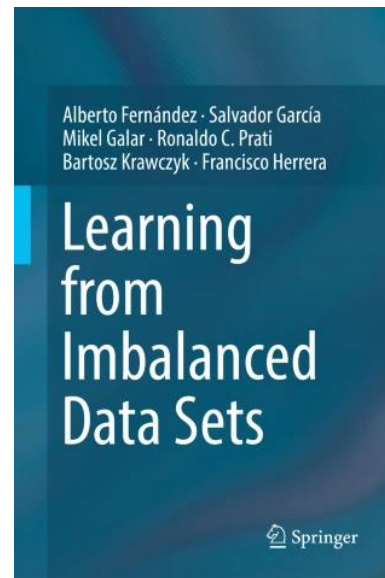
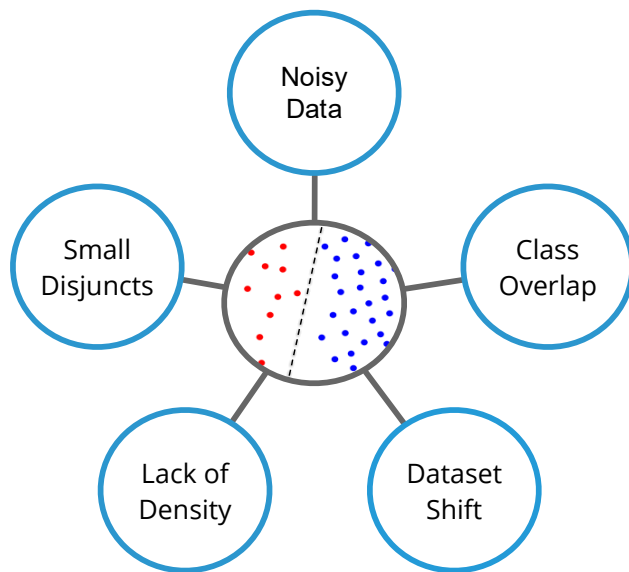
4

Borderline-SMOTE



Imbalanced Data: interplay with other factors

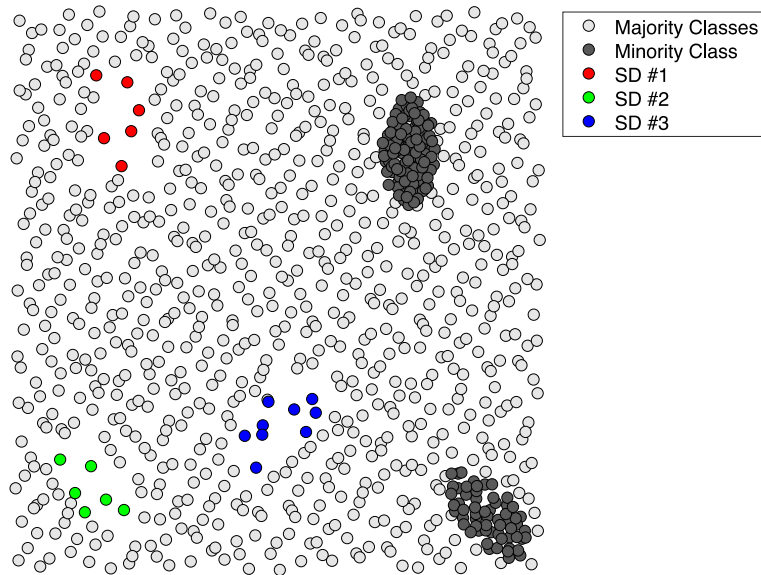
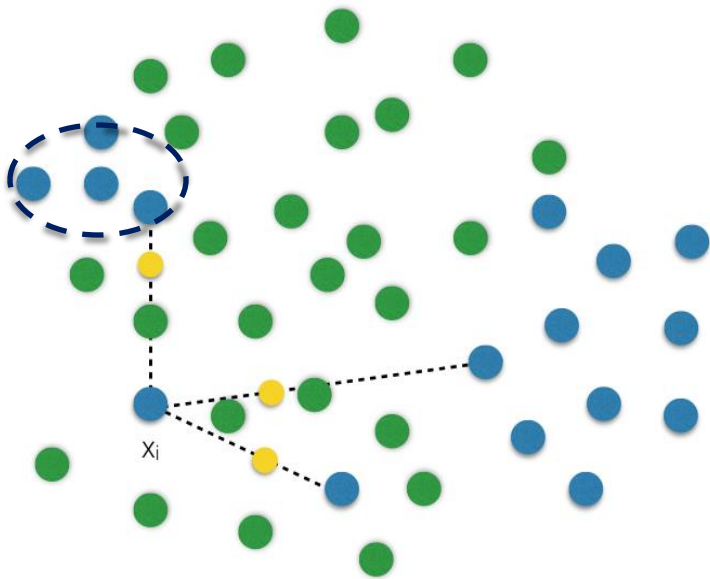
- Although class imbalance is an **important problem in isolation**, its **combination with other factors** creates a much more difficult setting for classifiers.



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

Imbalanced Data: the problem of Small Disjuncts

- **Between and within class imbalance**
 - Classifiers are typically biased towards classifying **larger disjuncts** accurately



Small Disjuncts: CBO – Cluster-Based Oversampling

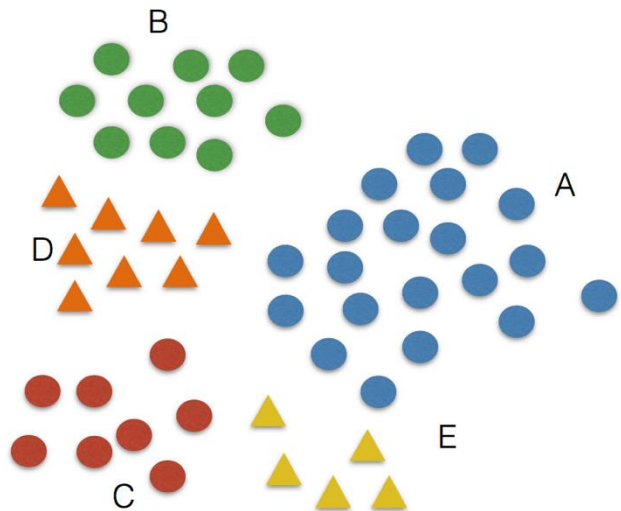
Number of examples in each cluster:

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:

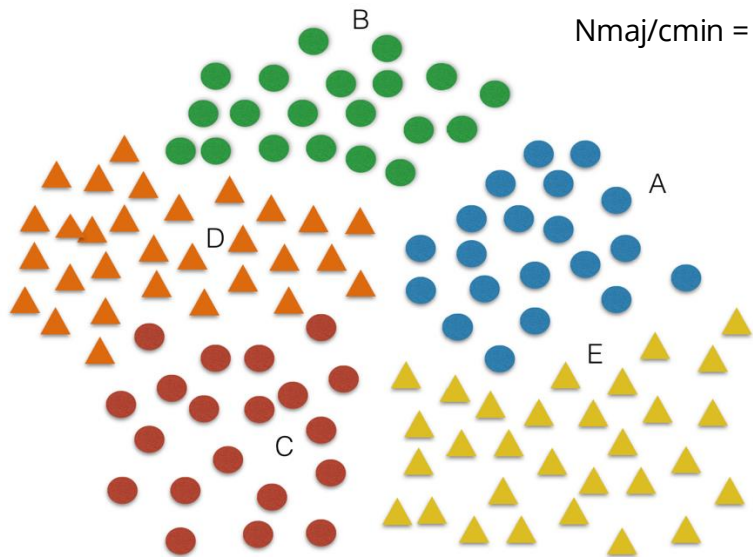
Majority class: A: 20; B: 20; C: 20

$$C_{maj} = 3$$

Minority class: D: 30; E: 30

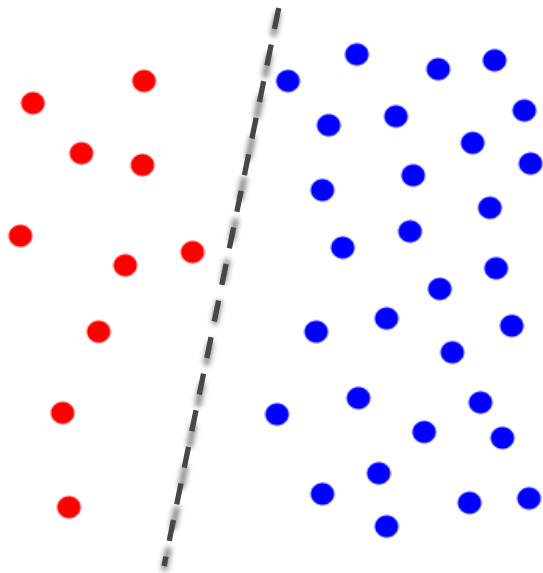
$$C_{min} = 2$$

$$N_{maj}/c_{min} = 60/2 = 30$$

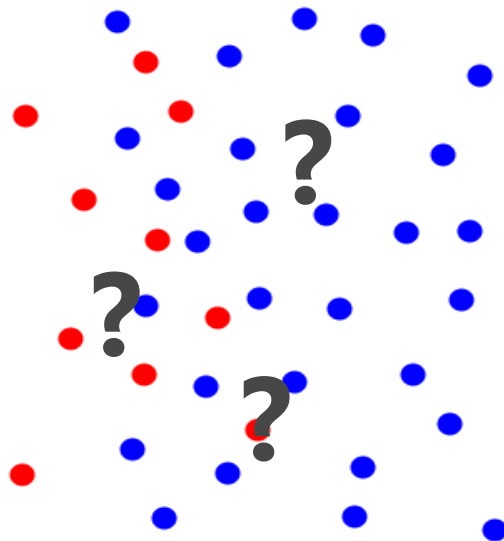


Imbalanced Data: the problem of Class Overlap

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

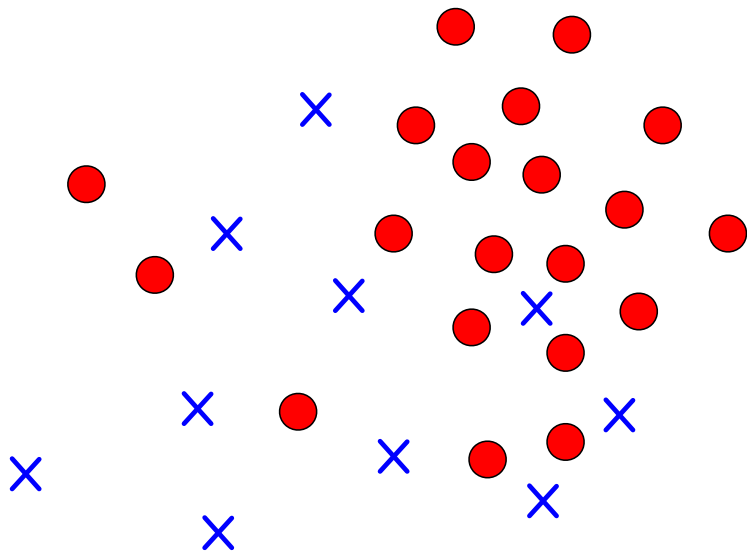


IR = 3:1

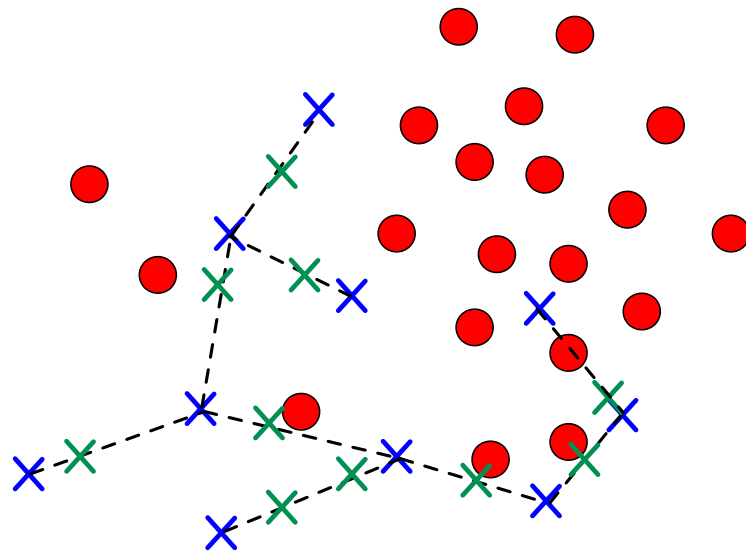


IR = 3:1

Class Overlap: SMOTE-TL and SMOTE-ENN

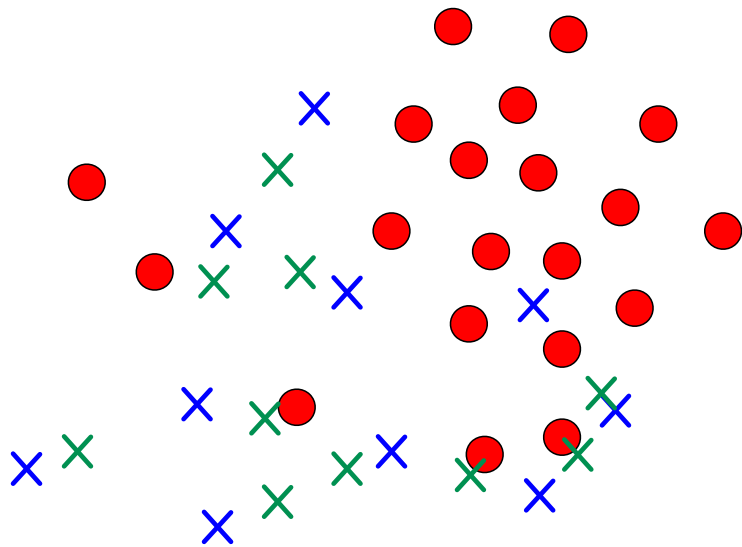


Imbalanced Data

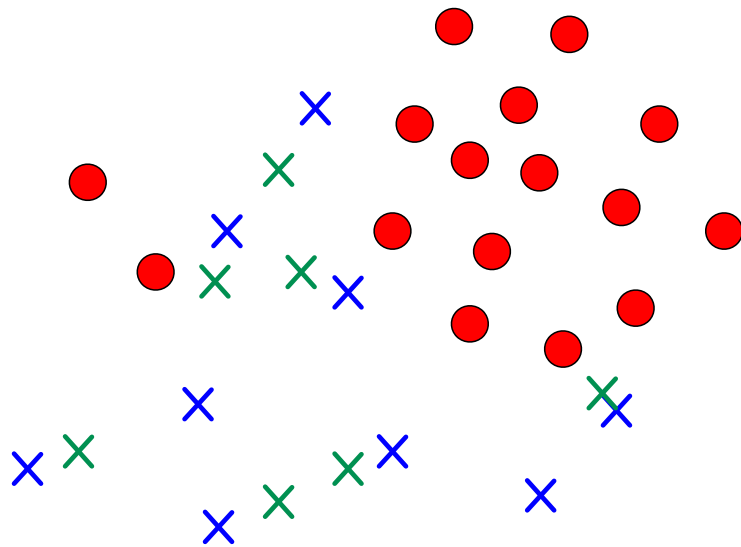


SMOTE

SMOTE-TL: SMOTE + Tomek Links

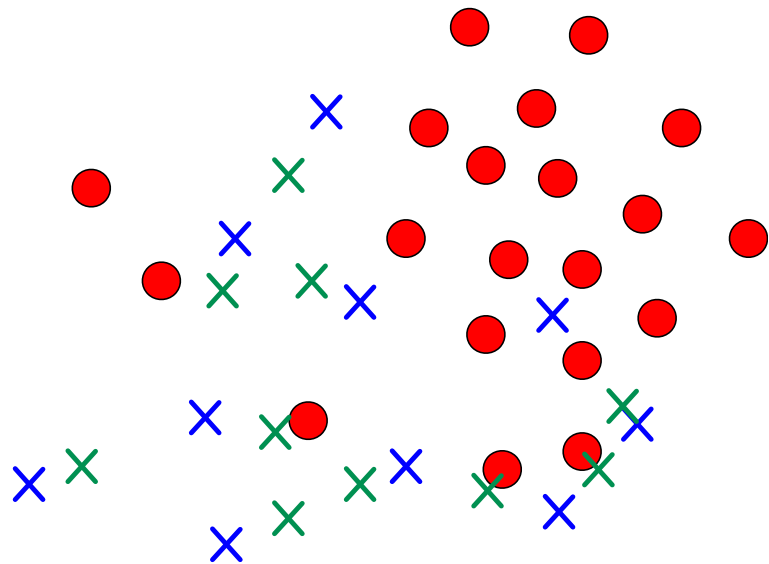


Tomek Links

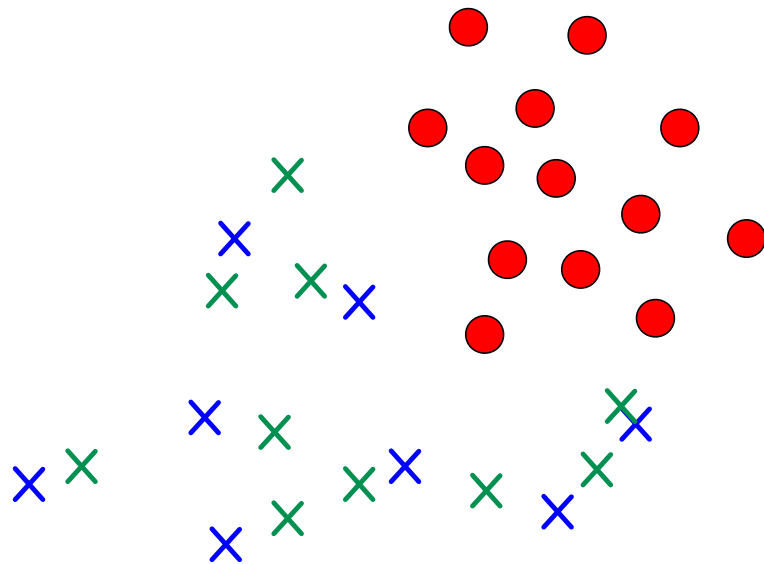


SMOTE-TL

SMOTE-ENN: SMOTE + Edited Nearest Neighbor

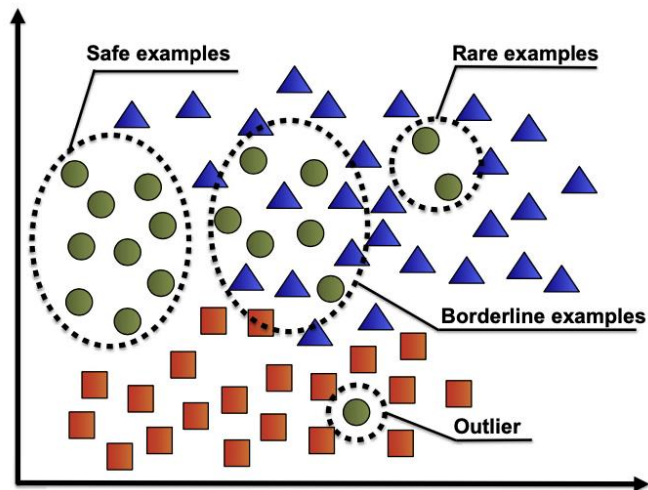
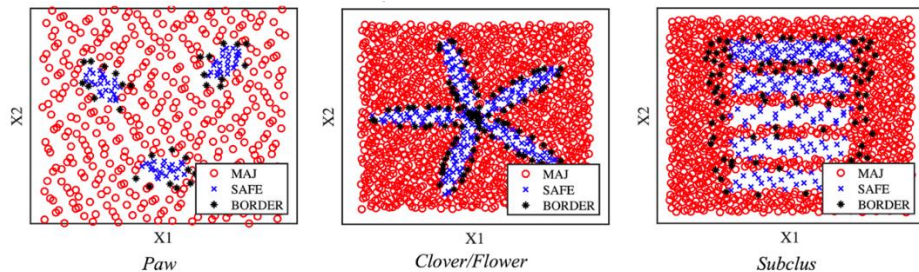


ENN



SMOTE-ENN

Imbalanced Data: Data Difficulty Factors


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

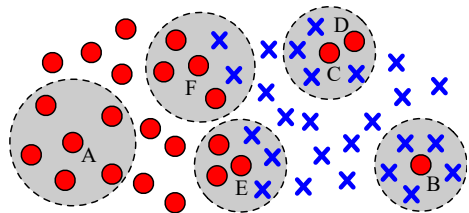
Dataset	S [%]	B [%]	R [%]	O [%]
breast-w	91.29	7.88	0.00	0.83
abdominal-pain	59.90	22.28	8.90	7.92
acl	67.50	30.00	0.00	2.50
new-thyroid	68.57	31.43	0.00	0.00
vehicle	74.37	24.62	0.00	1.01
nursery	82.00	17.00	1.00	0.00
satimage	47.47	39.76	4.58	8.19
car	47.83	39.13	8.70	4.35
scrotal-pain	38.98	45.76	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

Imbalanced Data: Data Difficulty Factors

Table 4

Performance results of C4.5 comparing not preprocessing (*None*), preprocessing all the classes (*All*) and the best configuration found to oversample (*Best*). Best results for each dataset are remarked in bold. For each dataset, the best valid configuration is given, indicating the class preprocessed (column *class*) and whether the different types of examples are preprocessed (columns *safe*, *borderline*, *rare* and *outlier*).

Dataset	None	All	Best	Safe	Borderline	Rare	Outlier	Class
Automobile	72.50	80.08	83.87	True	True	False	True	4
Balance	55.92	55.03	56.57	False	False	True	False	1
Car	80.21	92.29	81.57	True	False	False	False	2
Cleveland	25.78	28.43	34.30	False	True	False	True	1
Contraceptive	50.42	50.25	52.73	False	True	True	True	1
Dermatology	95.98	95.79	97.11	True	False	False	False	1
Ecoli	63.94	64.68	72.21	False	True	False	True	5
Flare	60.51	63.80	65.54	False	True	True	True	1
Glass	66.85	71.16	71.14	False	True	False	True	4
Hayes-roth	85.04	86.11	85.56	True	True	False	True	2
Led7digit	71.63	72.38	72.81	False	True	True	False	1
Lymphography	74.22	71.41	80.55	False	False	False	True	3
Newthyroid	91.24	94.54	93.65	False	True	False	False	2
Pageblocks	80.24	75.86	86.30	True	True	False	False	4
Post-operative	45.23	33.07	45.61	False	True	True	True	2
Thyroid	97.03	80.50	93.63	False	True	False	False	1
Vehicle	72.23	73.30	73.48	True	True	True	True	2
Wine	94.85	92.20	95.11	False	True	False	False	1
Winequality-red	34.57	37.25	37.43	False	True	False	True	2
Yeast	52.19	50.26	54.84	False	True	True	True	9
Zoo	85.48	82.26	89.76	True	True	False	False	4
Average	69.34	69.08	72.56	False (14/21)	True (17/21)	False (14/21)	True (12/21)	1 (8/21)
Best (out of 21)	1	4	16	–	–	–	–	–



Imbalanced Data: Experimental Design Pitfalls

Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches

Research
Frontier

Abstract

Although cross-validation is a standard procedure for performance evaluation, its joint application with oversampling remains an open question for researchers farther from the imbalanced data topic. A frequent experimental flaw is the application of oversampling algorithms to the entire dataset, resulting in biased models and overly-optimistic estimates. We emphasize and distinguish overoptimism from overfitting, showing that the former is associated with the cross-validation procedure, while the latter is influenced by the chosen oversampling algorithm. Furthermore, we perform a thorough empirical comparison of well-established oversampling algorithms, supported by a data complexity analysis. The best oversampling techniques seem to possess three key characteristics: use of cleaning procedures, cluster-based example synthetization and adaptive weighting of minority examples, where Synthetic Minority Oversam-

pling Technique coupled with Tomek Links and Majority Weighted Minority Oversampling Technique stand out, being capable of increasing the discriminative power of data.

1. Introduction

Imbalanced Data (ID) occurs when there is a considerable difference between the

an under-represented concept (a minority class) when compared to the other (a majority class) [1]. Prediction models built from imbalanced datasets are most often biased towards the majority concept, which is especially critical when there is a higher cost of misclassifying the minority examples, such as diagnosing rare diseases [2].

Approaches to handle imbalanced scenarios can be mainly divided into data-level approaches, where the data is preprocessed in order to achieve a balanced dataset for classification, and algorithmic-level approaches, where the classifiers are adapted to deal with the characteristic issues of imbalanced data [3–6]. By far, data-level approaches are the most commonly used, as they have proven to be efficient, are simple to implement and completely classifier-independent [2], [7]. Data-level strategies fall into two main categories, undersampling and oversampling: the former consists in removing majority examples while the latter replicates the minority examples. Researchers often invest in oversampling procedures since they are capable



IMAGE LICENSED BY PICTUREM PUBLISHING

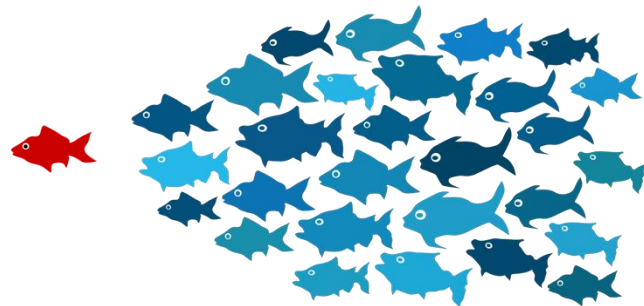
class priors of a given problem. Considering a binary classification problem, a dataset is said to be imbalanced if there exists

- **Poor performance evaluation** in imbalanced domains
- **Faulty application of resampling** approaches
- Distinguishing between **overoptimistic** and **overfitting** approaches

Imbalanced Data: Performance Evaluation

- **100 blue** fish vs.
10 red fish

		Actual Class	
		Blue fish	Red fish
Predicted Class	Blue fish	100	8
	Red fish	0	2

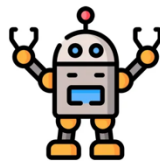


A	B	C	Class
			0
			0
			0
			...
			1

100 samples
of class 0

10 samples
of class 1

Say we train a ML model
without fixing the class
imbalance problem.



ML model

The results in terms
of a confusion matrix
would probably be:

	Real class 1	Real class 0
Predicted class 1	2	0
Predicted class 0	8	100

Accuracy = 92%

The given dataset presents
the class imbalance problem
with a ratio 1:10

- **Accuracy** is 92%! *Good result?*

Imbalanced Data: Performance Evaluation

	Real class 1	Real class 0
Predicted class 1	2	0
Predicted class 0	8	100

True Positives = 2

False Positives = 0

True Negatives = 100

False Negatives = 8

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total \# samples}}$$

$$\text{Accuracy} = \frac{2 + 100}{110}$$

$$\text{Accuracy} = 0.92$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Precision} = \frac{2}{2 + 0}$$

$$\text{Precision} = 1$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Recall} = \frac{2}{2 + 8}$$

$$\text{Recall} = 0.2$$

$$\text{F1-Score} = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}}$$

$$\text{F1-Score} = \frac{2 \times (0.2 \times 1)}{0.2 + 1} = \frac{0.4}{1.2}$$

$$\text{F1-Score} = 0.33$$

- **Blue fish recognition: 100%** (*specificity*)
- **Red fish recognition: 20%** (*sensitivity*)
- **Common performance measures** in Imbalanced Learning literature

- Sensitivity
- F-measure
- G-mean
- AUC

$$\text{Sensitivity} = \frac{TP}{\text{Total Positive}} = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$G_{\text{mean}} = \sqrt{\text{Sensitivity} \times \text{Specificity}}$$

Imbalanced Data: Cross Validation

SMOTE oversampling and cross-validation

I am working on a binary classification problem in Weka with a highly imbalanced data set (90% in one category and 10% in the other). I first applied SMOTE (<http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/node6.html>) to the entire data set to even out the categories and then performed 10-fold cross-validation over the newly obtained data. I found (overly?) optimistic results with F1 around 90%.

Is this due to oversampling? Is it bad practice to perform cross-validation on data on which SMOTE is applied? Are there any ways to solve this problem?

[machine-learning](#)[weka](#)[text-classification](#)

Imbalanced Data: Cross Validation

The logo consists of a teal square containing the text 'R' in white, with a superscript 'G' to its upper right.

What is a possible solution for cross validation of an imbalanced data set problem?

What is a possible solution for cross validation of an imbalanced data set problem? The question is in three sections. 1. 1- Oversample the minority class examples using (SMOTE, ADASYN etc), then split it into 10 folds, train the classifier on first nine folds and test on 10th fold and repeat this process 10 times and take the average of metric measure then what about overfitting problem? 2. what about if we divide the data set into 10 folds, oversample the minority class examples in first ninth folds and train the classifier and test the trained classifier on the original (Not oversampled) 10th fold repeat this process 10 times and take the average .. question is what about distribution because basic assumption is training and test set follow the same distribution. 3. If we oversample the minority class examples same as number of majority class examples, then it is necessary to measure F-Measure, G-mean and AUC or accuracy measure is sufficient.

[Data Analysis](#)[Data Mining and Knowledge Discovery](#)[Cross-Validation](#)

Imbalanced Data: Cross Validation



Cross Validated

I am working on severely imbalanced data. In literature, several methods are used to re-balance the data using re-sampling (over- or under-sampling). Two good approaches are:

- SMOTE: Synthetic Minority Over-sampling TEchnique (**SMOTE**)
- ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning (**ADASYN**)

I have implemented ADASYN because its adaptive nature and ease to extension to multi-class problems.

My question is how to test the oversampled data produced by ADASYN (or any other oversampling methods). It is not clear in the mentioned two paper how they performed their experiments. There are two scenarios:

1- Oversample the whole dataset, then split it to training and testing sets (or cross validation).

2- After splitting the original dataset, perform oversampling on the training set only and test on the original data test set (could be performed with cross validation).

In the first case the results are much better than without oversampling, but I am concerned if there is overfitting. While in the second case the results are slightly better than without oversampling and much worse than the first case. But the concern with the second case is if all minority class samples goes to the testing set, then no benefit will be achieved with oversampling.

I am not sure if there are any other settings to test such data. Waiting for your inputs.

[classification](#)[dataset](#)[resampling](#)[unbalanced-classes](#)[oversampling](#)

Imbalanced Data: Cross Validation

Will SMOTE make you more prone to overfit? (self.datascience)

submitted 1 year ago by Cjh411

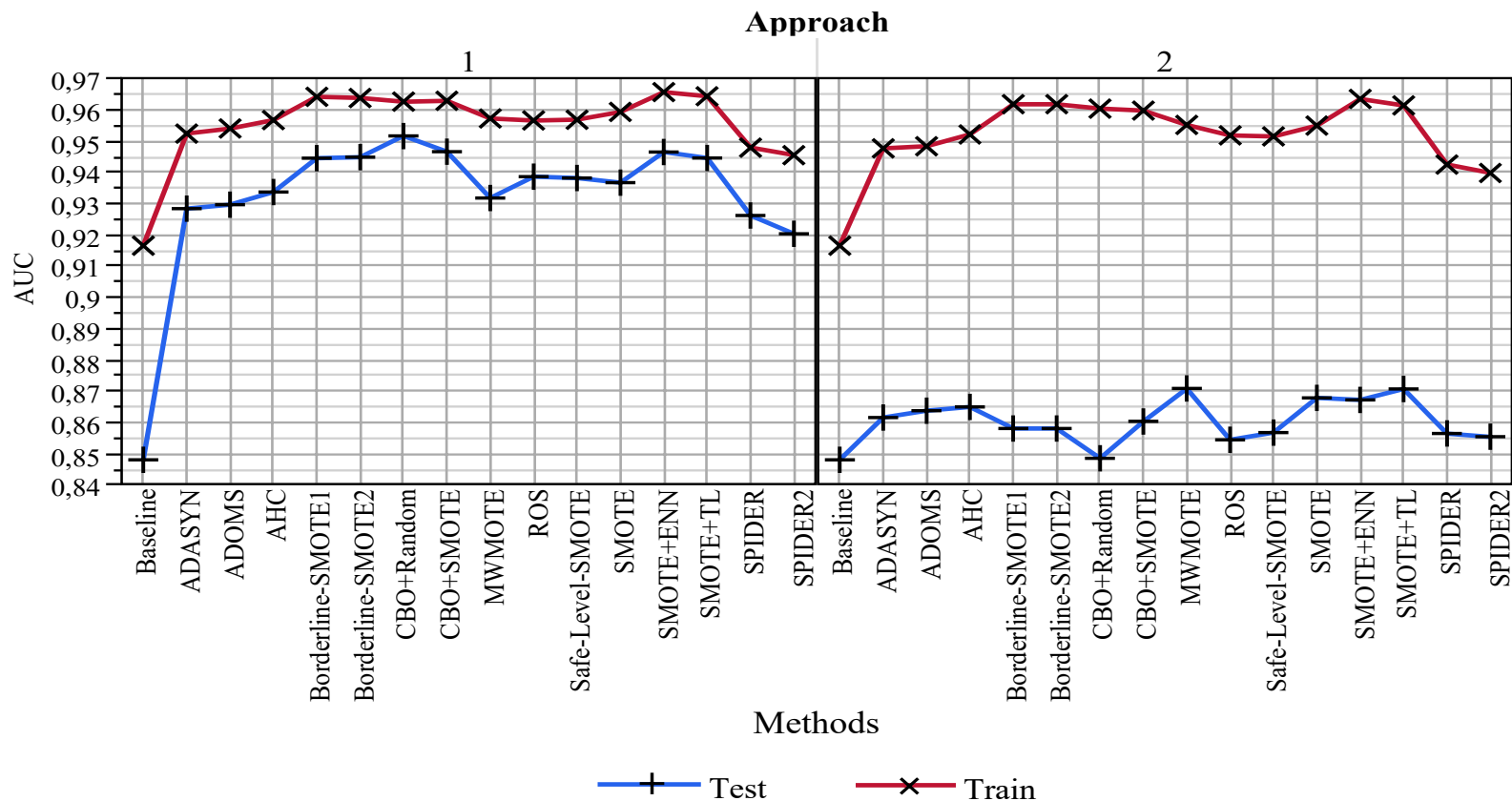
Someone brought up using SMOTE on a project where there is not only a large data imbalance but also very few minority records in absolute value (50 out of 1000). The data is very noisy and there is a lot of overlap between the minority and majority cases.

I understand the benefit of SMOTE in correcting for bias in your model. But In this situation do you run the risk of overfitting your model and losing generalizability on future data? I've tried searching the web and can't find much on the relationship between SMOTE and overfitting on small samples. Intuitively it feels dangerous to strengthen a signal in your data that you're skeptical of to begin with. We're using 10 fold cv for evaluation.

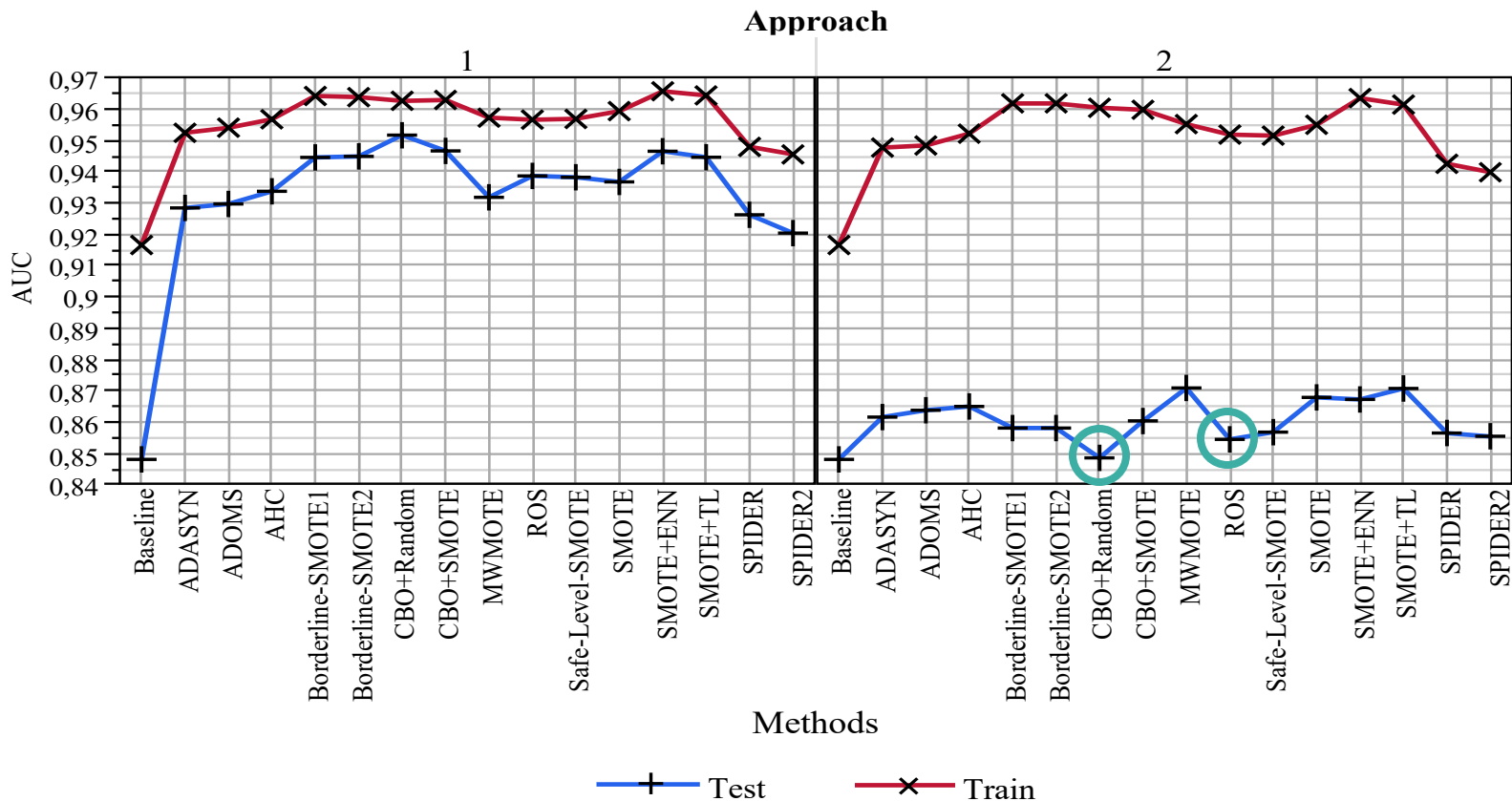
Imbalanced Data: Cross Validation



Imbalanced Data: Cross Validation



Imbalanced Data: Cross Validation



Imbalanced Data: Experimental Design Pitfalls

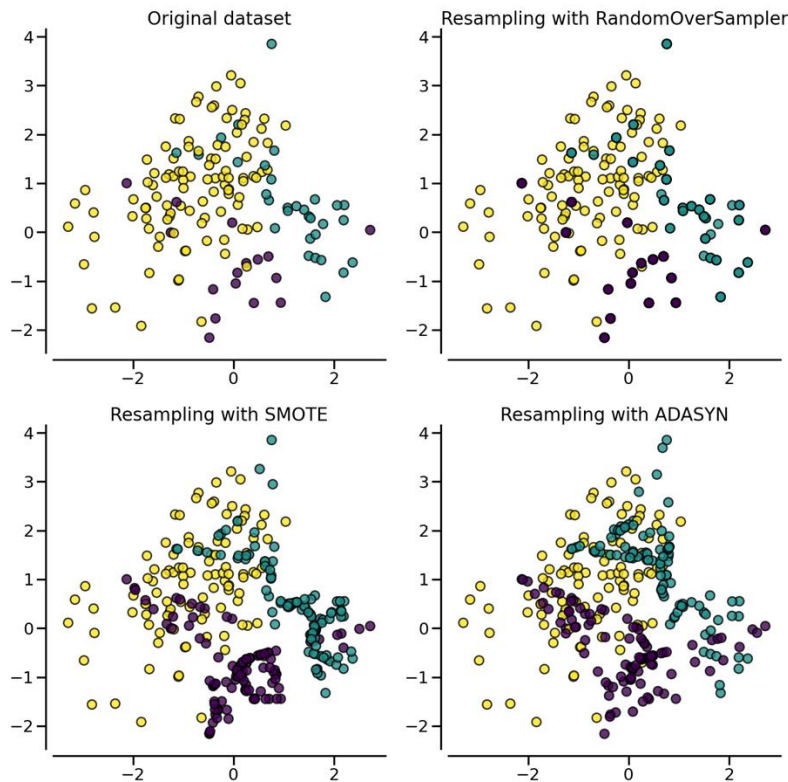
- **Imbalanced Data requires more informative measures:**
 - Accuracy / Error Rate are not appropriate
- **Overoptimism is associated with inappropriate validation setups:**
 - Oversampling should be applied after crossvalidation (only on the training set)
- **Overfitting is mostly related to the random oversampling algorithm:**
 - Creating exact replicas of existing patterns is the most prejudicial technique

The best oversampling techniques possess three key characteristics: use of cleaning procedures, cluster-based synthetization of examples and adaptive weighting of minority examples.

Imbalanced Data

Practice with Python

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)
```

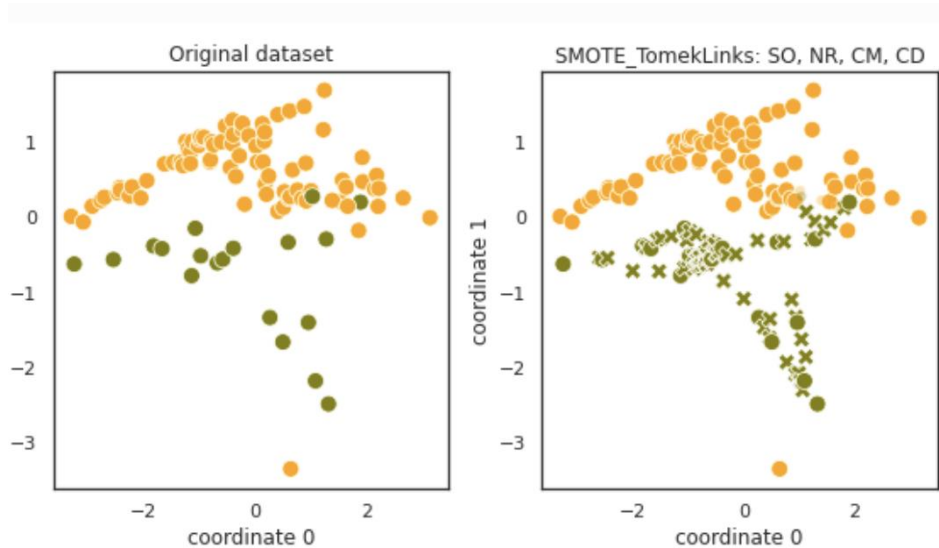


<https://imbalanced-learn.org/stable/>



`pip install imbalanced-learn`

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.

```
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.


KEEL Platform



- Description** 
- Download KEEL Software Tool** 
- Reference Manual** 
- KEEL-dataset Data set repository** 
- KEEL Included Algorithms** 
- Links** 
- Identification Access** 

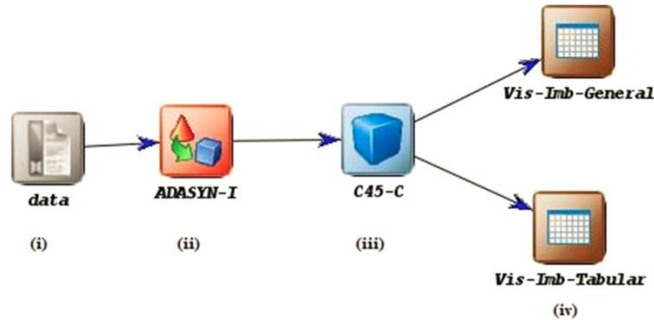


KNOWLEDGE EXTRACTION *based on* EVOLUTIONARY LEARNING



KEEL (Knowledge Extraction based on Evolutionary Learning) is an open source (GPLv3) Java software tool that can be used for a large number of different knowledge data discovery tasks. KEEL provides a simple GUI based on data flow to design experiments with different datasets and computational intelligence algorithms (paying special attention to evolutionary algorithms) in order to assess the behavior of different extraction selection computational methods or complete existing and educational menu.

Fig. 1 Experimental setup for comparing methods in KEEL Tool



```

graph LR
    i[i] --> ii[ii]
    ii --> iii[iii]
    iii --> iv1[Vis-Ins-General]
    iii --> iv2[Vis-Ins-Tabular]
  
```

KEEL Platform (Datasets)



Name ▼	#Attributes (R/I/N) ▼	#Examples ▼	IR ▼	Data set	5-fcv	Header
glass1	9 (9/0/0)	214	1.82			
ecoli-0_vs_1	7 (7/0/0)	220	1.86			
wisconsin	9 (0/9/0)	683	1.86			
pima	8 (8/0/0)	768	1.87			
iris0	4 (4/0/0)	150	2			
glass0	9 (9/0/0)	214	2.06			
yeast1	8 (8/0/0)	1484	2.46			
haberman	3 (0/3/0)	306	2.78			
vehicle2	18 (0/18/0)	846	2.88			
vehicle1	18 (0/18/0)	846	2.9			
vehicle3	18 (0/18/0)	846	2.99			
glass-0-1-2-3_vs_4-5-6	9 (9/0/0)	214	3.2			

References and Further Reading

- Chawla, Nitesh V., et al. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002): 321-357.
- He and Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21.9 (2009): 1263-1284.
- López et al. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. *Open problems on intrinsic data characteristics. Expert Systems with Applications* 39.7 (2012): 6585-6608.
- Napierala and Stefanowski. Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems* 46 (2016): 563-597.
- Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in artificial intelligence* 5.4 (2016): 221-232.
- Haixiang et al. Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications* 73 (2017): 220-239.
- Santos et al. Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches. *IEEE Computational Intelligence Magazine* 13.4 (2018): 59-76.
- Santos et al. On the joint-effect of class imbalance and overlap: a critical review. *Artificial Intelligence Review* 55.8 (2022): 6207-6275.
- Vargas et al. Imbalanced data preprocessing techniques for machine learning: a systematic mapping study. *Knowledge and Information Systems* 65.1 (2023): 31-57.
- Santos et al. A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research. *Information Fusion* 89 (2023): 228-253
- Chen et al. A survey on imbalanced learning: latest research, applications and future directions. *Artificial Intelligence Review* 57.6 (2024): 1-51

Tutorial

T03: Imbalanced Data

Artificial Intelligence and Society

Module 03: Imbalanced Data

Miriam Seoane Santos

LIAAD, INESC TEC, FCUP, University of Porto

miriam.santos@fc.up.pt