Train In Data

Boosting +

Data Pre-Processing

Boosting

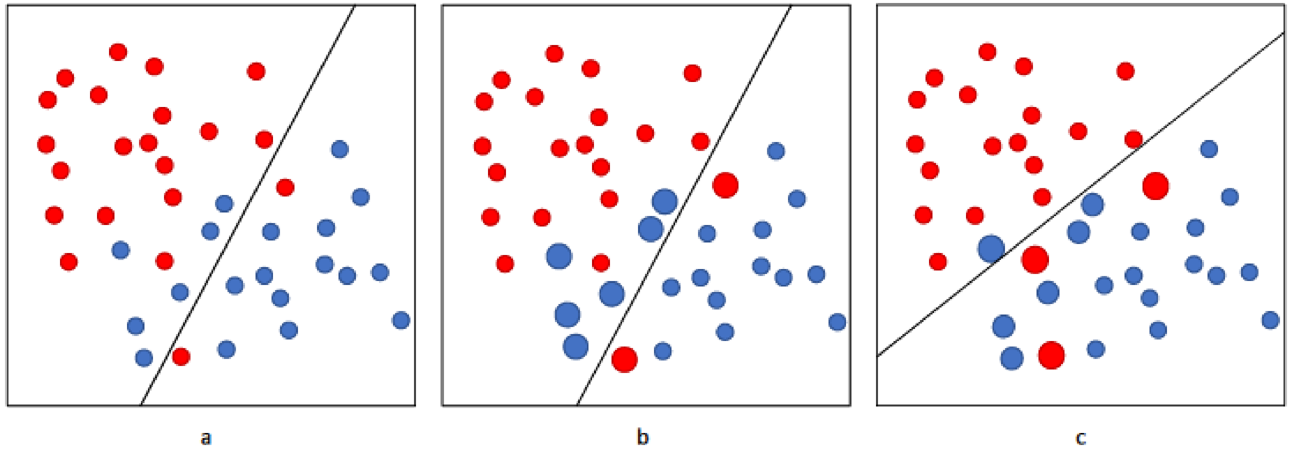


Figure 6: The boosting approach. A classifier is trained on the original data (a). The weights of misclassified instances (dot size in the figure) are increased (b). A new classifier is trained on the new data set and weights are updated accordingly (c).

Image taken from Data Mining: Accuracy and Error Measures for Classification and Prediction. Galdi and Tagliaferri, 2018

- Classifiers are built sequentially
- Classifiers are trained on all data
- Observations are given **weights** which reflect how difficult they are to classify.
- Each classifier's prediction has a different **weight** towards final decision, based on their accuracy

Boosting + Resampling

RESAMPLE DATA AT EACH ITERATION

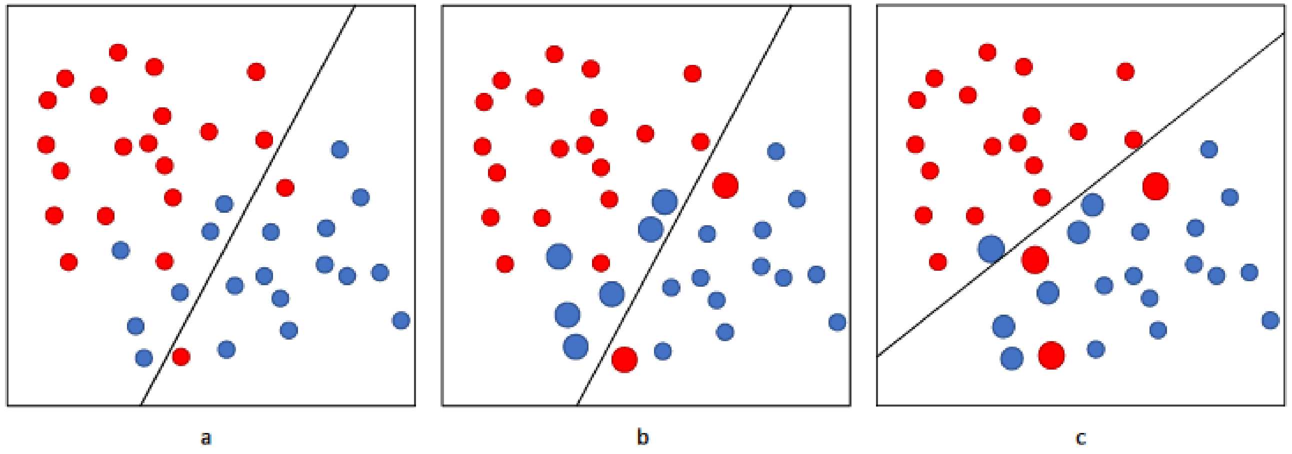
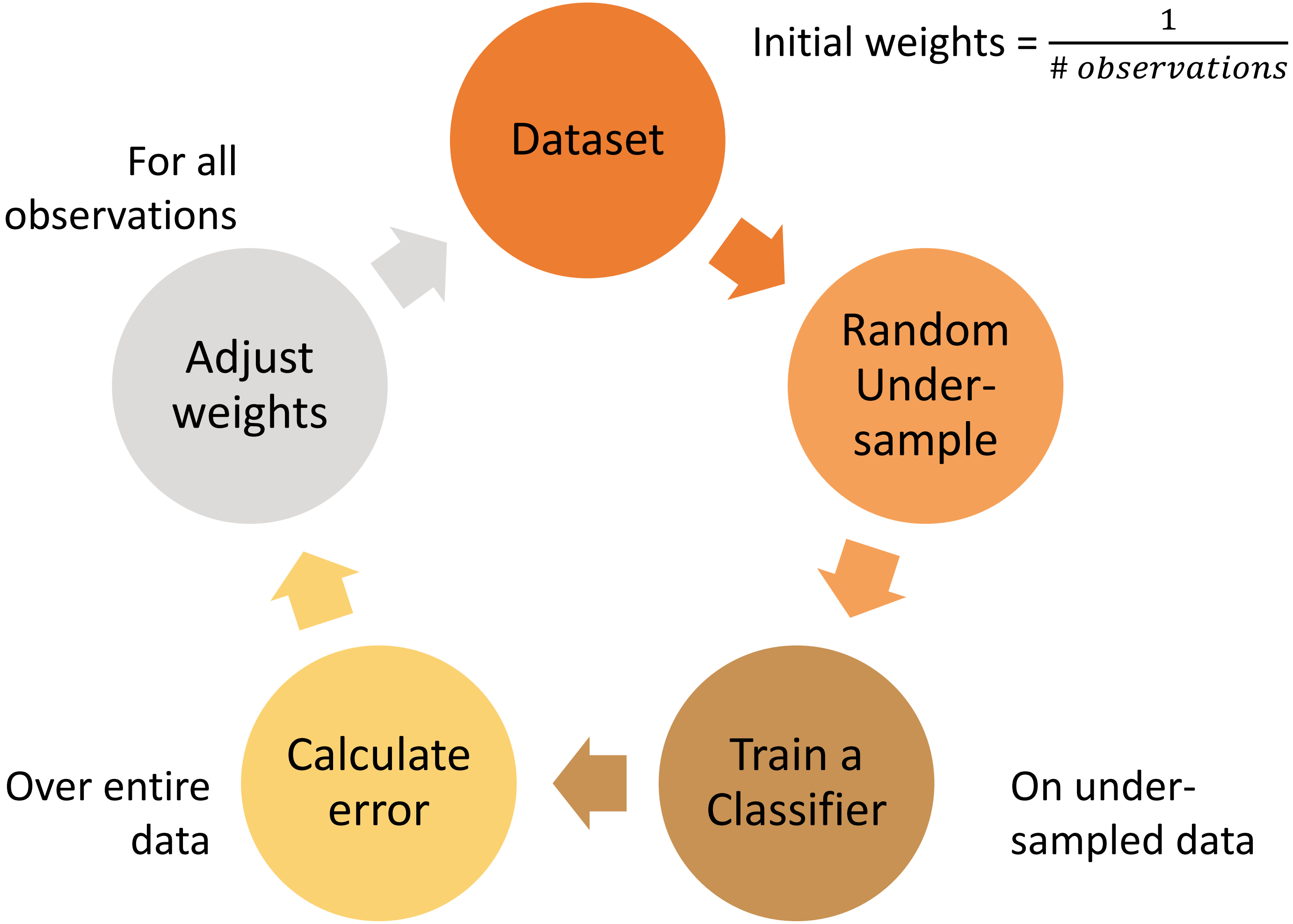


Figure 6: The boosting approach. A classifier is trained on the original data (a). The weights of misclassified instances (dot size in the figure) are increased (b). A new classifier is trained on the new data set and weights are updated accordingly (c).

Image taken from Data Mining: Accuracy and Error Measures for Classification and Prediction. Galdi and Tagliaferri, 2018

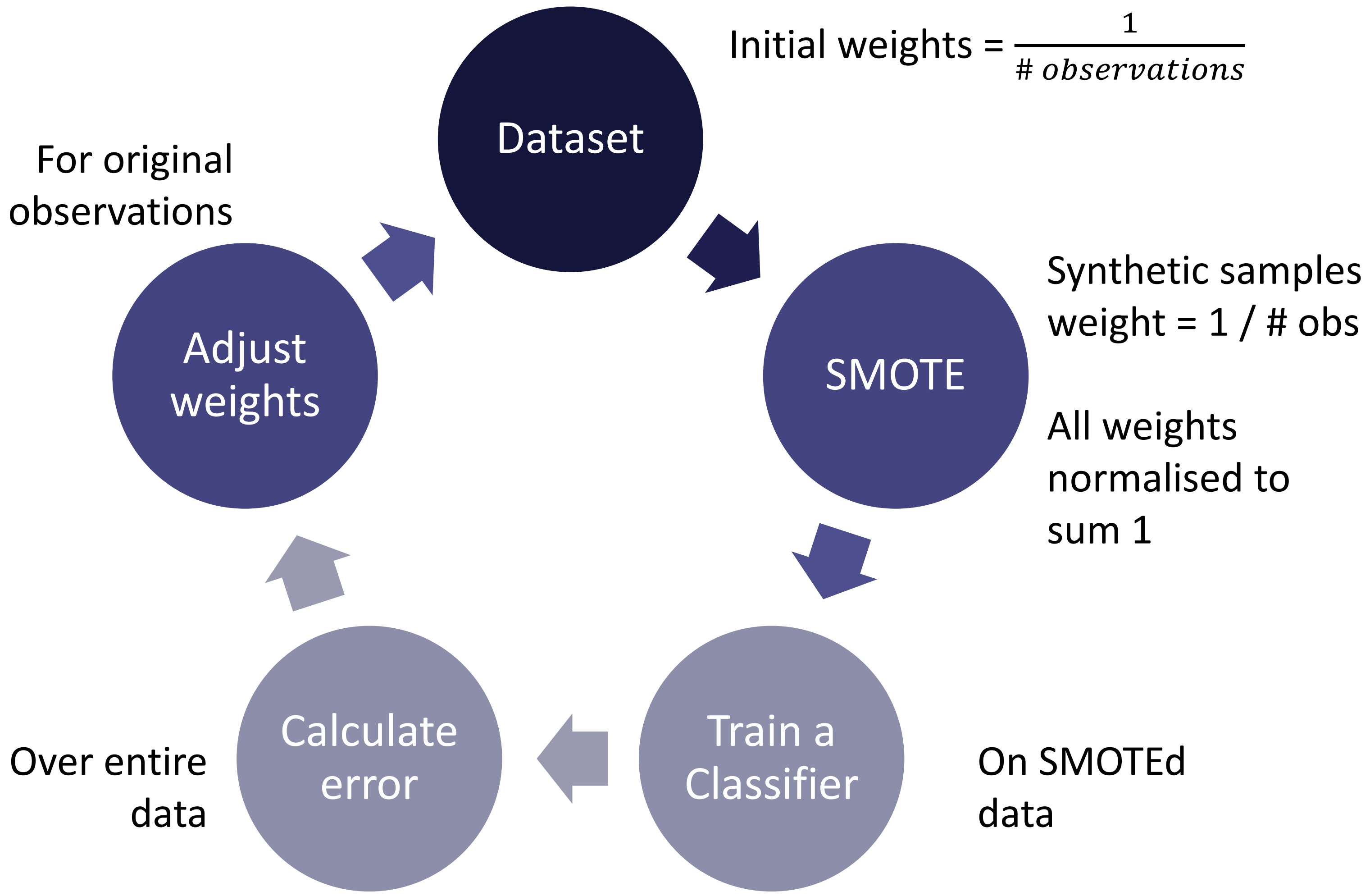
- Classifiers are built sequentially
- Classifiers are trained on resampled data
- Observations are given **weights** which reflect how difficult they are to classify.
- Each classifier's prediction has a different **weight** towards final decision, based on their accuracy

RUSBoost



- Random under-sampling + Boosting
- Smaller datasets ➔ faster training
- Lost information is compensated in subsequent iterations
- Authors claims performs better than under-sampling or boosting alone, and also at least as well as SMOTEBoost, but it is much faster to train.

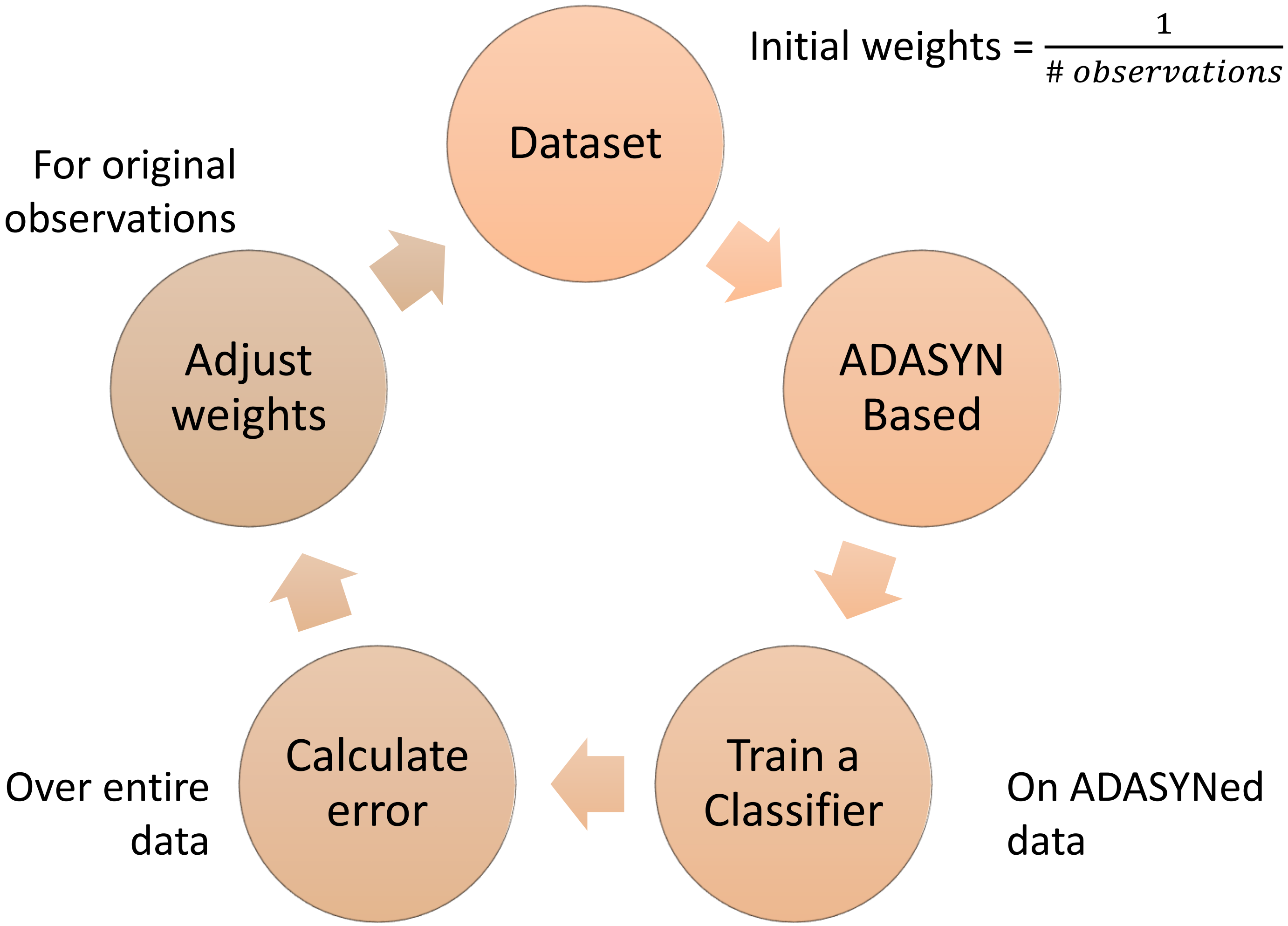
SMOTEBoost



- SMOTE + Boosting
- Create more instances of the minority class
  - adds diversity
  - Improves classifier accuracy
- Requires quite a bit of computational time to perform SMOTE at each iteration

RAMOBoost

Initial weights = $\frac{1}{\# \, observations}$



- ADASYN (adaptation) + Boosting
- Resample more Xmin with more neighbours from Xmaj
- Create more synthetic examples from more Xmin with more neighbours from Xmaj
- Computationally expensive

RUSBoost with Imbalanced Learn

Random Under-Sampling + Boosting

`imblearn.ensemble`: Ensemble methods

The `imblearn.ensemble` module include methods generating under-sampled subsets combined inside an ensemble.

`ensemble.RUSBoostClassifier`([...])	Random under-sampling integrated in the learning of AdaBoost.

THANK YOU

www.trainindata.com