indeed

Help! My Classes are Imbalanced!

ODSC West 2019
#ODSC

Samuel Taylor
Data Scientist
@SamuelDataT

Malte Wingen

We help people get jobs.

Agenda

What is class imbalance?

Recognition

Solutions

Recommendations

Agenda

What is class imbalance?

Recognition

Solutions

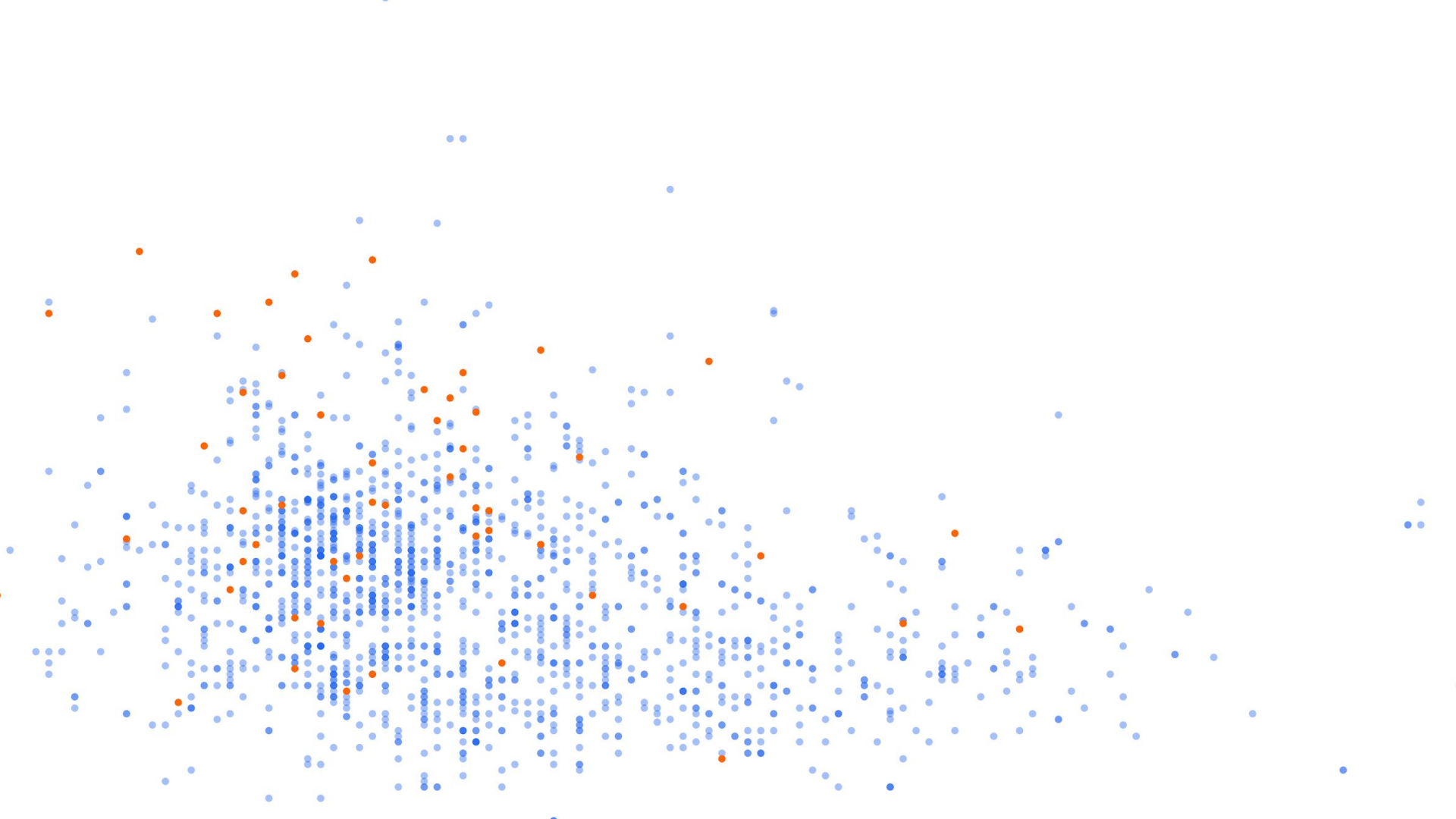
Recommendations

Class imbalance

Class imbalance occurs when **certain values**

Class imbalance occurs when **certain values** of the **target variable**

Class imbalance occurs when **certain values** of the **target variable** are **more common** than others



Causes of class imbalance

CAUSES

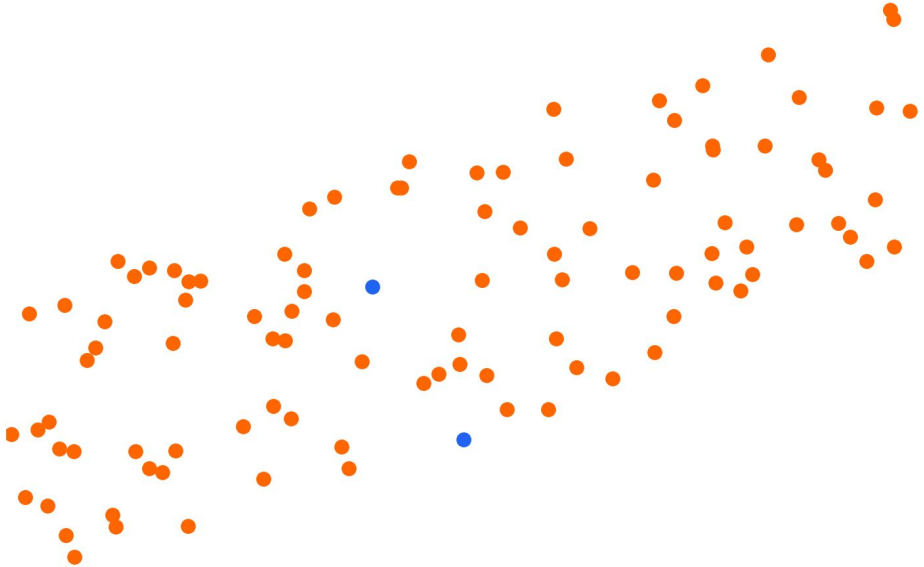
01 Lack of data

02 Overlapping

03 Noise

04 Biased Estimators

Lack of data about the minority class



CAUSES

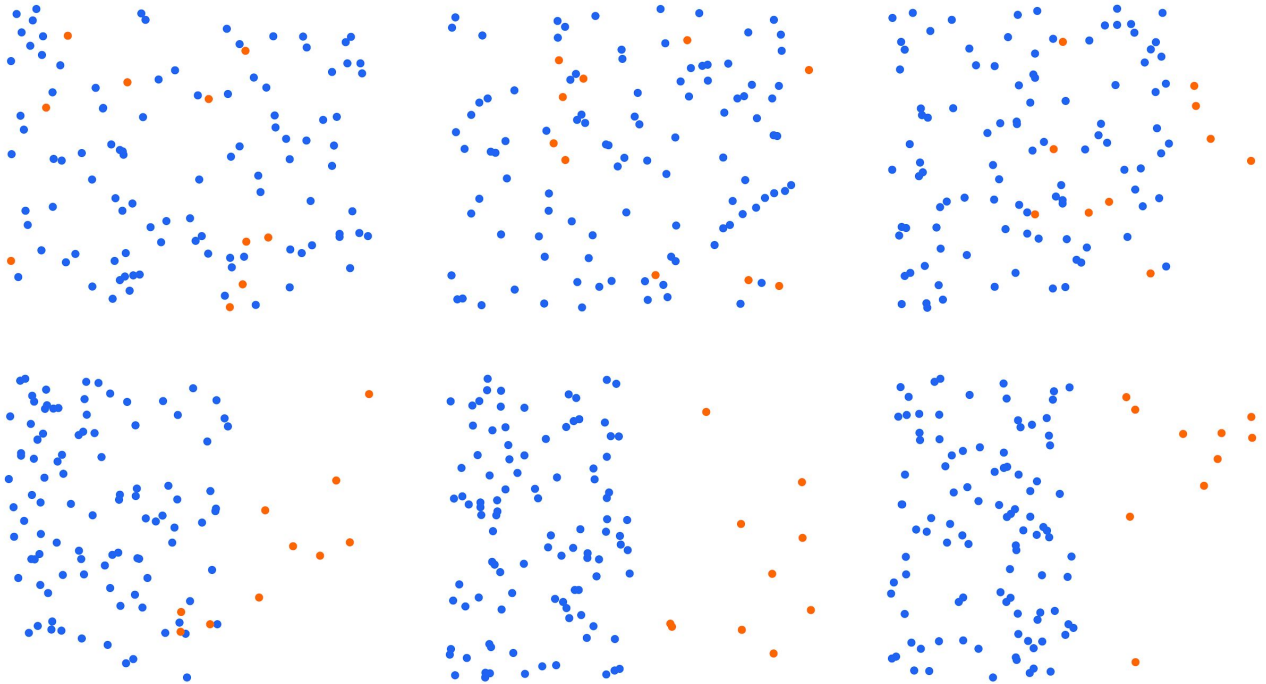
01 Lack of data

02 Overlapping

03 Noise

04 Biased Estimators

Overlapping



CAUSES

01 Lack of data

02 Overlapping

03 Noise

04 Biased Estimators

Noise

CAUSES

01 Lack of data

02 Overlapping

03 Noise

04 Biased Estimators

Noise

CAUSES

01 Lack of data

02 Overlapping

03 Noise

04 Biased Estimators

Noise

CAUSES

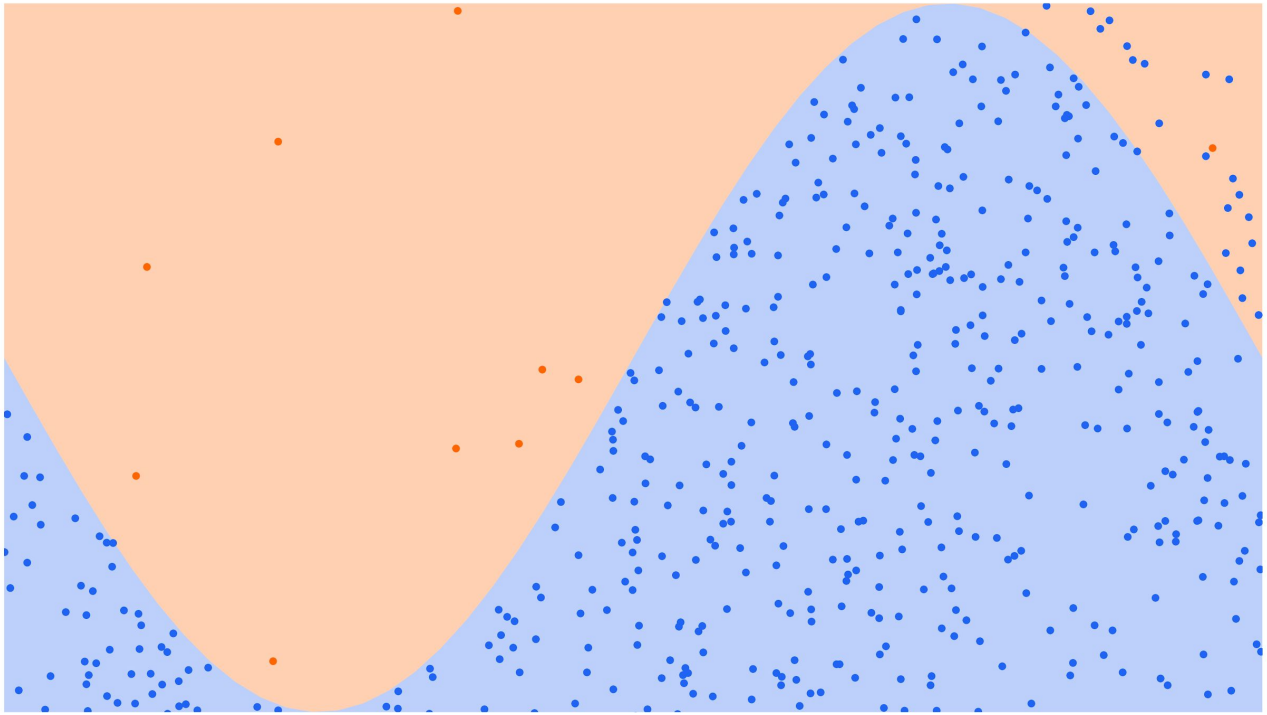
01 Lack of data

02 Overlapping

03 Noise

04 Biased Estimators

Noise



CAUSES

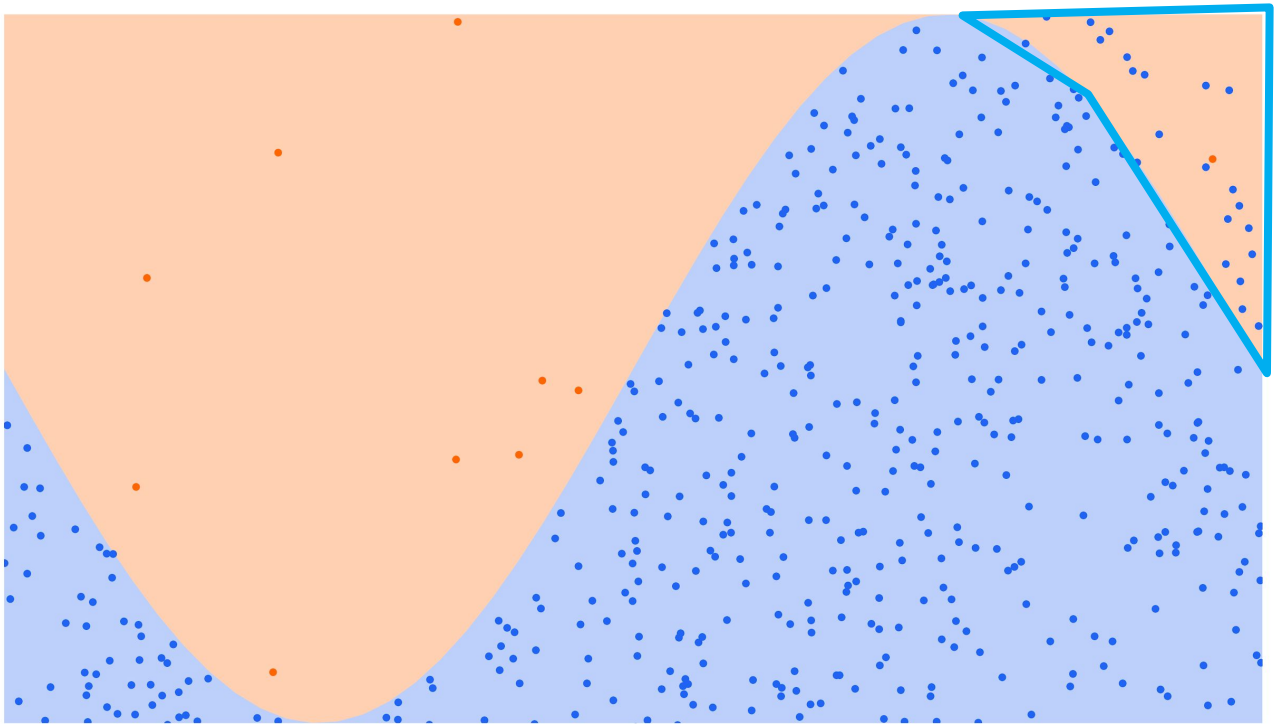
01 Lack of data

02 Overlapping

03 Noise

04 Biased Estimators

Noise



CAUSES

01 Lack of data

02 Overlapping

03 Noise

04 Biased estimators

Biased estimators

CAUSES

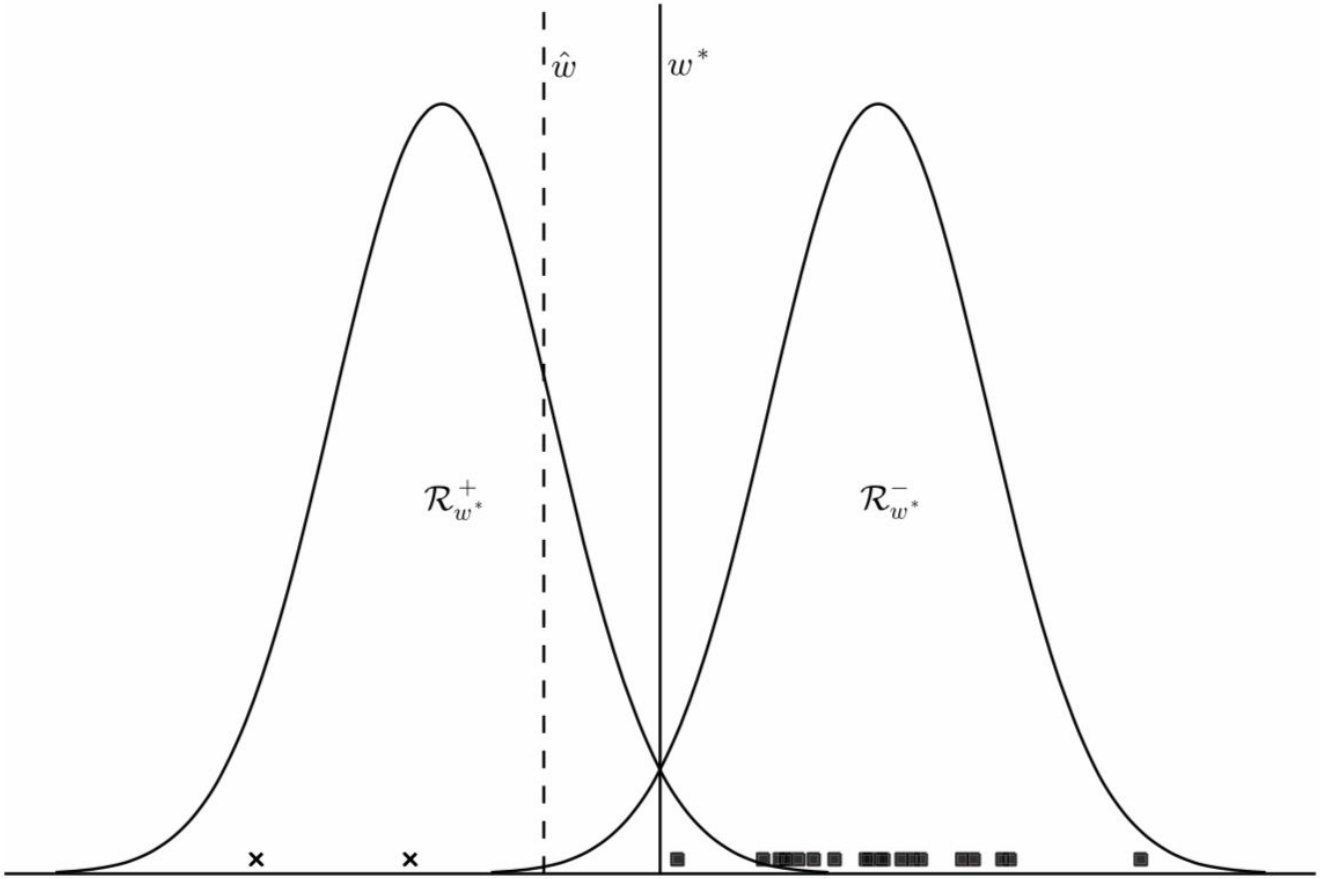
01 Lack of data

02 Overlapping

03 Noise

04 Biased estimators

Biased estimators



CAUSES

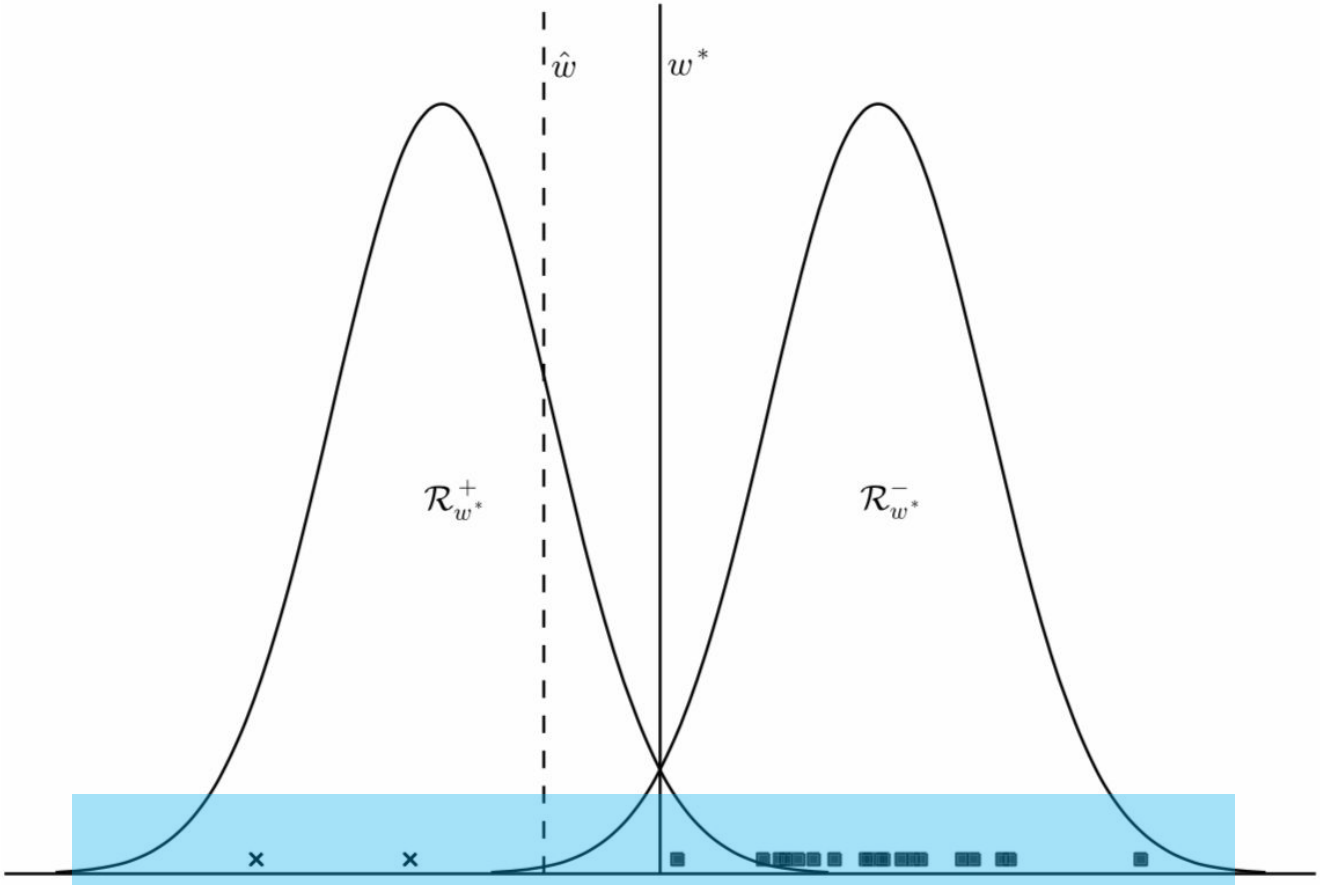
01 Lack of data

02 Overlapping

03 Noise

04 Biased estimators

Biased estimators



CAUSES

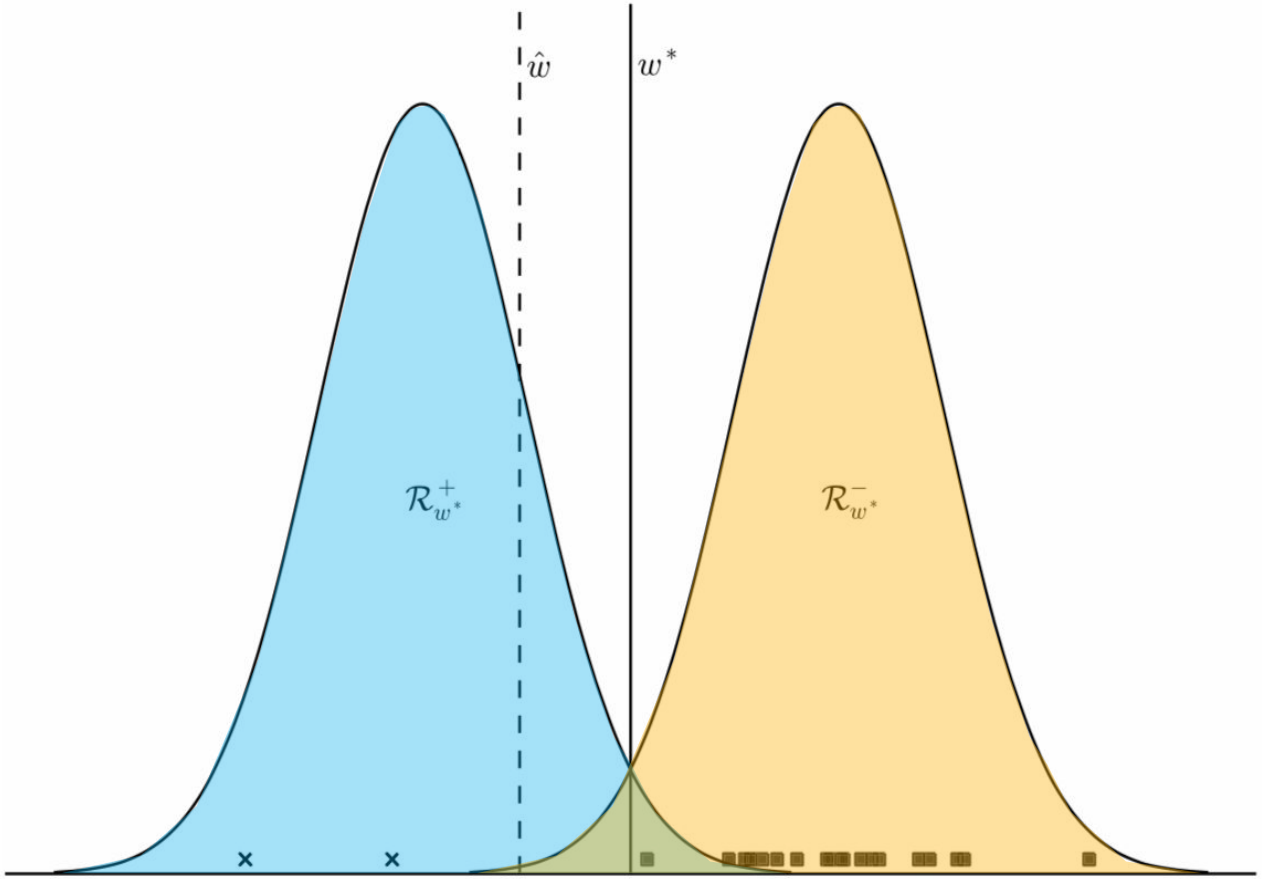
01 Lack of data

02 Overlapping

03 Noise

04 Biased estimators

Biased estimators



CAUSES

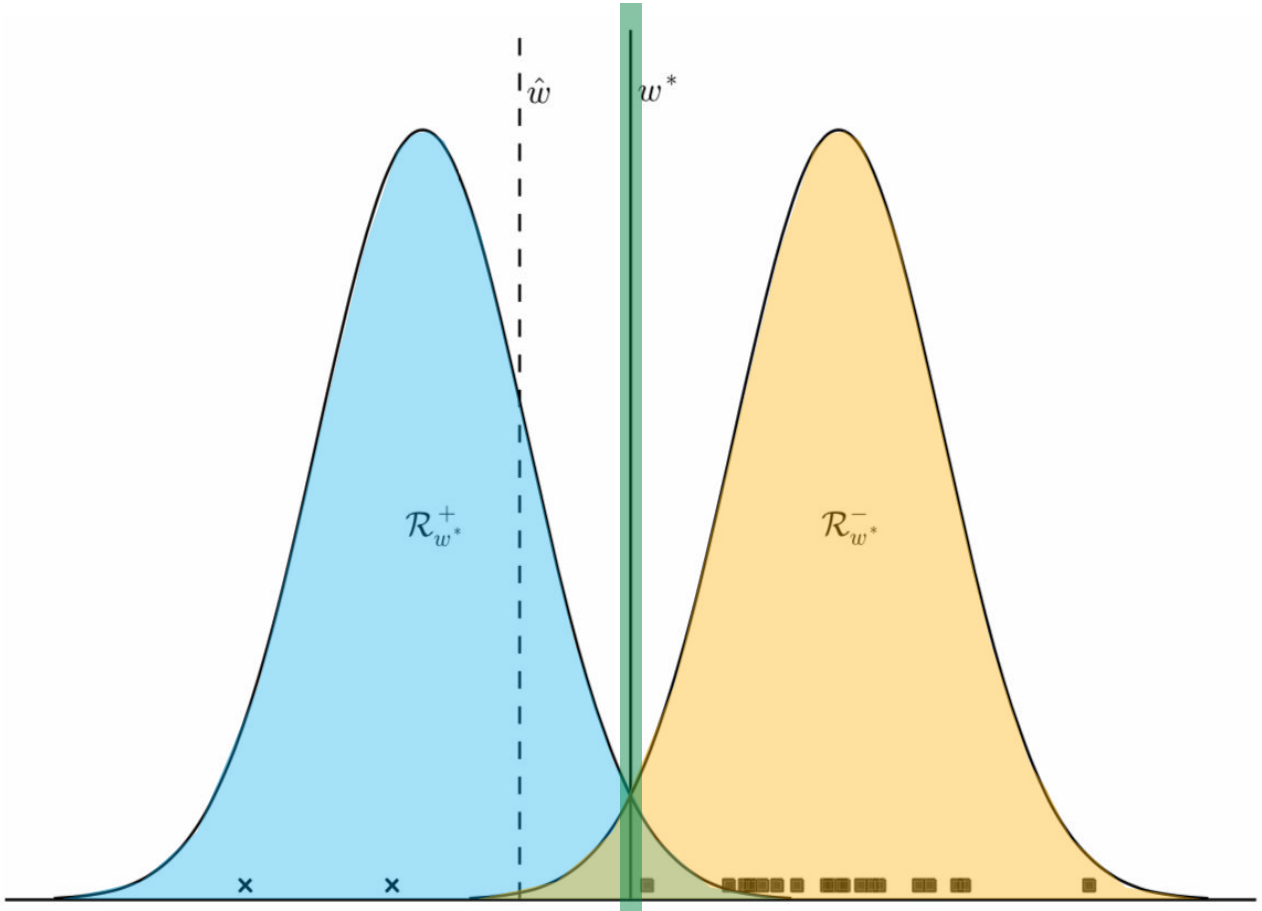
01 Lack of data

02 Overlapping

03 Noise

04 Biased estimators

Biased estimators



CAUSES

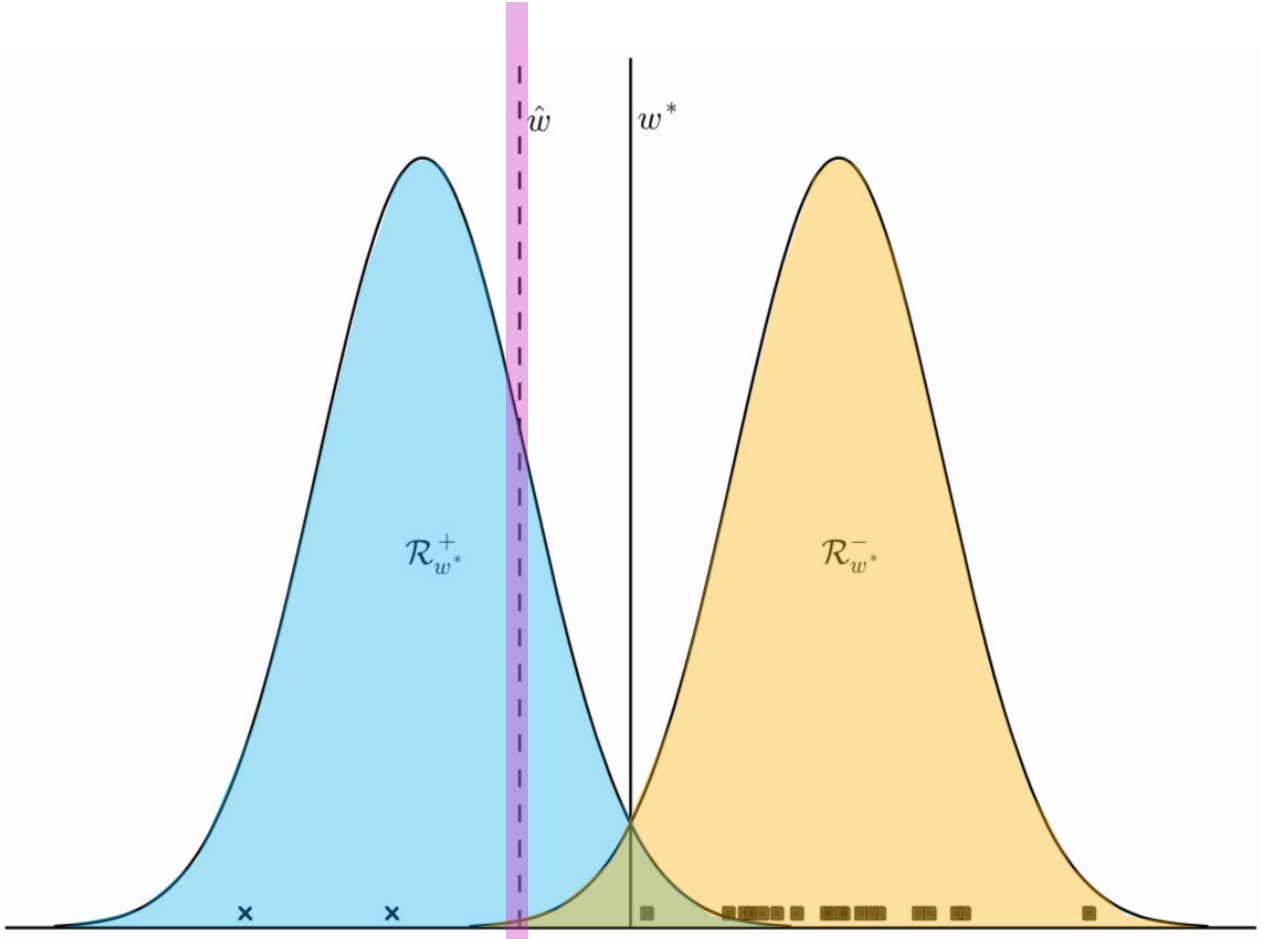
01 Lack of data

02 Overlapping

03 Noise

04 Biased estimators

Biased estimators



Agenda

What is class imbalance?

Recognition

Solutions

Recommendations

RECOGNIZING IT

01 Check for it

02 Compare it

03 Use better metrics

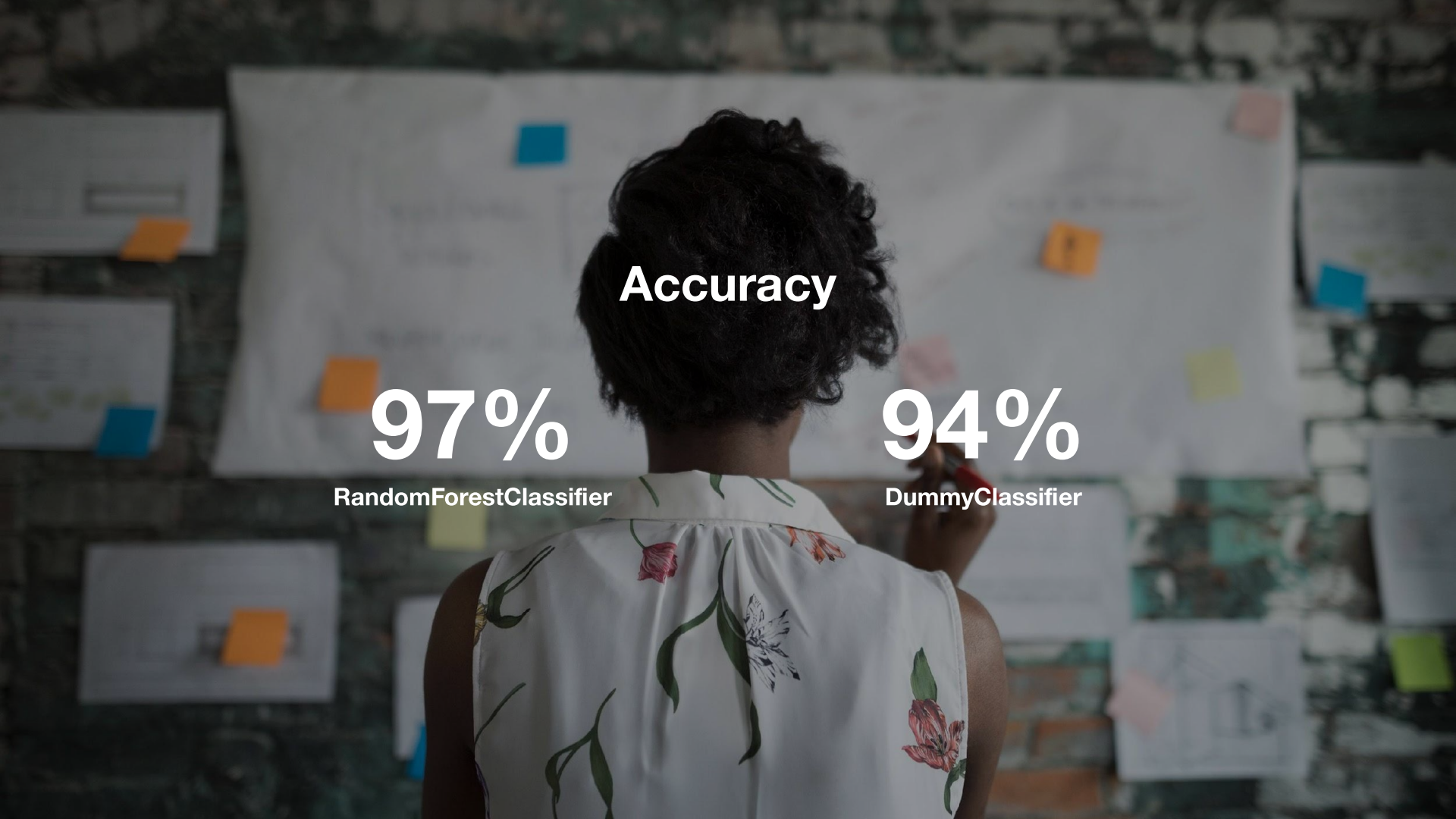
04 Be careful with train/test splits

Explicitly check for it

```
df['class'].value_counts()

negative    1546
positive      53
Name: class, dtype: int64
```


Accuracy
97%
RandomForestClassifier


Accuracy
97%
RandomForestClassifier
94%
DummyClassifier

RECOGNITION

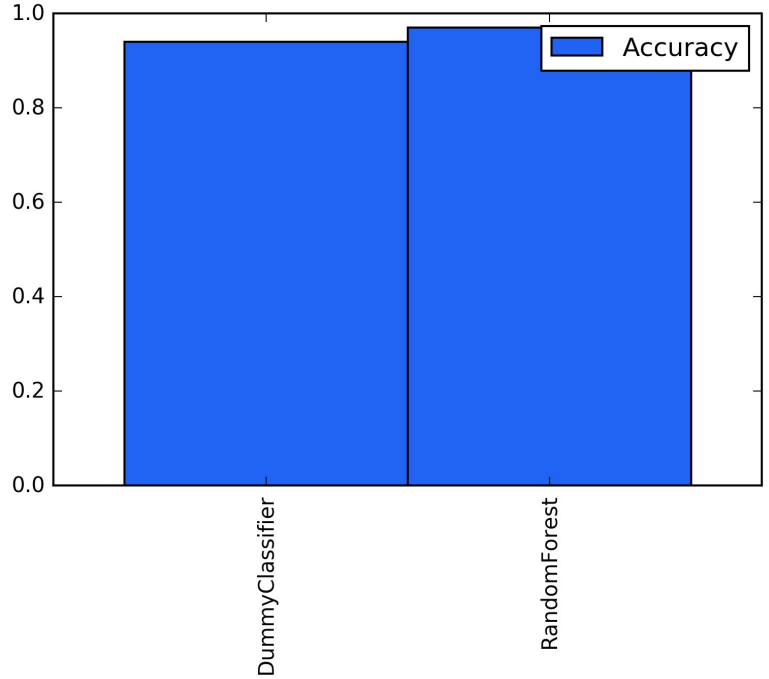
01 Check for it

02 Compare it

03 Use better metrics

04 Be careful with train/test splits

Compare to an incredibly simple baseline



```
from sklearn.dummy import DummyClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

dumb_model = DummyClassifier().fit(X_train, y_train)
y_pred = dumb_model.predict(X_test)
dumb_accuracy = accuracy_score(y_test, y_pred)  # 0.9375

fancy_model = RandomForestClassifier().fit(X_train, y_train)
y_pred = fancy_model.predict(X_test)
fancy_accuracy = accuracy_score(y_test, y_pred) # 0.9675
```

RECOGNITION

01 Check for it

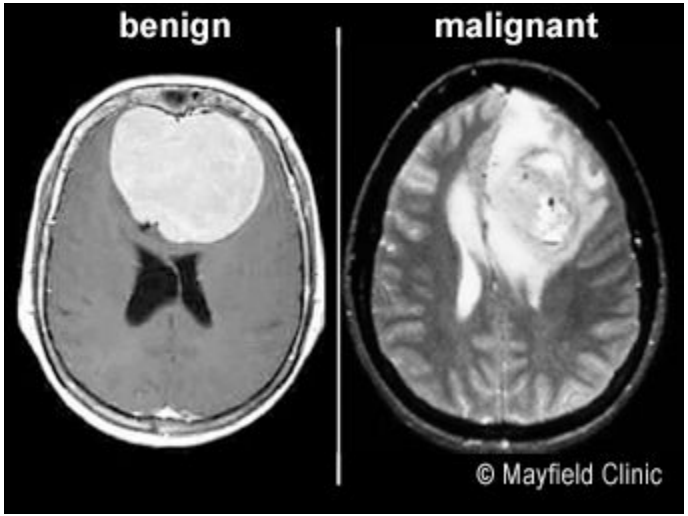
02 Compare it

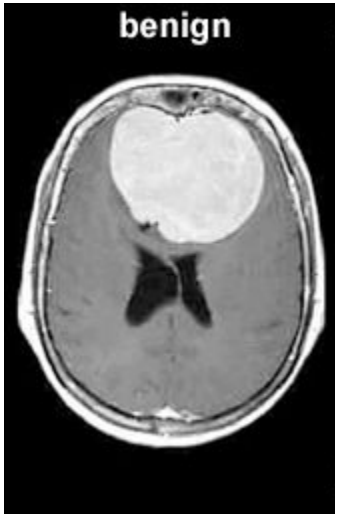
03 Use better metrics

04 Be careful with train/test splits

Use better metrics

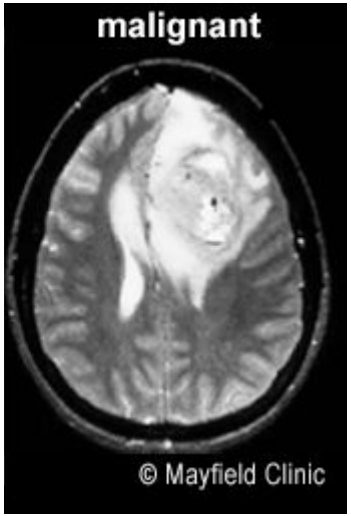
Accuracy assumes all errors are equally costly


benign
malignant
© Mayfield Clinic



Cost of mistake:

- Patient worry
- Further tests



Cost of mistake:

- Death

RECOGNITION

01 Check for it

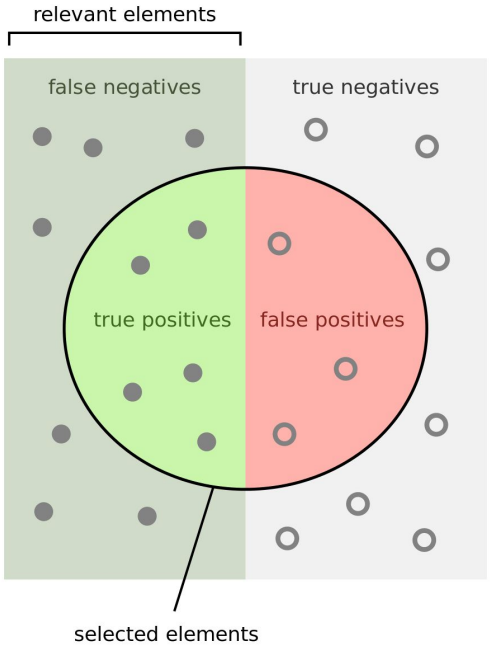
02 Compare it

03 Use better metrics

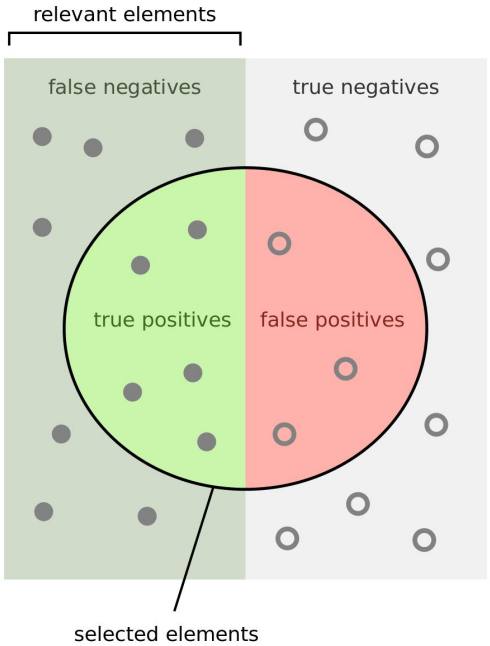
04 Be careful with train/test splits

Use better metrics

Accuracy assumes all errors are equally costly


relevant elements
false negatives
true negatives
true positives
false positives
selected elements

Wikipedia



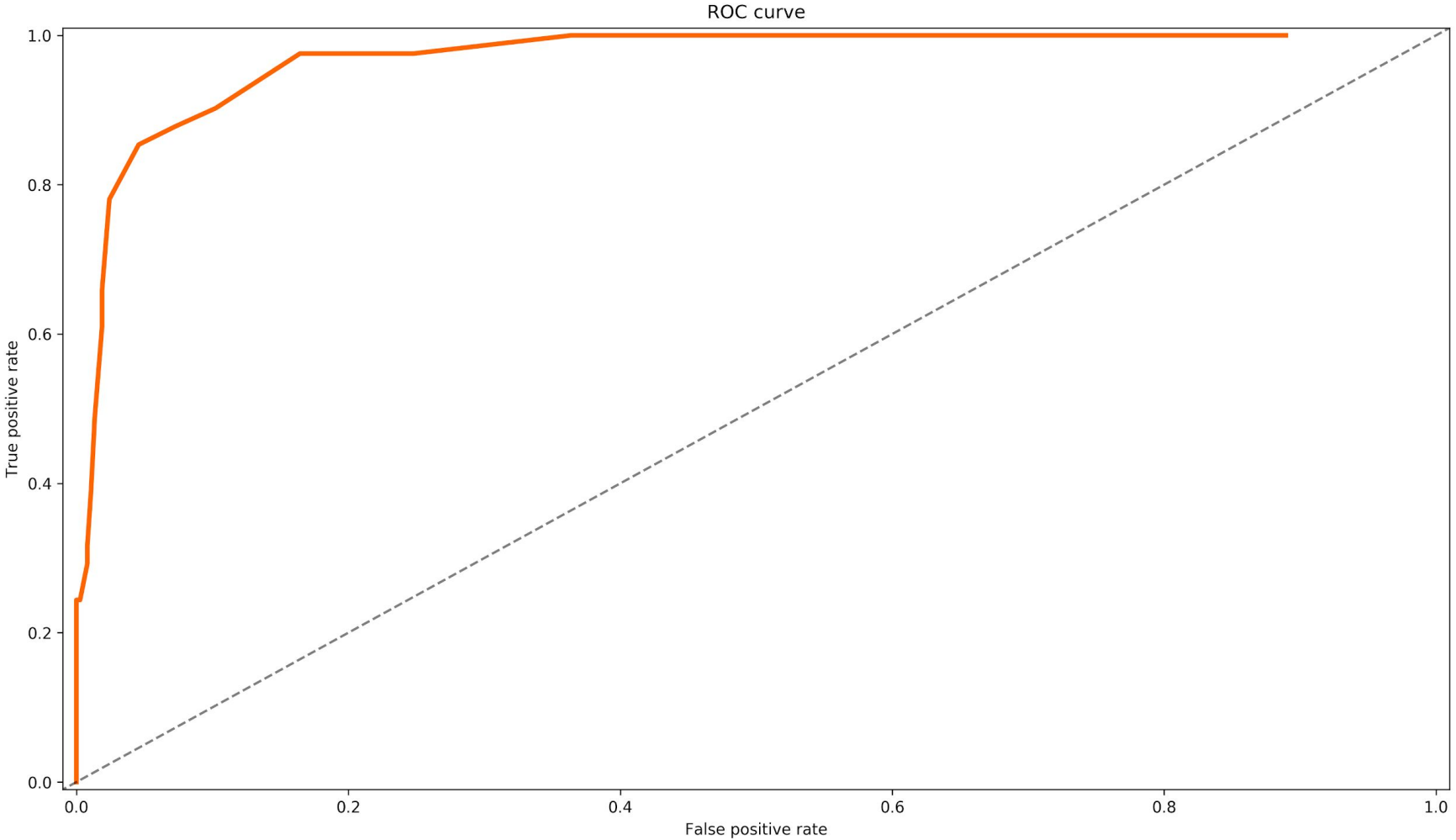
How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Wikipedia


ROC curve
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

“

ROC curves… can be used to evaluate classifier performance when prior probabilities and misclassification costs are difficult to estimate a priori

– *Sinha and May*

RECOGNITION

01 Check for it

02 Compare it

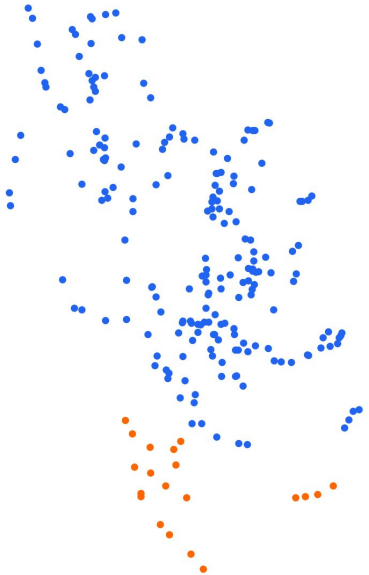
03 Use better metrics

04 Be careful with train/test splits

Be careful with train/test splits

Minority prevalance: 4.8%

Minority prevalance: 9.1%



RECOGNITION

01 Check for it

02 Compare it

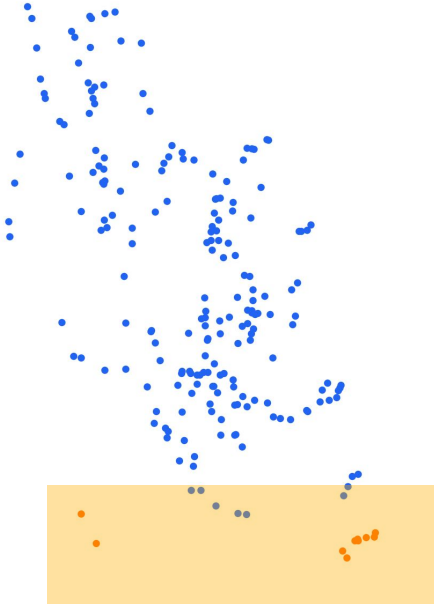
03 Use better metrics

04 Be careful with train/test splits

Be careful with train/test splits

Minority prevalance: 4.8%

Minority prevalance: 4.8%



Agenda

What is class imbalance?

Recognition

Solutions

Recommendations

Gather more data

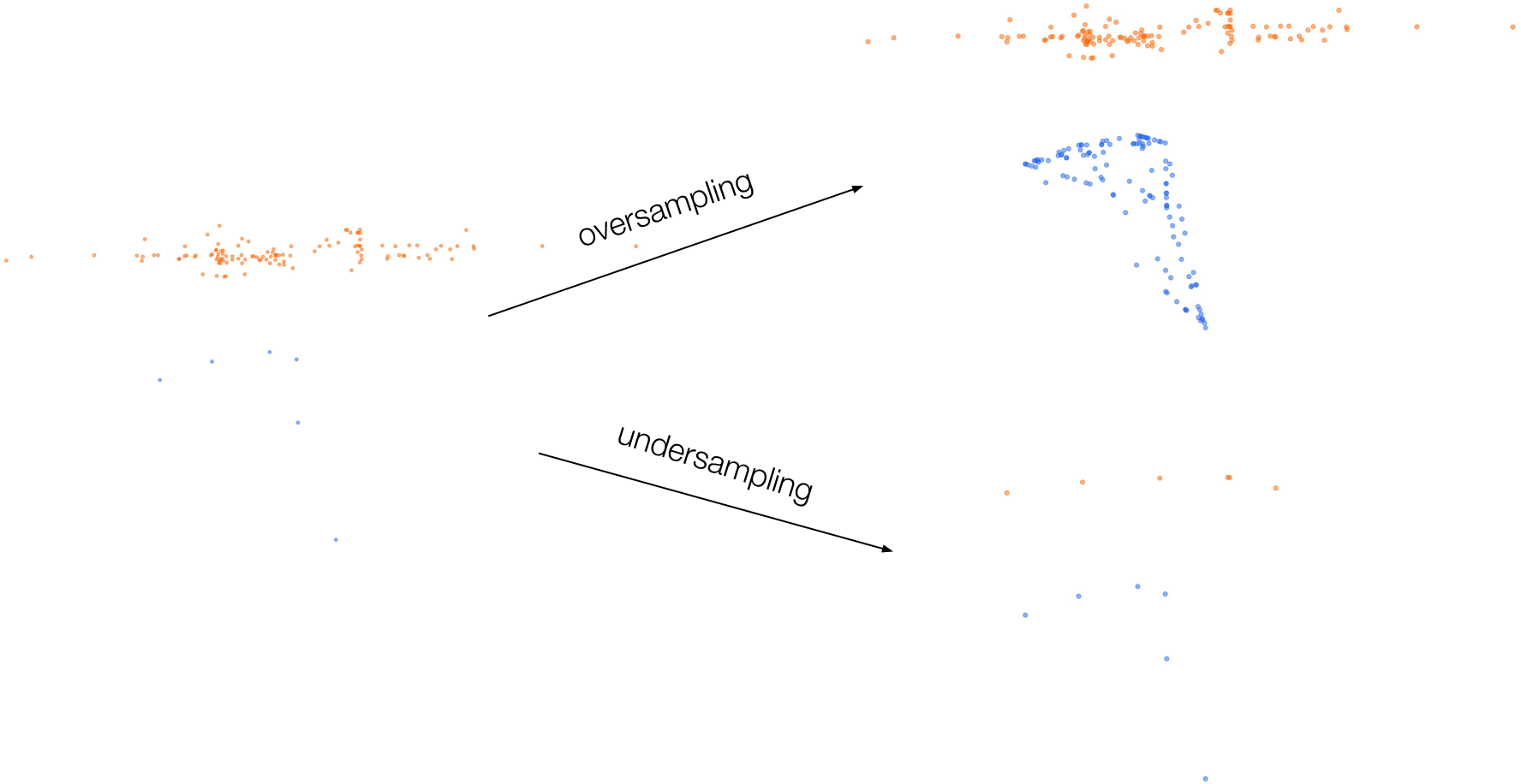
Taxonomy from Branco, Torgo & Ribeiro

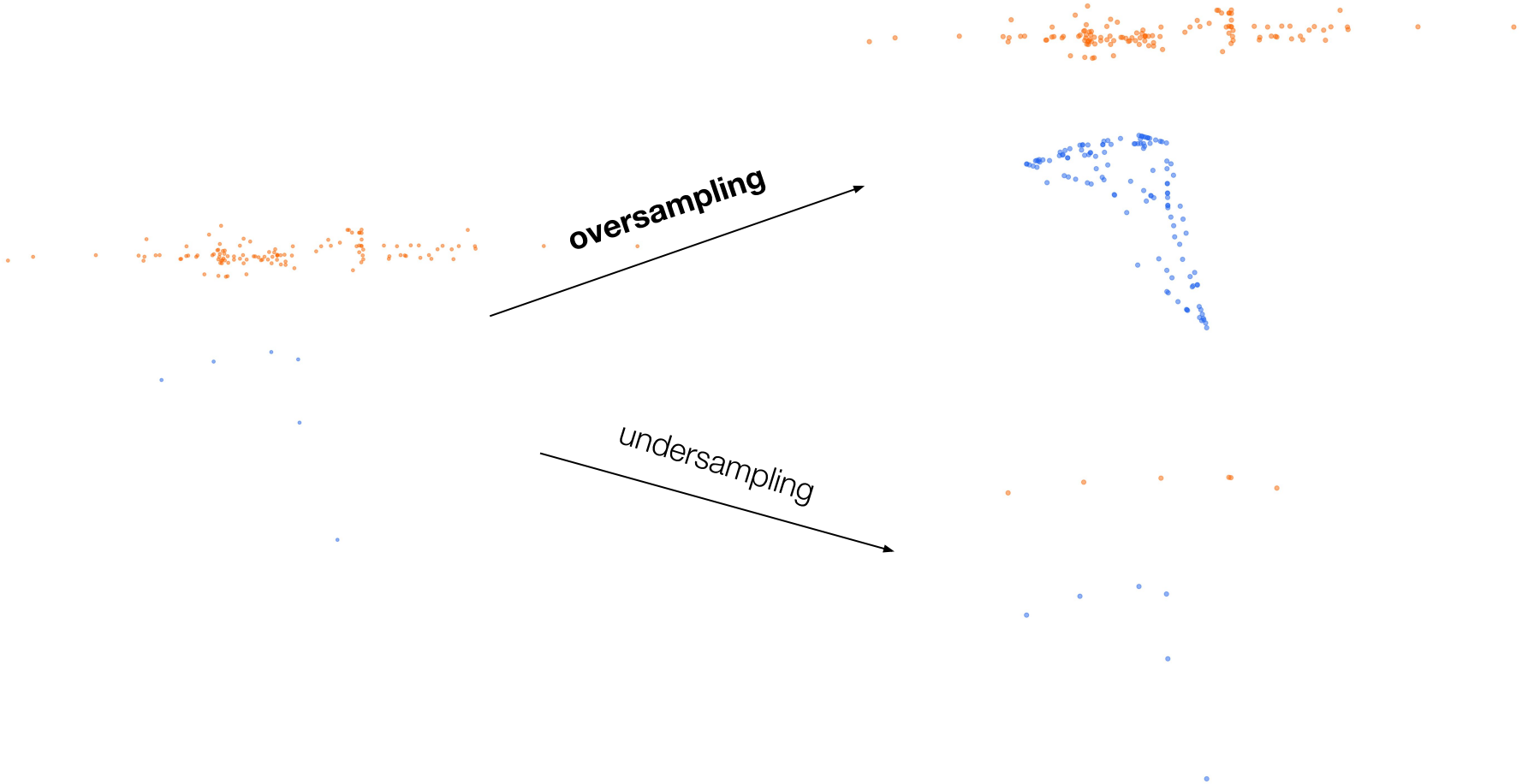
Pre-processing

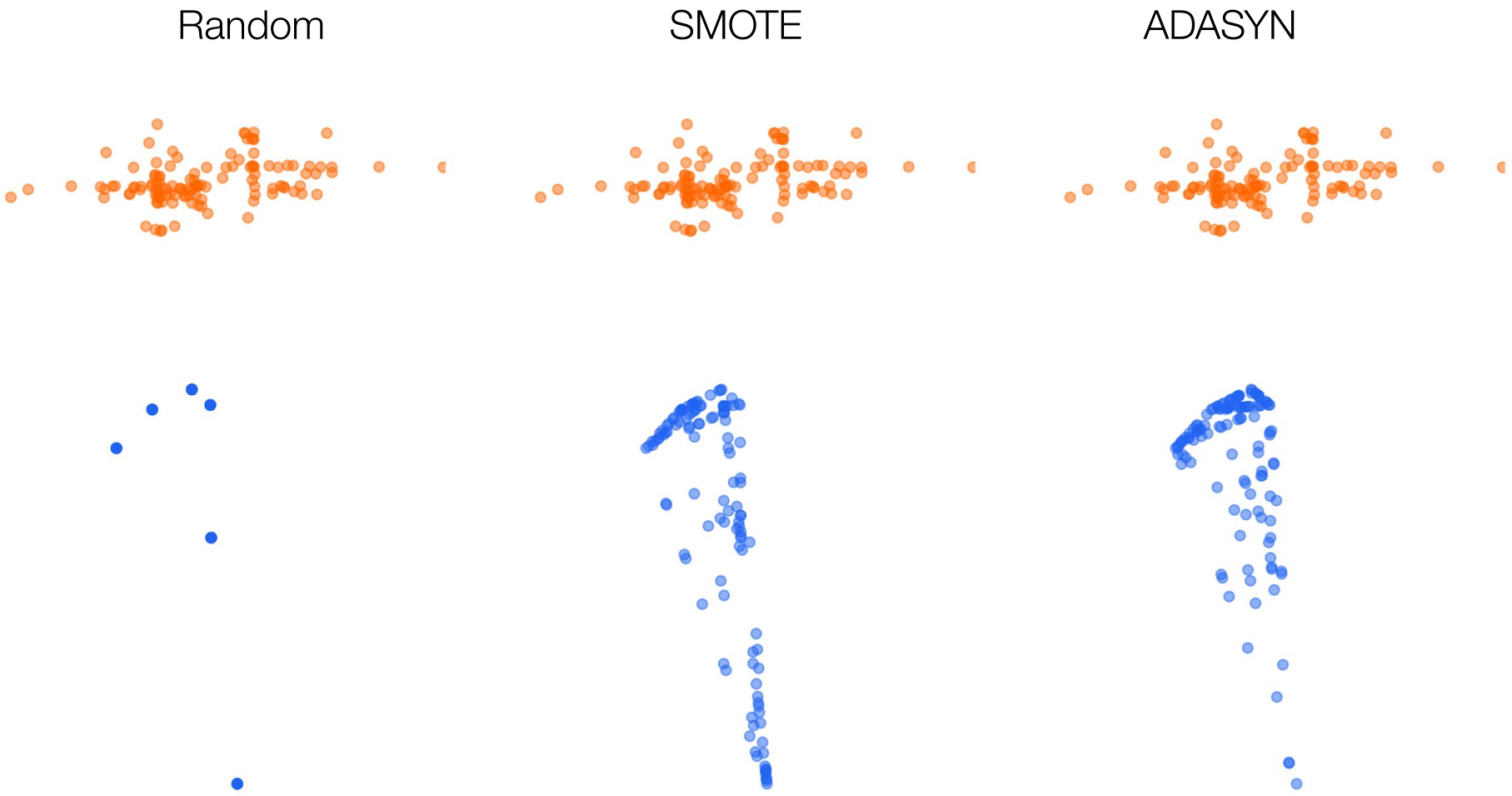
Special-purpose learning

Prediction post-processing

01
Pre-processing


oversampling
undersampling


oversampling
undersampling


Random
SMOTE
ADASYN

SMOTE

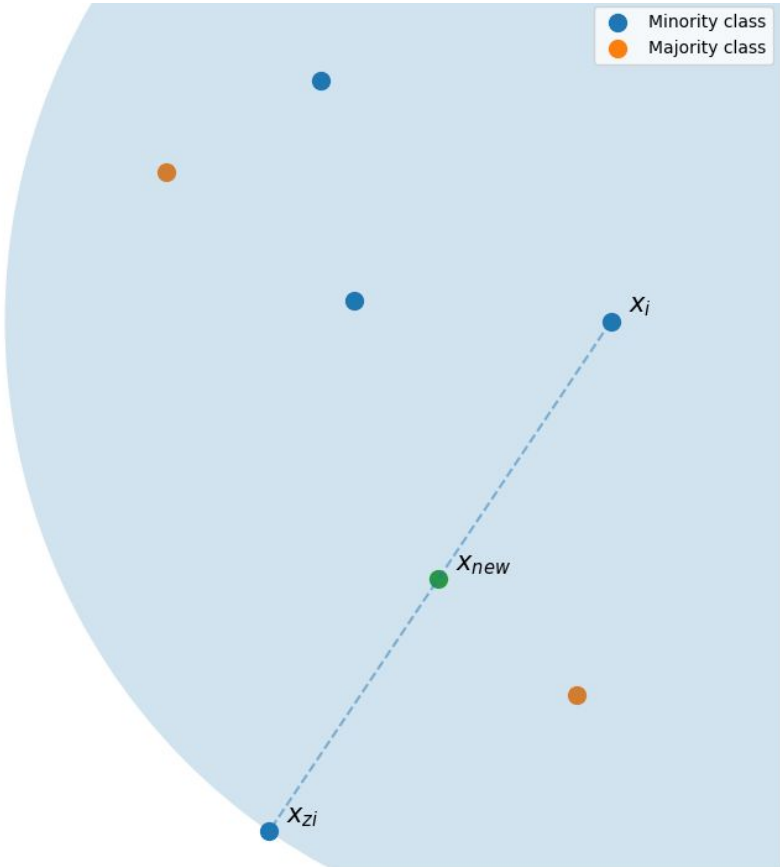
01 Select member of minority class

02 Find its *k* nearest neighbors and select one

03 Interpolate a point *p*% of the way between the two points

(*p* selected randomly on [0, 1])

04 Repeat until desired level of balance



imbalanced-learn

SMOTE

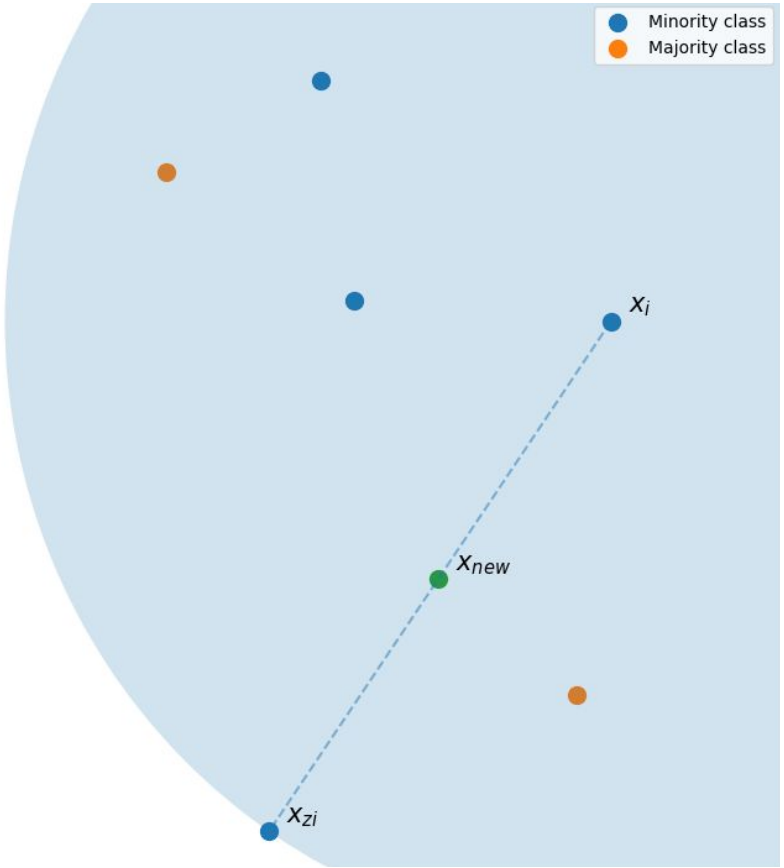
01 **Select member of minority class**

02 **Find its *k* nearest neighbors and select one**

03 **Interpolate a point *p*% of the way between the two points**

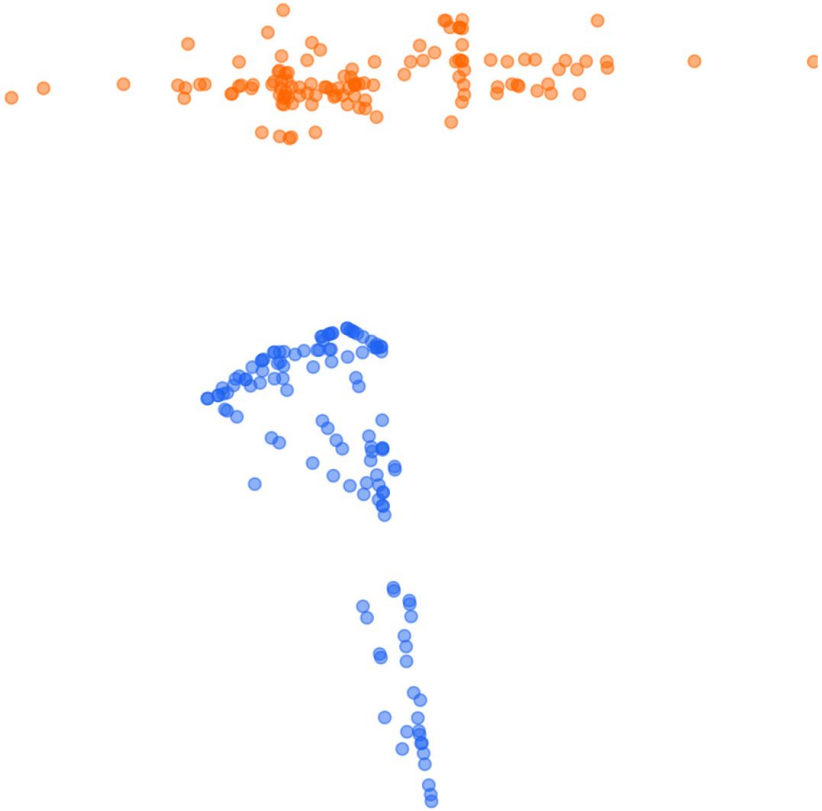
(*p* selected randomly on [0, 1])

04 **Repeat until desired level of balance**

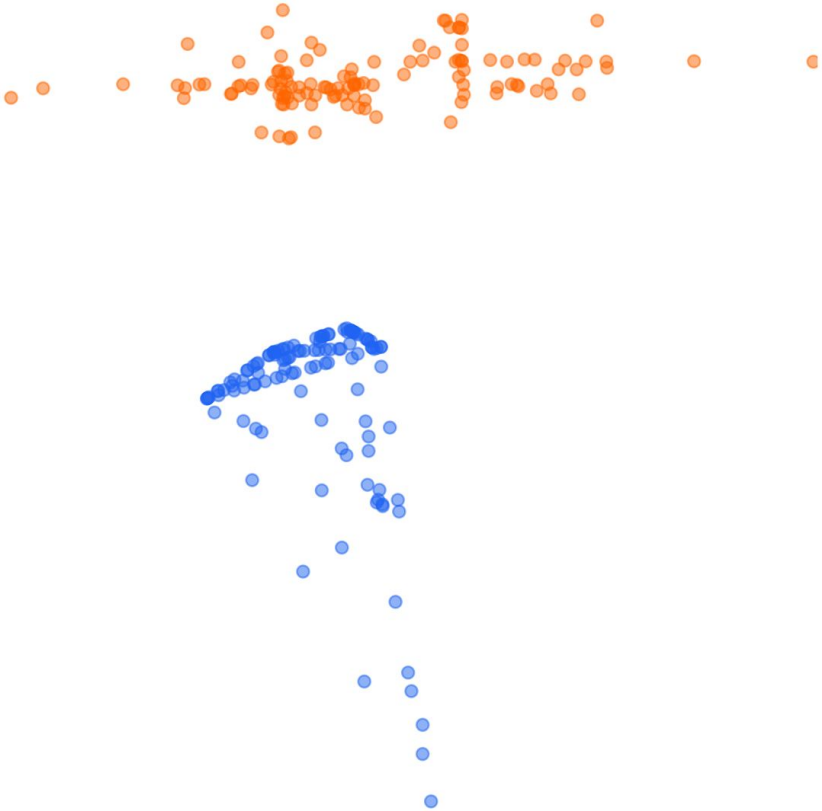


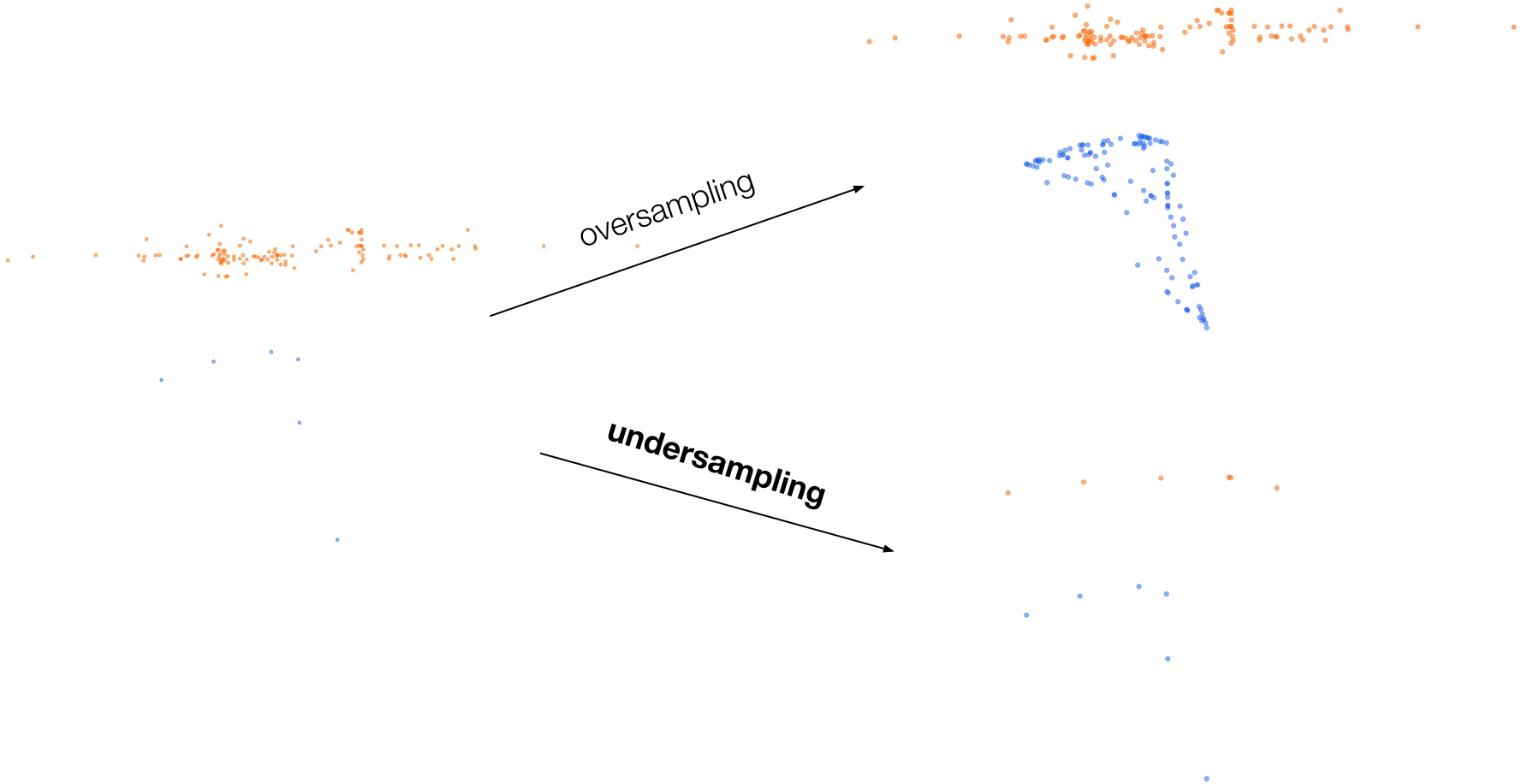
imbalanced-learn

SMOTE

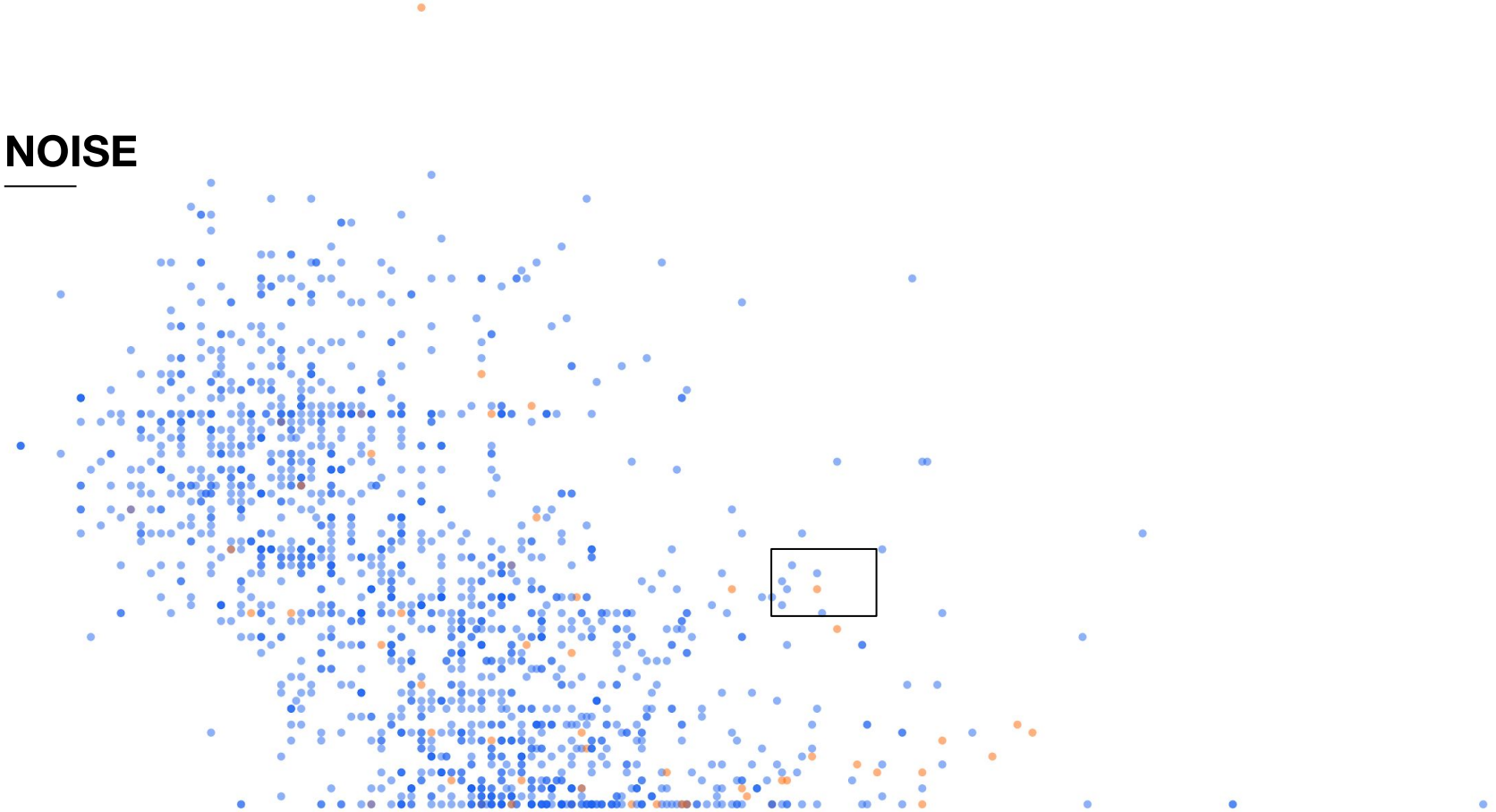


ADASYN

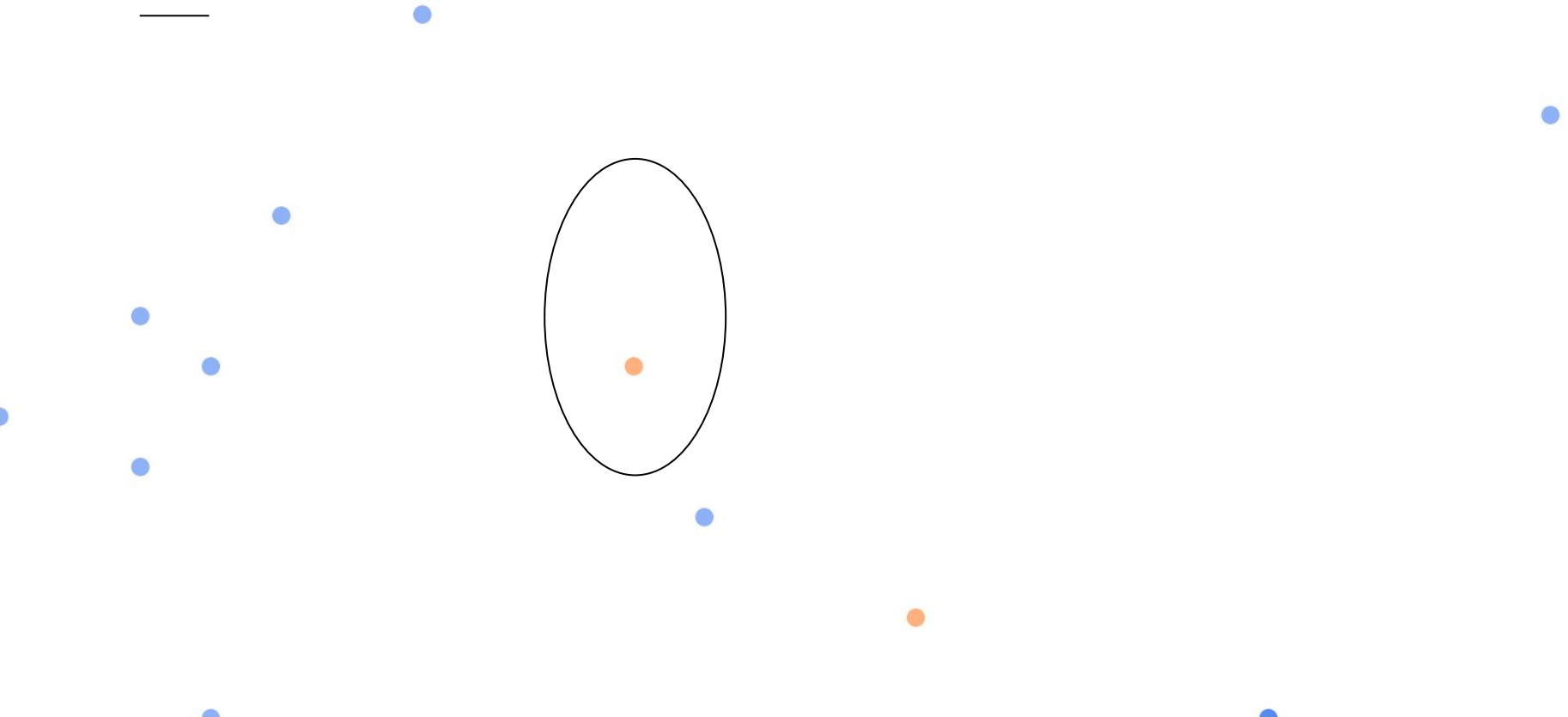



oversampling
undersampling

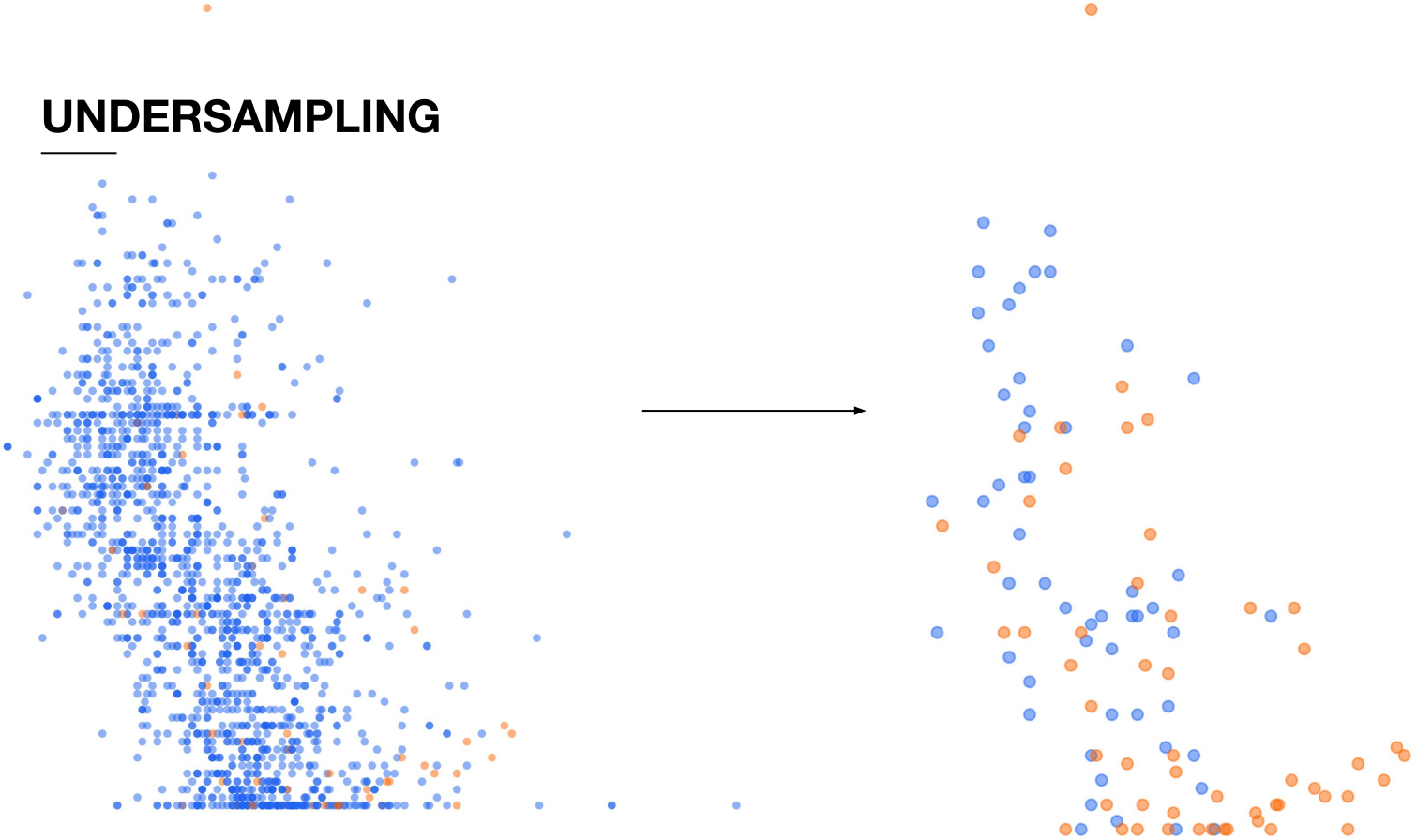
NOISE



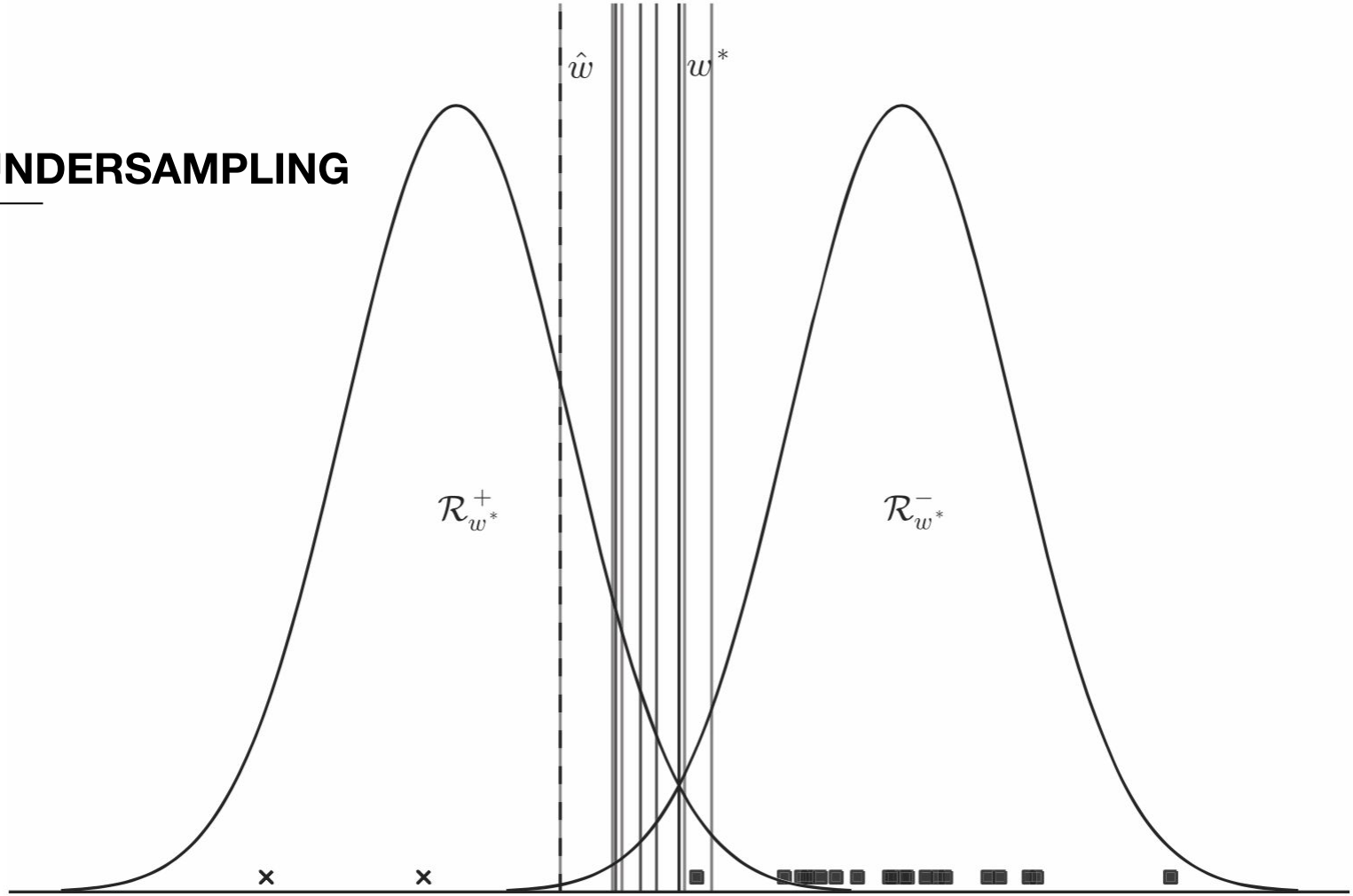
TOMEK LINKS



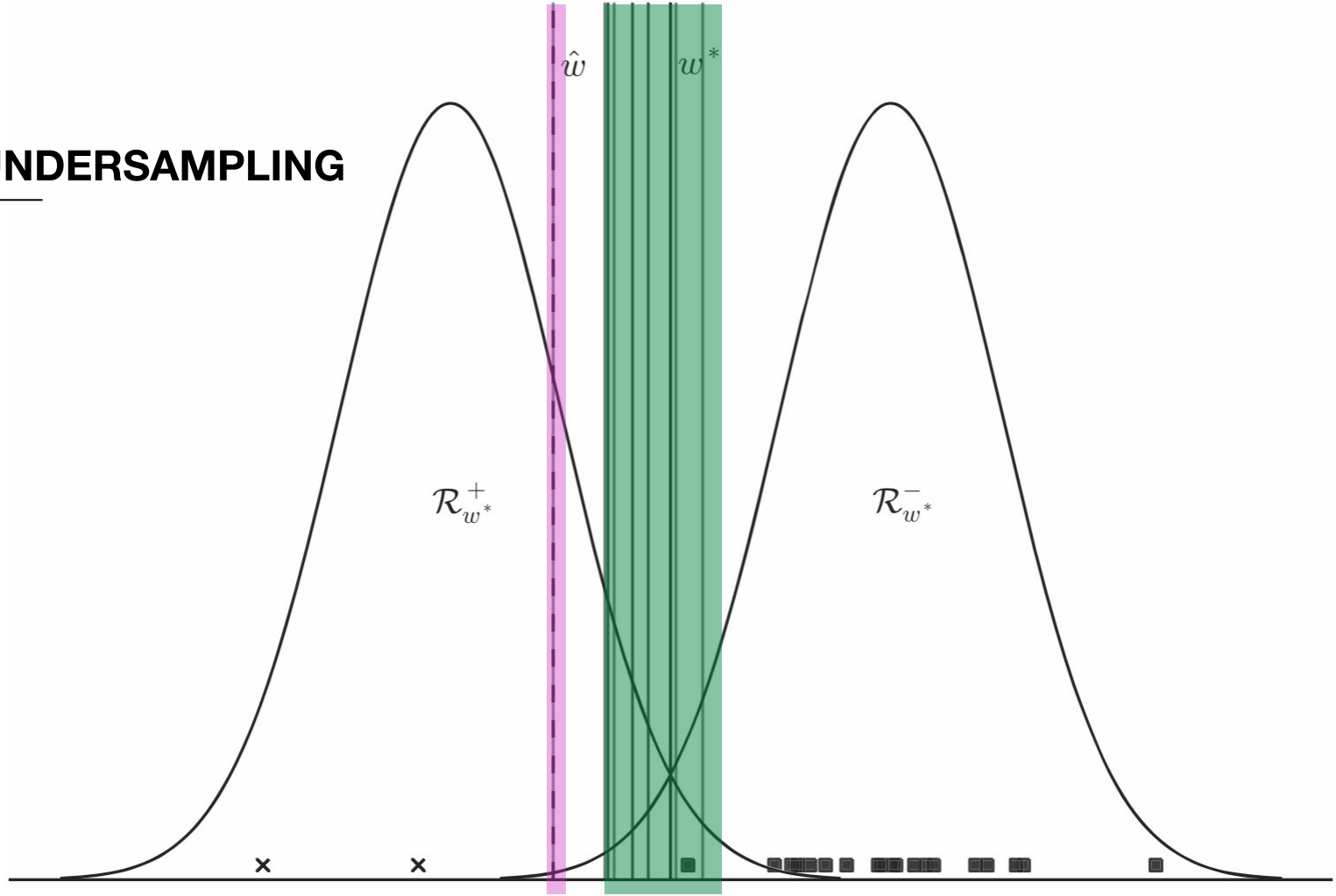
UNDERSAMPLING



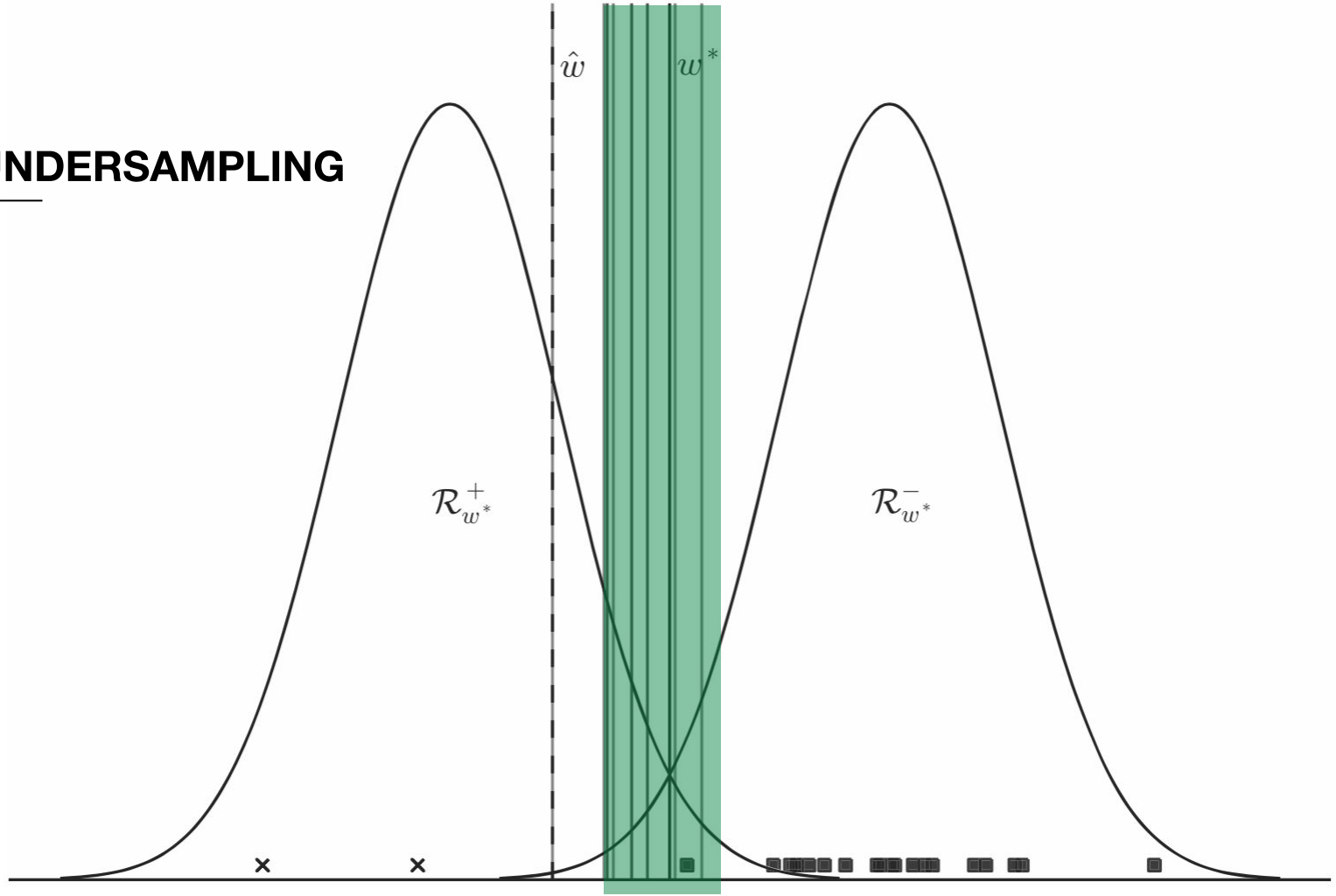
UNDERSAMPLING

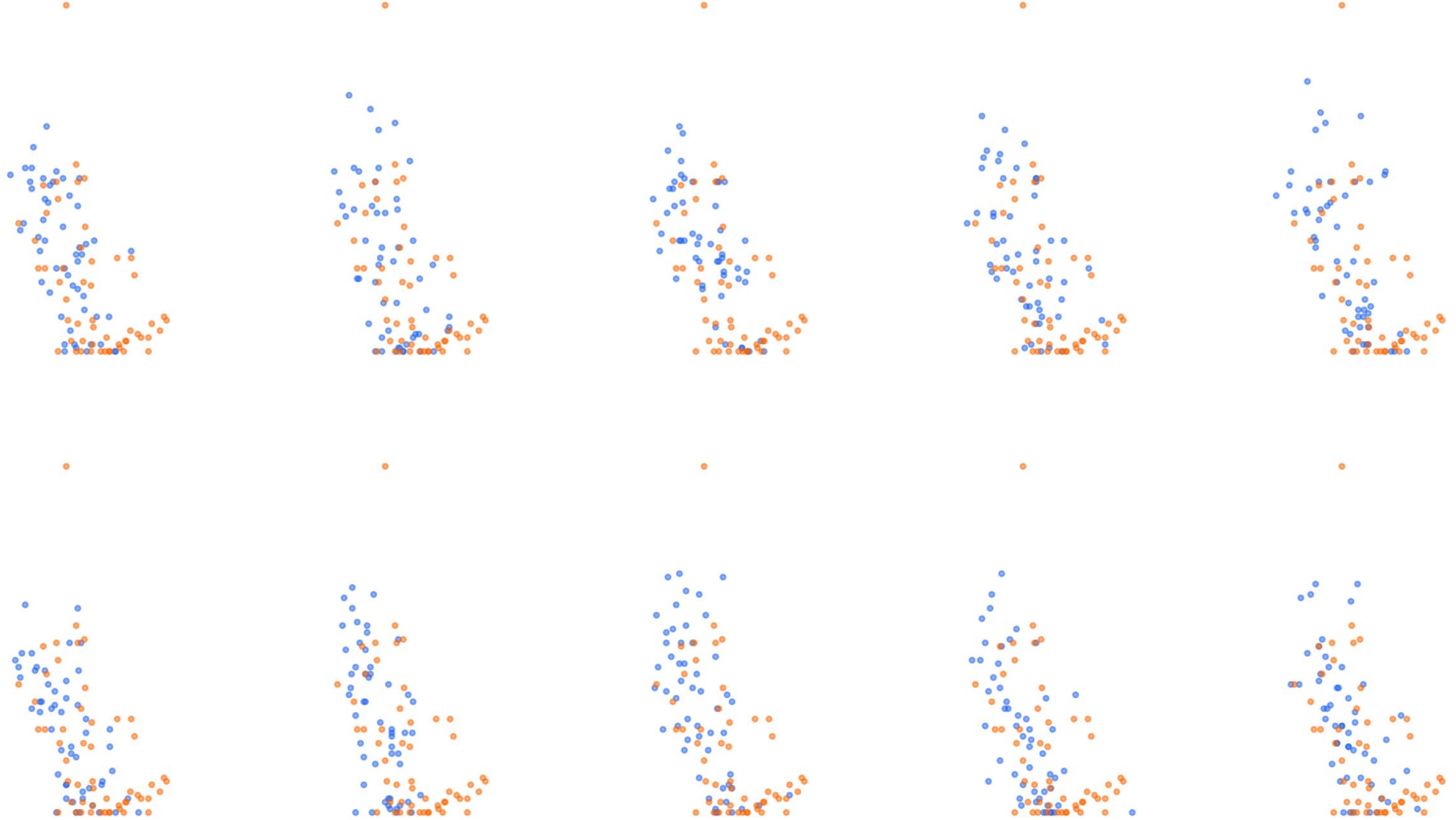


UNDERSAMPLING



UNDERSAMPLING





imbalanced-learn

PRE-PROCESSING

Libraries exist

PRE-PROCESSING

Libraries exist

Biases models toward user desires

PRE-PROCESSING

Libraries exist

Biases models toward user desires

Changes the cost of training a model

PRE-PROCESSING

Libraries exist

Biases models toward user desires

Changes the cost of training a model

Can be difficult to apply

02
Special-purpose learners

class sklearn.ensemble.**RandomForestClassifier**(*n_estimators='warn'*, *criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features='auto'*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *bootstrap=True*, *oob_score=False*, *n_jobs=None*, *random_state=None*, *verbose=0*, *warm_start=False*, ***class_weight=None***)

class sklearn.linear_model.**LogisticRegression**(*penalty='l2'*, *dual=False*, *tol=0.0001*, *C=1.0*, *fit_intercept=True*, *intercept_scaling=1*, ***class_weight=None***, *random_state=None*, *solver='warn'*, *max_iter=100*, *multi_class='warn'*, *verbose=0*, *warm_start=False*, *n_jobs=None*, *l1_ratio=None*)

class sklearn.svm.**SVC**(*C=1.0*, *kernel='rbf'*, *degree=3*, *gamma='auto_deprecated'*, *coef0=0.0*, *shrinking=True*, *probability=False*, *tol=0.001*, *cache_size=200*, ***class_weight=None***, *verbose=False*, *max_iter=-1*, *decision_function_shape='ovr'*, *random_state=None*)

class lightgbm.**LGBMClassifier**(*boosting_type='gbdt', num_leaves=31, max_depth=-1, learning_rate=0.1, n_estimators=100, subsample_for_bin=200000, objective=None, class_weight=None, min_split_gain=0.0, min_child_weight=0.001, min_child_samples=20, subsample=1.0, subsample_freq=0, colsample_bytree=1.0, reg_alpha=0.0, reg_lambda=0.0, random_state=None, n_jobs=-1, silent=True, importance_type='split',* ***scale_pos_weight=1.0****, **kwargs*)

class xgboost.**XGBClassifier**(*max_depth=3, learning_rate=0.1, n_estimators=100, verbosity=1, silent=None, objective='binary:logistic', booster='gbtree', n_jobs=1, nthread=None, gamma=0, min_child_weight=1, max_delta_step=0, subsample=1, colsample_bytree=1, colsample_bylevel=1, colsample_bynode=1, reg_alpha=0, reg_lambda=1,* ***scale_pos_weight=1****, base_score=0.5, random_state=0, seed=None, missing=None, **kwargs*)

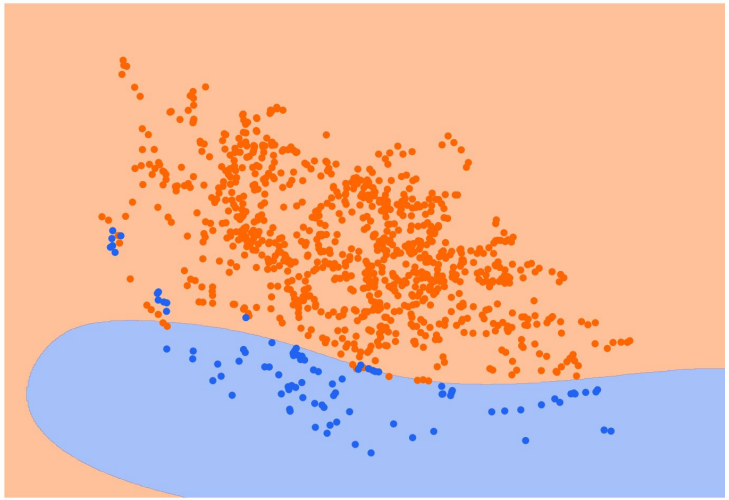
Weighting in tree models affects

Weighting in tree models affects impurity calculations

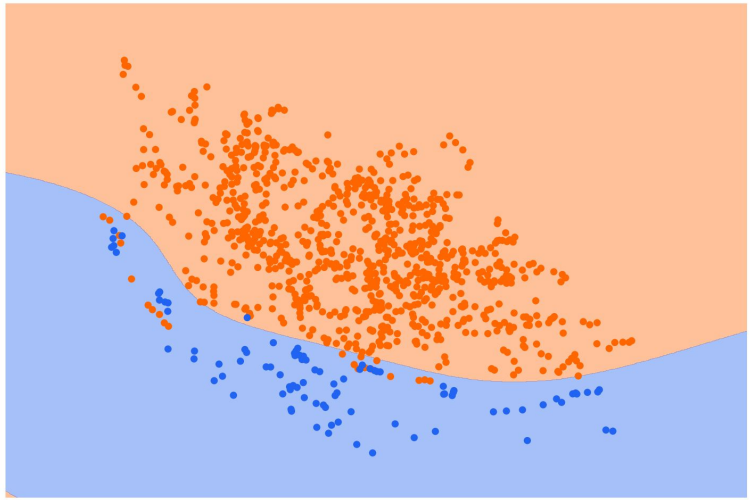
Weighting in tree models affects impurity calculations and prediction-time voting

Weighting in SVM's pushes the hyperplane away from the minority class

SVM, minority class weight: 1 (default)



SVM, minority class weight: 4



Weighting in logistic regression pushes the hyperplane away from the minority class

Weighting in kNN changes the distance metric

Weighting in _____ changes _____

When does this work?

Weighting is less effective under high imbalance

> **As the degree of imbalance increases… the probability that using weighted empirical cost minimization to counter imbalance will be effective in reducing bias decreases.**
>
> **– Wallace et al.**

Weighting is more effective with more data

> [A]s the size of the training set increases, such strategies [i.e. class weighting] will become more effective, in general
>
> – Wallace et al.

SPECIAL-PURPOSE LEARNERS

Directly addresses the issue

SPECIAL-PURPOSE LEARNERS

Directly addresses the issue

Requires knowledge of cost/benefit

SPECIAL-PURPOSE LEARNERS

Directly addresses the issue

Requires knowledge of cost/benefit

Effective when closer to balance, with lots of data

SPECIAL-PURPOSE LEARNERS

Directly addresses the issue

Requires knowledge of cost/benefit

Effective when closer to balance, with lots of data

Difficult (if not already supported)

03
Prediction post-processing

POST-PROCESSING

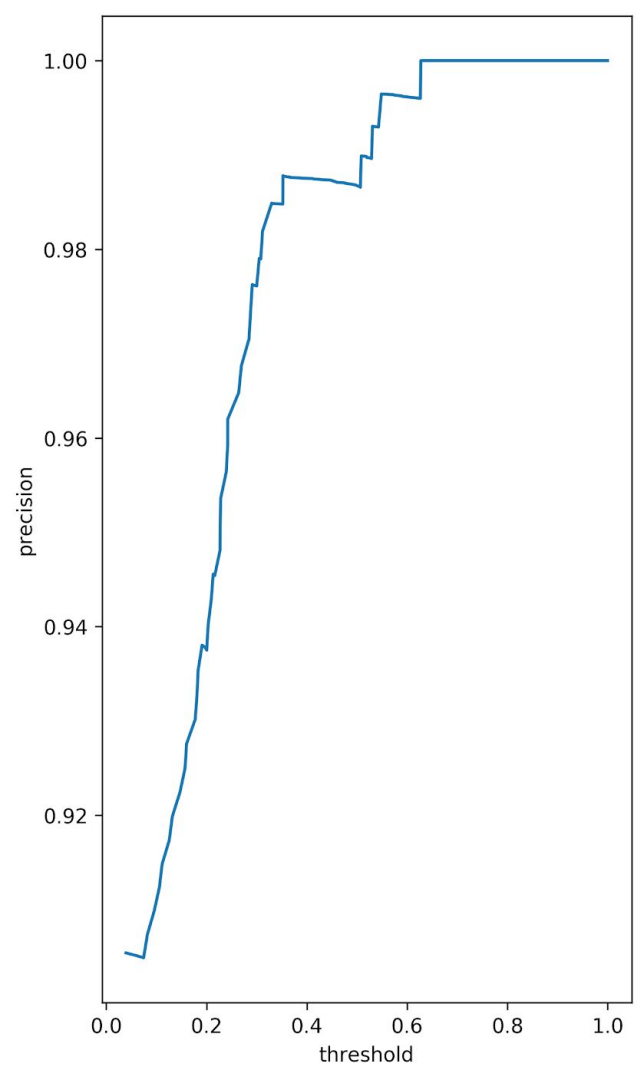
+ Threshold selection

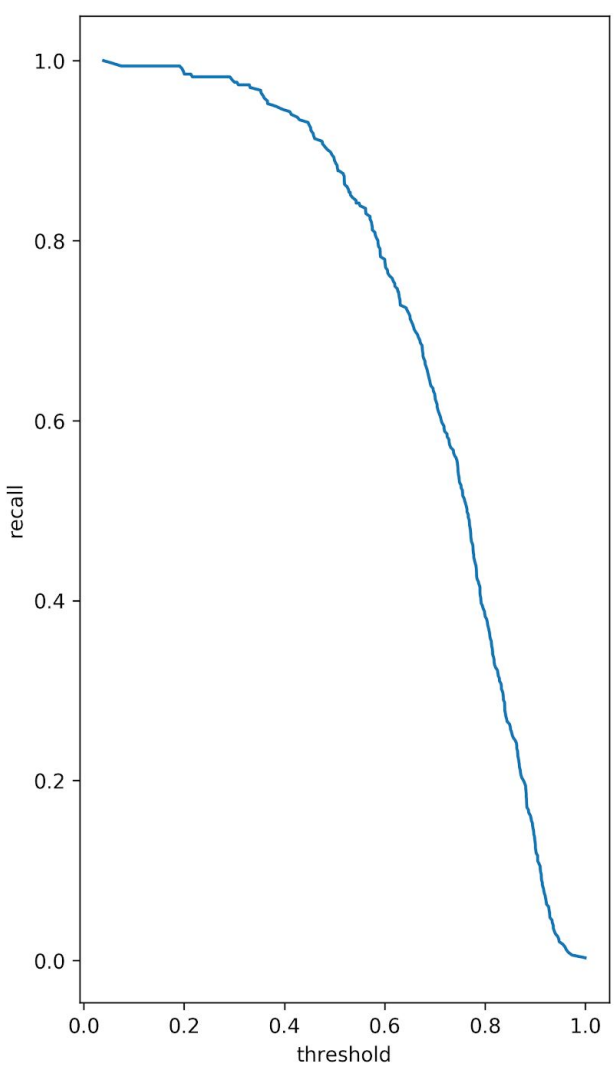
+ Cost-based classification

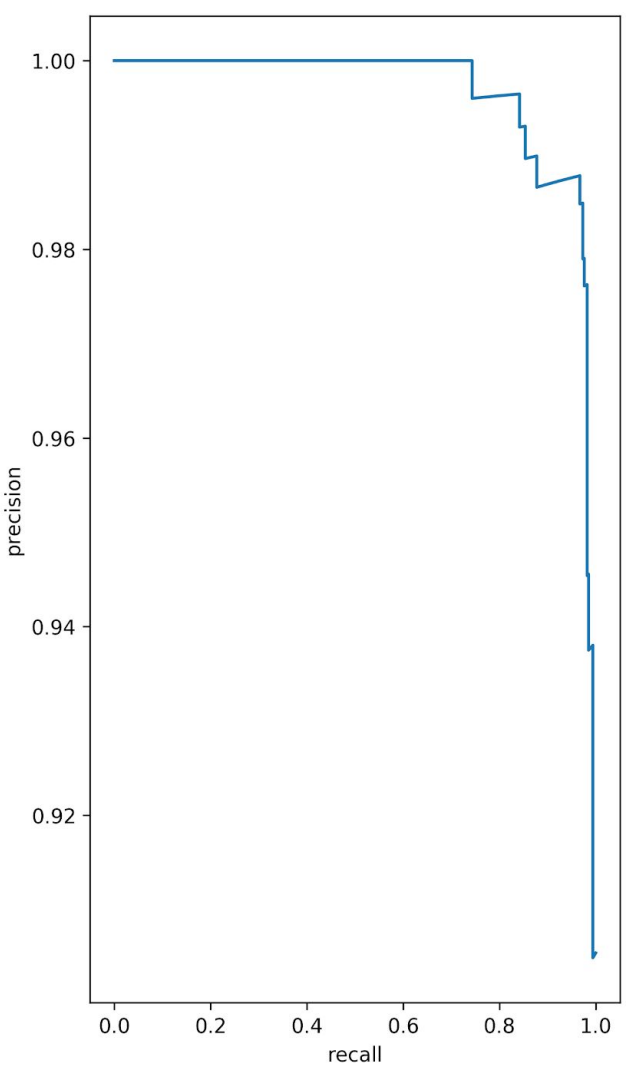
Can we make this a ranking problem?

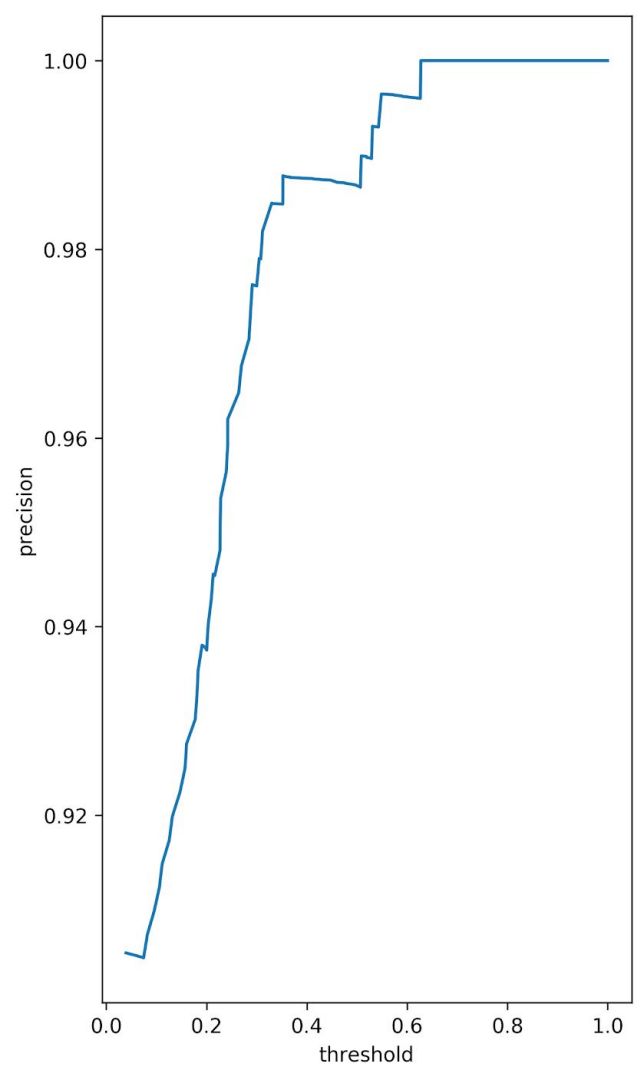
Can we make this a ranking problem?

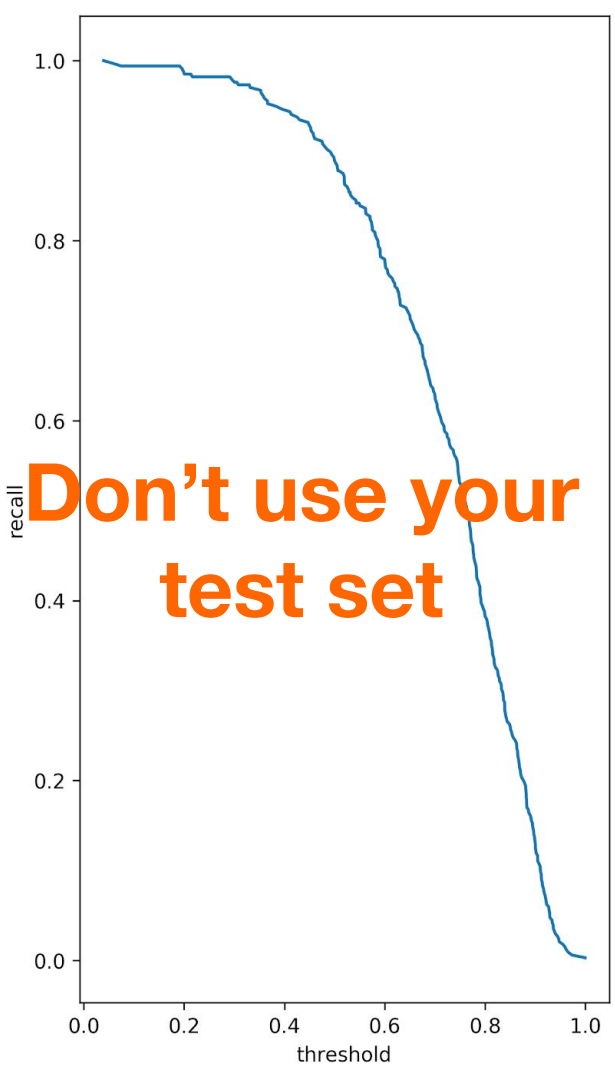
If not: choose a threshold to optimize your metrics


precision
threshold
1.00
0.98
0.96
0.94
0.92
0.0
0.2
0.4
0.6
0.8
1.0

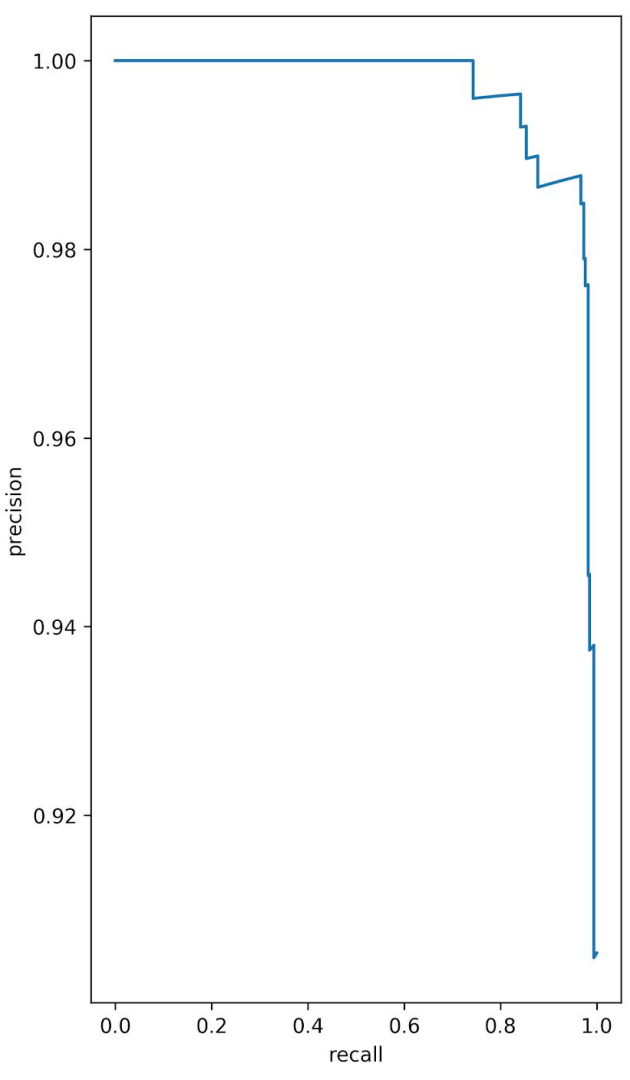

recall
threshold
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


precision
recall
1.00
0.98
0.96
0.94
0.92
0.0
0.2
0.4
0.6
0.8
1.0





Don’t use your test set



POST-PROCESSING

+ Threshold selection

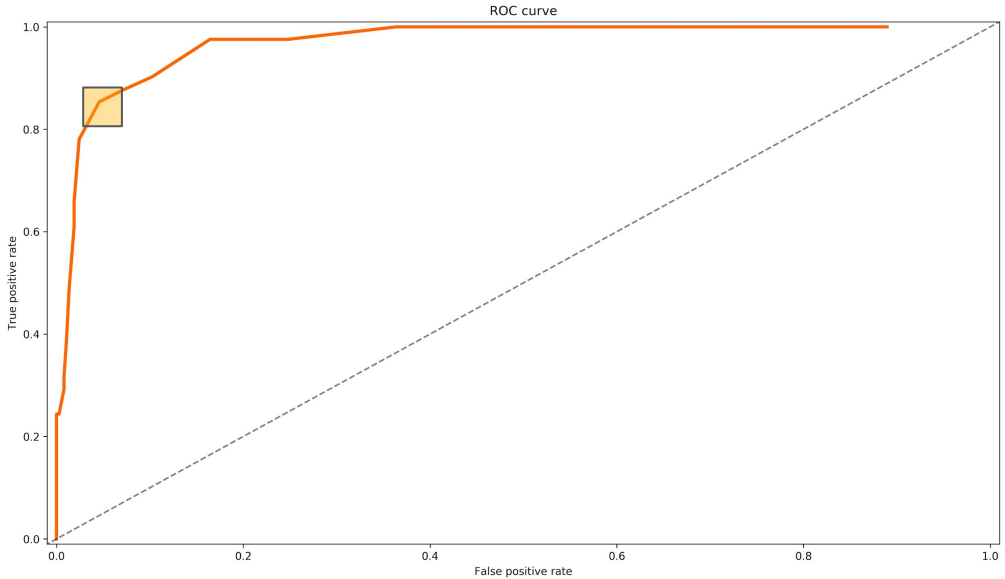
+ Cost-based classification

POST-PROCESSING

+ **Threshold selection**

+ **Cost-based classification**
  - **Use ROC curve to choose a threshold** (Sinha and May)
  - **MetaCost** (Domingos)

Each point on an ROC curve refers to a threshold for which we can calculate cost



True positive rate **0.85**

False positive rate **0.05**

Threshold **0.64**

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = p0 * cost_{FalsePos} * (1 - \text{TNR}) + p1 * cost_{FalseNeg} * (1 - \text{TPR})$$

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = p0 * cost_{FalsePos} * (1 - \text{TNR}) + p1 * cost_{FalseNeg} * (1 - \text{TPR})$$

Prior probability of negative class

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = p0 * cost_{FalsePos} * (1 - \text{TNR}) + p1 * cost_{FalseNeg} * (1 - \text{TPR})$$

Cost of a false positive

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = p0 * cost_{FalsePos} * (1 - \text{TNR}) + p1 * cost_{FalseNeg} * (1 - \text{TPR})$$

True negative rate (specificity)

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = p0 * cost_{FalsePos} * (1 - \text{TNR}) + p1 * cost_{FalseNeg} * (1 - \text{TPR})$$

Prior probability of positive class

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = p0 * cost_{FalsePos} * (1 - \text{TNR}) + p1 * cost_{FalseNeg} * (1 - \text{TPR})$$

Cost of a false negative

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = p0 * cost_{FalsePos} * (1 - \text{TNR}) + p1 * cost_{FalseNeg} * (1 - \text{TPR})$$

True positive rate (sensitivity)

Each point on an ROC curve refers to a threshold for which we can calculate cost

$$\text{Cost} = 0.1 * 5 * 0.05 + 0.9 * 1 * 0.15 = 0.16$$

A false positive is 5 times as bad as a false negative

Minority class = 10%

Pick the threshold with the lowest cost

Threshold	Cost
0.64	0.16
0.89	0.24
0.91	0.86

Pick the threshold with the lowest cost

Threshold	Cost
0.64	0.16
0.89	0.24
0.91	0.86

Cost-based classification is different from special-purpose learners

- Does not modify the learning algorithm
- Can be used with (almost) any model

PREDICTION POST-PROCESSING

Straightforward

PREDICTION POST-PROCESSING

Straightforward

Usable with most models

PREDICTION POST-PROCESSING

Straightforward

Usable with most models

Understudied in imbalanced domains

Agenda

What is class imbalance?

Recognition

Solutions

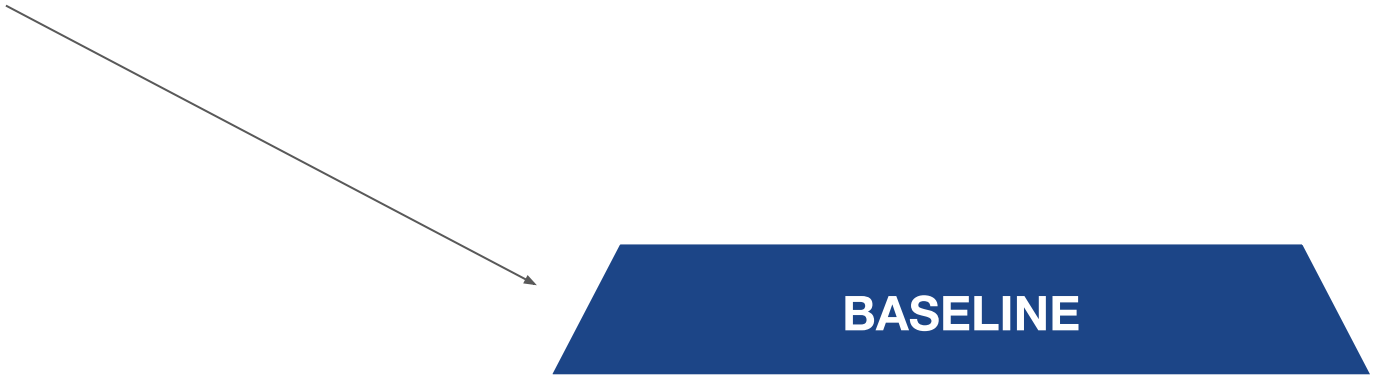
Recommendations

PRACTICAL TIPS FOR DEALING WITH IMBALANCE

PRACTICAL TIPS FOR DEALING WITH IMBALANCE

Establish a baseline

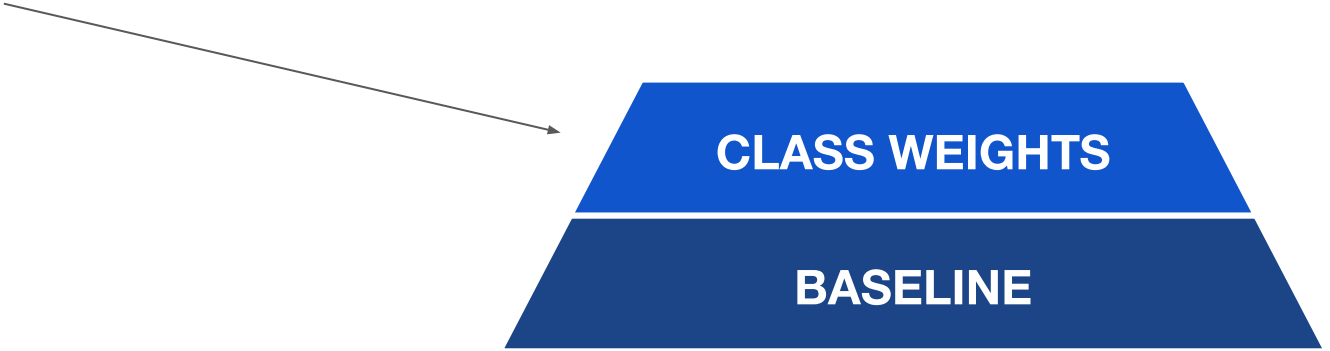
Use AUC



PRACTICAL TIPS FOR DEALING WITH IMBALANCE

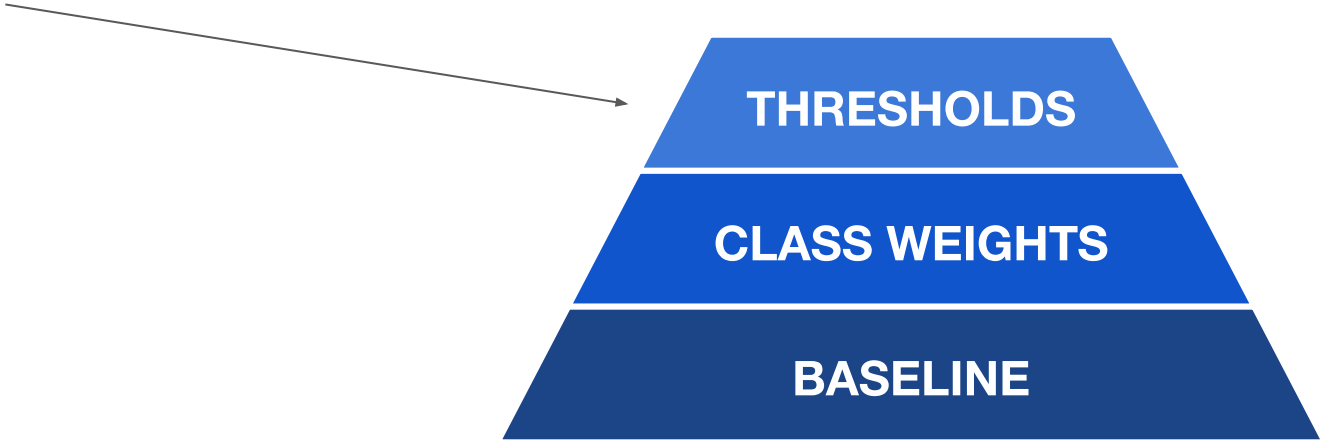
Provide class weights

(if possible)



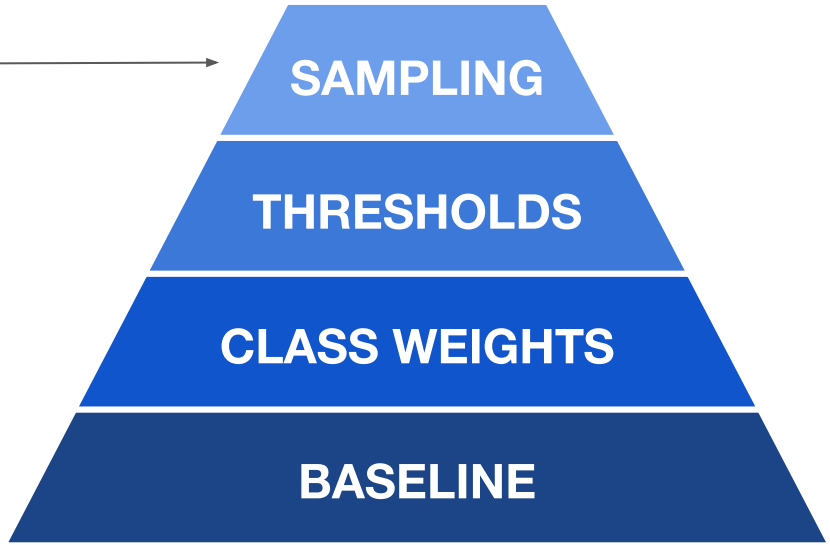
PRACTICAL TIPS FOR DEALING WITH IMBALANCE

Select thresholds wisely



PRACTICAL TIPS FOR DEALING WITH IMBALANCE

Use random sampling



“

In almost all imbalanced scenarios, practitioners should bag classifiers induced over balanced bootstrap samples
– Wallace et al.

“

In almost all imbalanced scenarios, practitioners should bag classifiers induced over balanced bootstrap samples

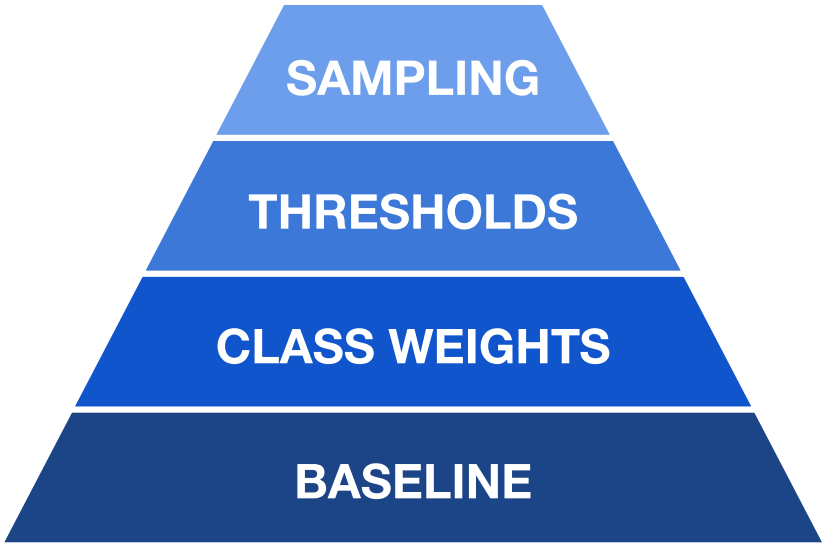
– Wallace et al.

“

Random over-sampling… is very competitive to more complex over-sampling methods

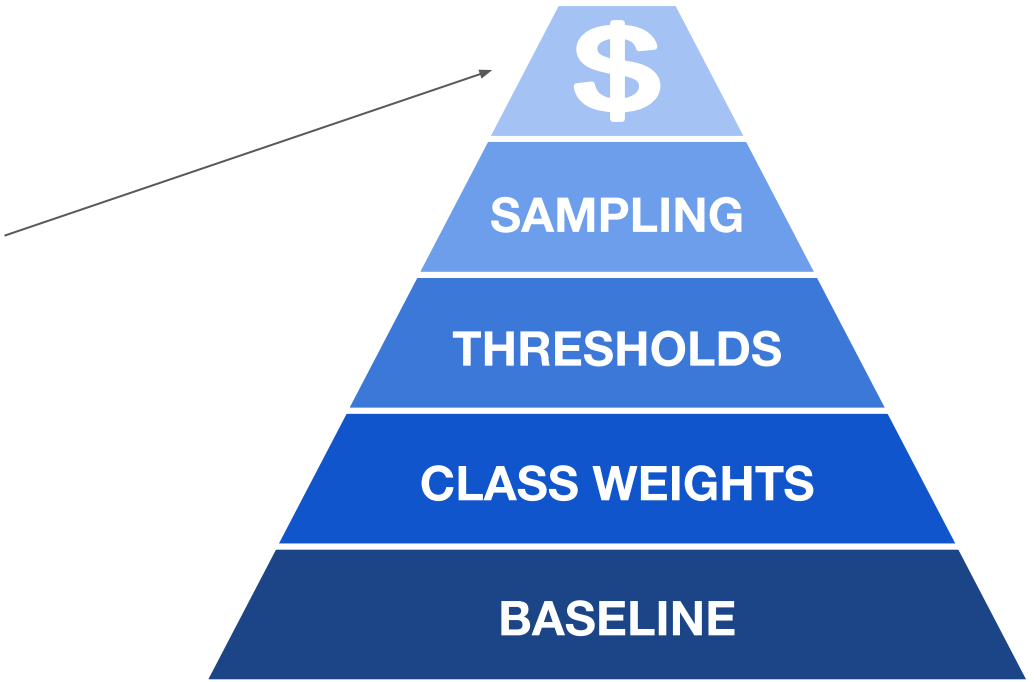
– Batista et al.

PRACTICAL TIPS FOR DEALING WITH IMBALANCE



PRACTICAL TIPS FOR DEALING WITH IMBALANCE

Explore more expensive techniques (e.g. SMOTE)



slides: **go.indeed.com/ODSC**
twitter: **@SamuelDataT**
email: **sgt@samueltaylor.org**

Appendix

REFERENCES

+ Batista, Prati, and Monard. A study of the behavior of several methods for balancing machine learning training data.
+ Branco, Torgo, and Ribeiro. A Survey of Predictive Modelling under Imbalanced Distributions.
+ Chawla, Bowyer, Hall, and Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique.
+ Domingos. MetaCost: a general method for making classifiers cost-sensitive.
+ He, Bai, Garcia, and Li. ADASYN: Adaptive Synthetic Sampling approach for Imbalanced Learning.

REFERENCES

+ KEEL
+ Luque, Carrasco, Martin, and de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix.
+ Sinha and May. Evaluating and Tuning Predictive Data Mining Models Using Receiver Operating Characteristic Curves.
+ Wallace, Small, Brodley, and Trikalinos. Class Imbalance, Redux.

HOW ABOUT F1 (AKA F MEASURE)?

+ **“F1 [is] highly biased and should be avoided for use in imbalanced datasets” - *Luque et al.***

+ **“F combines two values that should never be combined” - Soboroff**

Performance loss… is quite modest (below 5%) for the most balanced distributions up to 10% of minority examples

– Prati, Batista, and Silva