Train In Data

Bagging +

Data Pre-Processing

Bagging

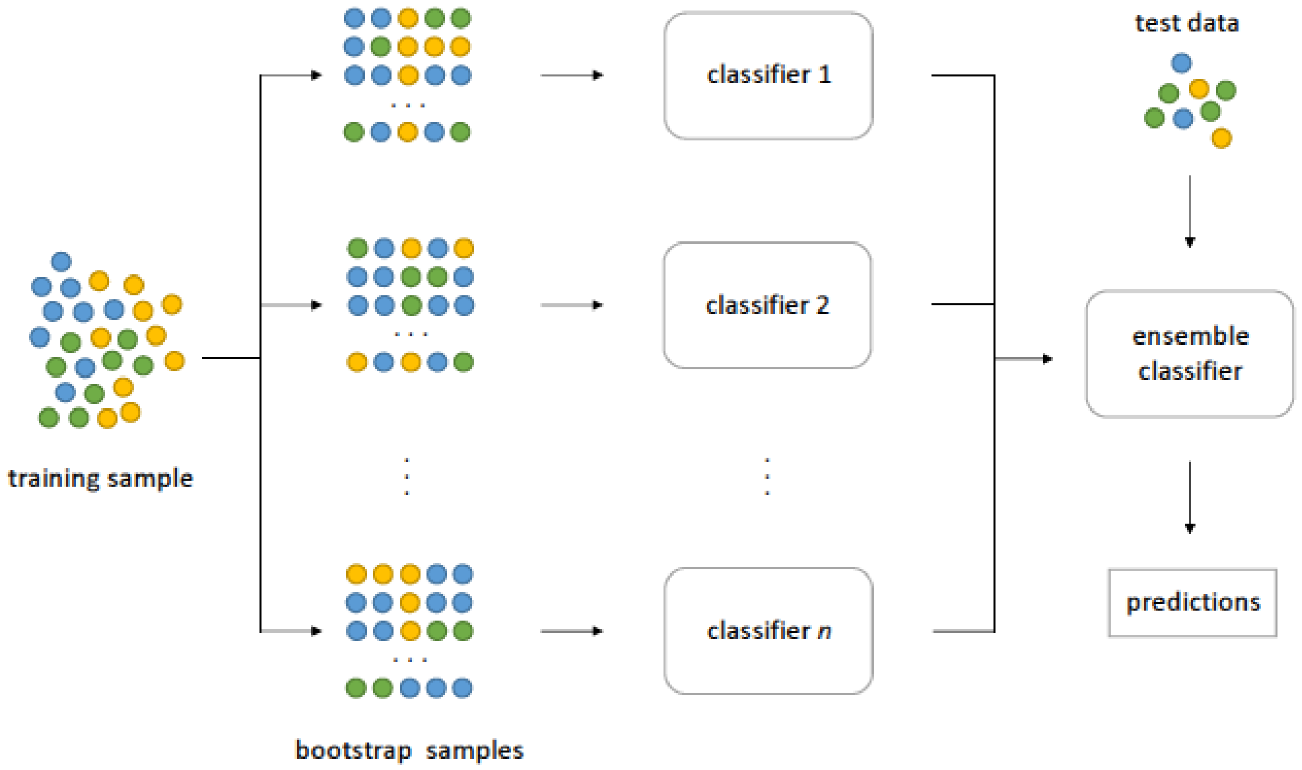


Figure 5: The bagging approach. Several classifier are trained on bootstrap samples of the training data. Predictions on test data are obtained combining the predictions of the trained classifiers with a majority voting scheme.

- Off-the-shelf algorithms optimize the accuracy
  - Not suitable for imbalanced datasets

Bagging + Re-Sampling

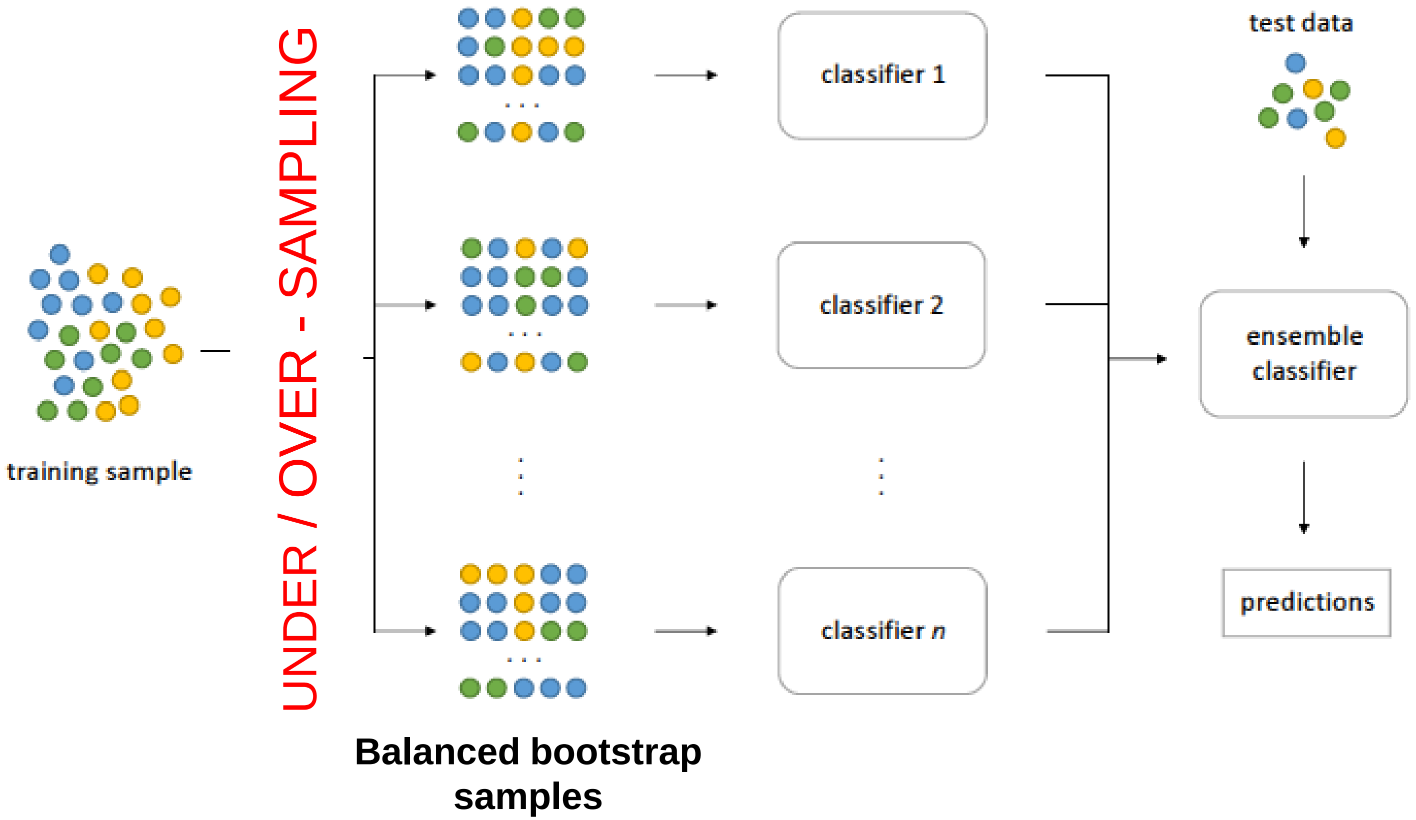
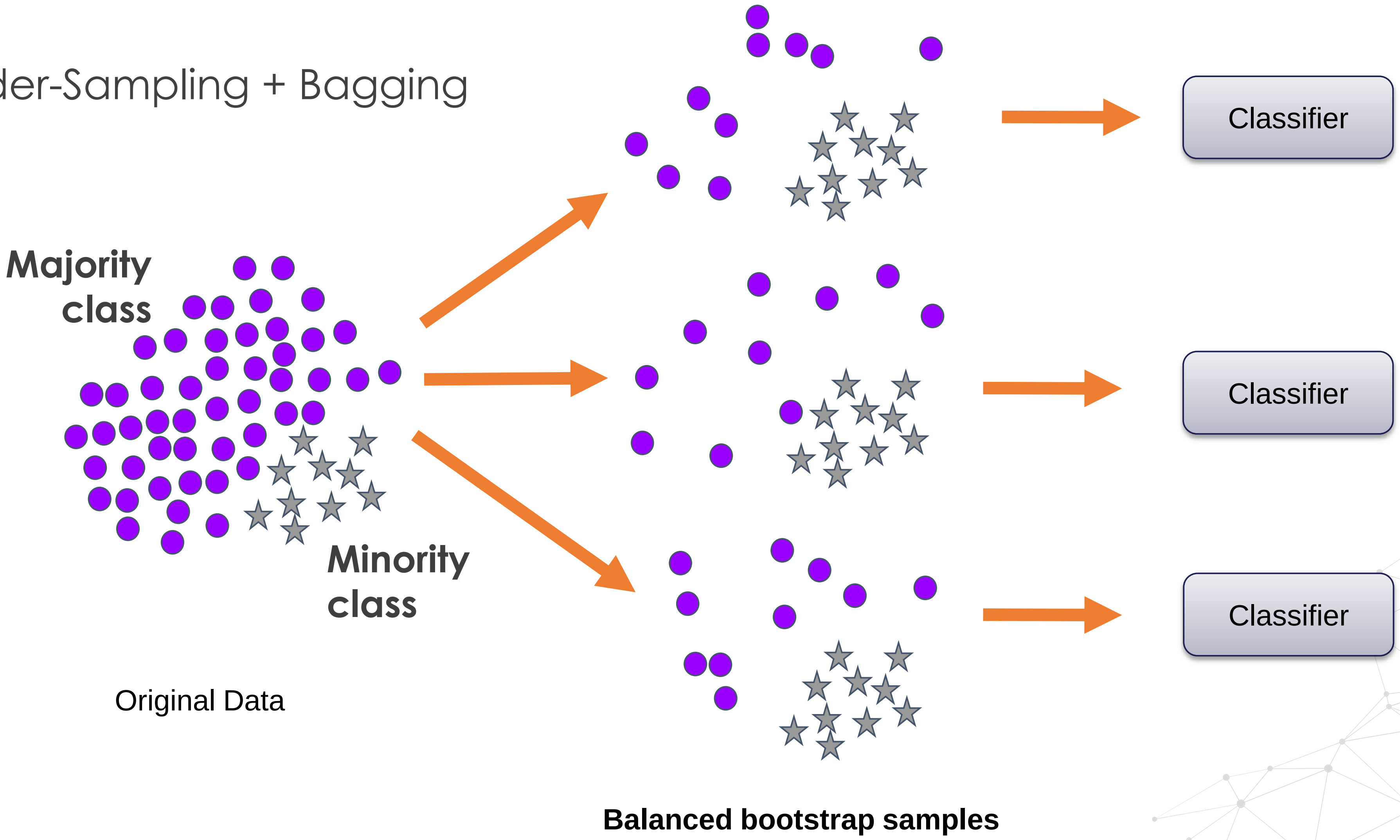


Figure 5: The bagging approach. Several classifier are trained on bootstrap samples of the training data. Predictions on test data are obtained combining the predictions of the trained classifiers with a majority voting scheme.

- Off-the-shelf algorithms optimize the accuracy
  - Not suitable for imbalanced datasets
- Use under- or over-sampling to create balanced datasets
  - During bootstrap

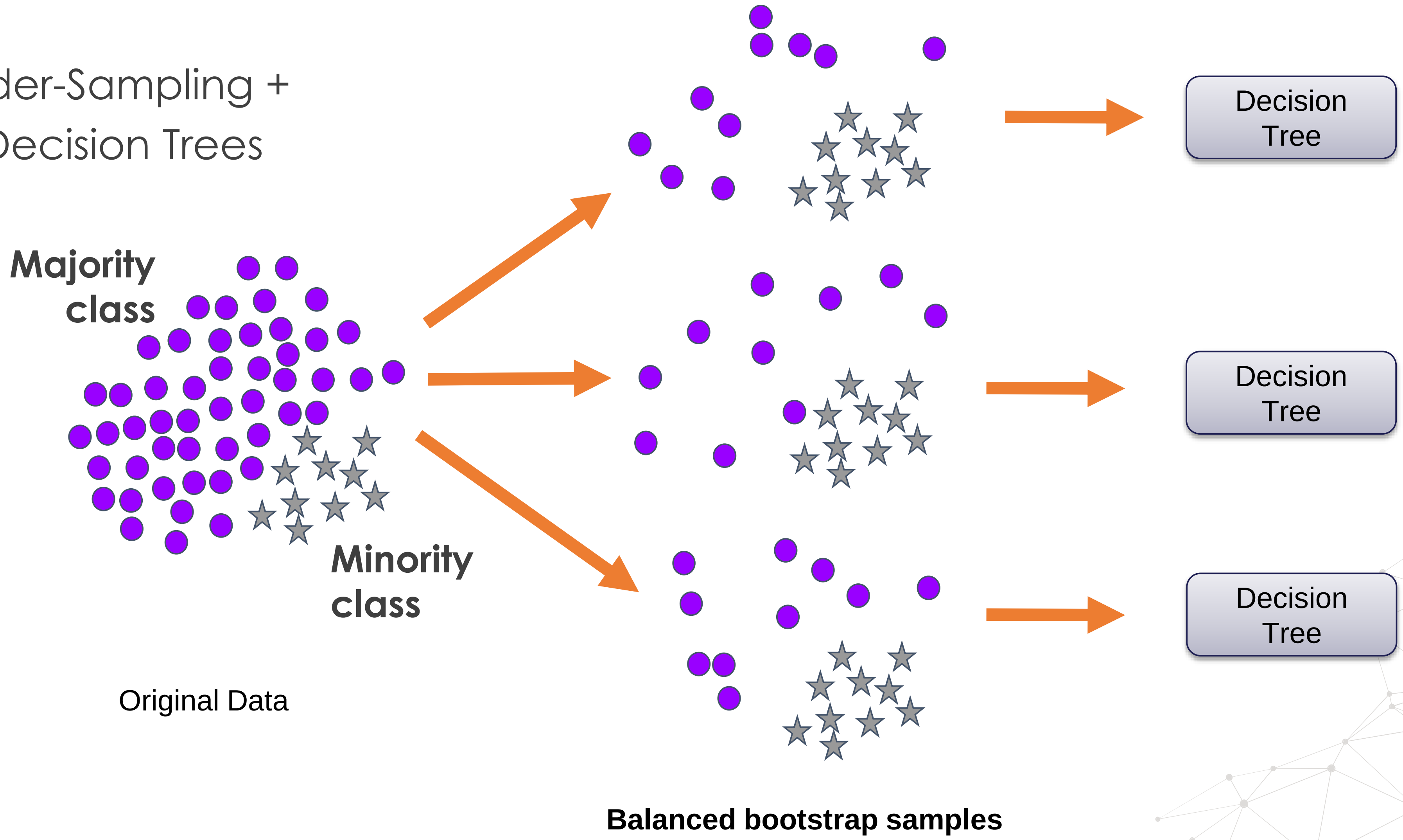
UnderBagging

Random Under-Sampling + Bagging



Balanced Random Forests

Random Under-Sampling +
Bagging of Decision Trees



UnderBagging with Imbalanced Learn

Random Under-Sampling + Bagging

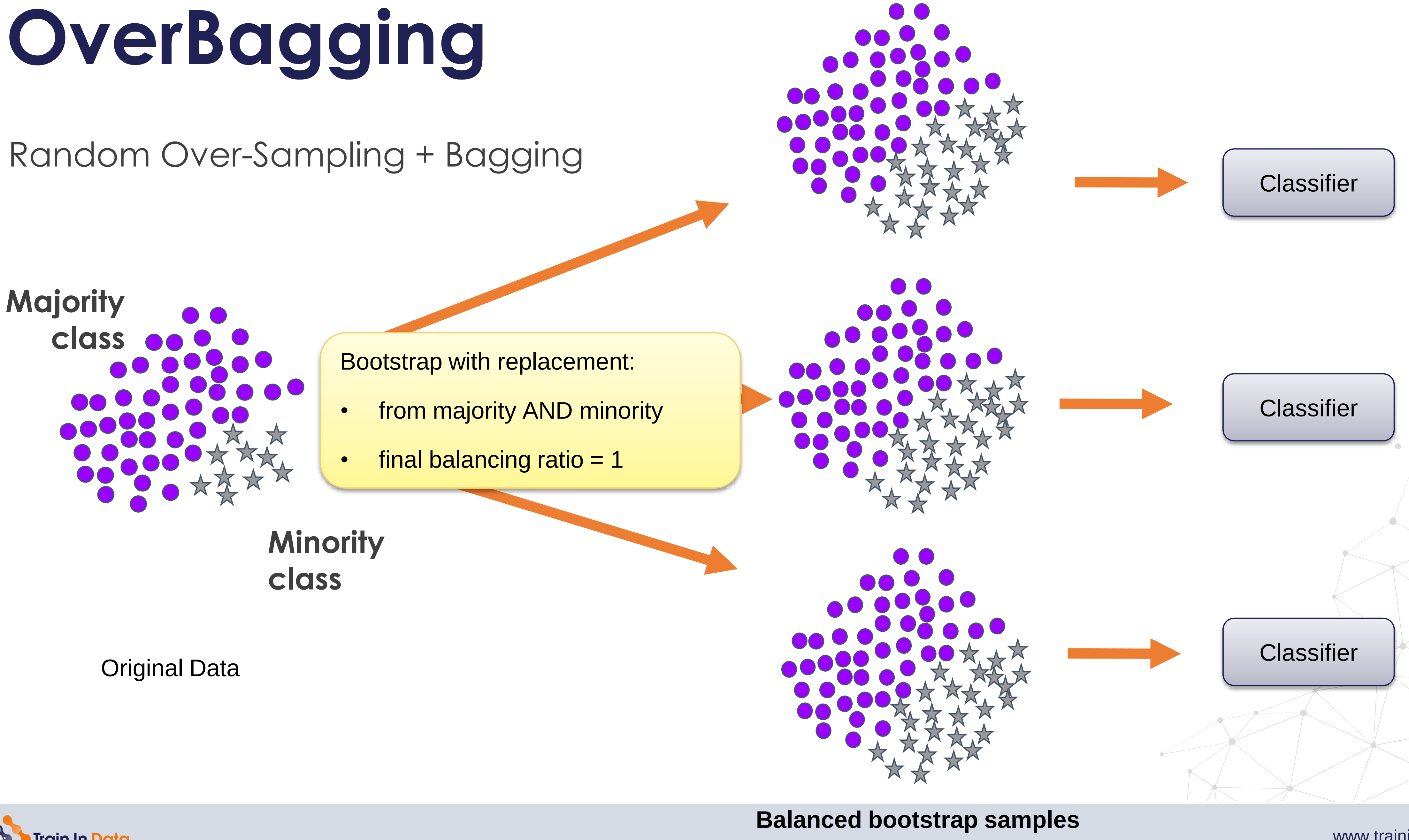
`imblearn.ensemble`: Ensemble methods

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

`ensemble.BalancedBaggingClassifier` ([...])	A Bagging classifier with additional balancing.
`ensemble.BalancedRandomForestClassifier` ([...])	A balanced random forest classifier.

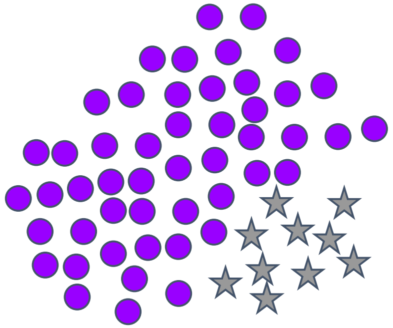
OverBagging

Random Over-Sampling + Bagging



SMOTEBagging

SMOTE + Bagging

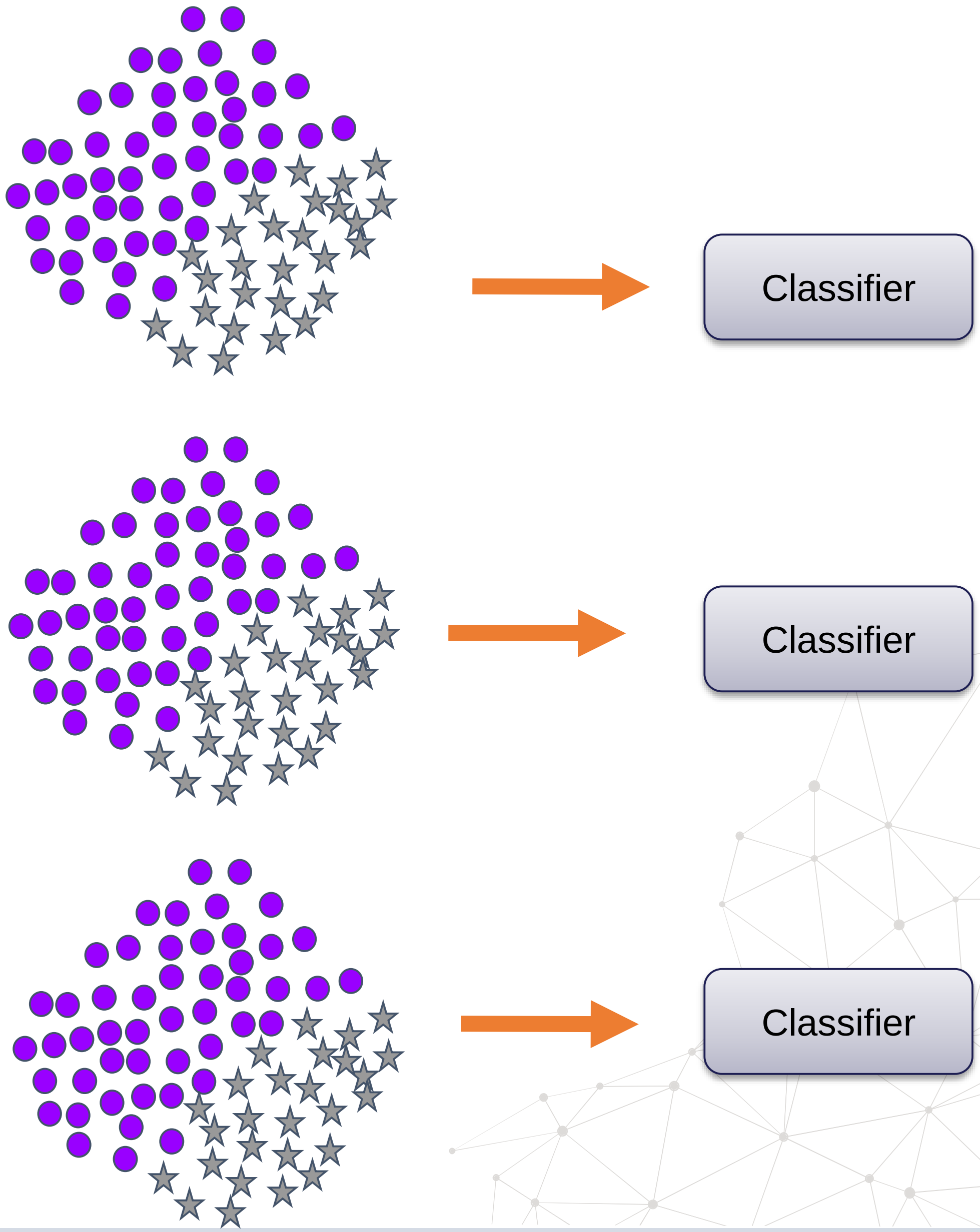


Original Data

- Majority class bootstrapped with replacement

Minority class:

- A % is bootstrapped with replacement from 10-100% on each iteration
- The rest are created by SMOTE till desired balancing ratio is reached



Balanced samples

Ensemble approaches