# Ensemble methods for Imbalanced data

1. Solutions for imbalanced datasets

* Data level – undersampling, oversampling
* Cost-sensitive – higher miss-classification costs
* Ensemble algorithm – bagging and boosting

1. Ensembles for imbalanced dataset

* Boosting and bagging
* Data level
* Cost-sensitive

1. Ensemble approaches

A diagram of a problem

AI-generated content may be incorrect.

# Foundations of Ensemble Learning

See video

Bagging & Boosting are ways to create classifiers that make different mistakes -> combined and generalize the learner

Bagging – random forests

Boosting – Gradient boosting

# Bagging + Over / under sampling

Use under or oversampling during bootstrap to create balanced datasets

# Boosting + Resampling

1. Boosting + Resampling

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

1. RUSBoost

* Random undersampling + boosting
* Procedure
* Dataset - Initial weights = 1 / #obs
* Random undersampling
* Train a classifier
* Calculate error (over entire data)
* Adjust weights for all observations
* Repeat
* Lost information is compensated in subsequent iterations
* Author’s claim: performs better than undersampling or boosting alone, and also at least as well as SMOTEBoost, but much faster to train

1. SMOTEBoost

* SMOTE + Boosting
* Procedure
* Dataset
* SMOTE: Synthetic samples weight = 1 / # obs. All weights normalized to sum to 1
* Train a classifier (on SMOTEd data)
* Calculate error
* Adjust weights
* Repeat
* Create more instances of the minority class
* Adds diversity
* Improves accuracy
* Requires a bit of computational time to perform SMOTE at each iteration

1. RAMOBoost

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

# Hybrid Methods

1. Bagging + Boosting + Resampling

A diagram of a diagram of a sample

AI-generated content may be incorrect.

* AdaBoost is trained on each bag
* Bags are created with random undersampling

1. Balanced Random Forests

A diagram of a boottrap sample

AI-generated content may be incorrect.

1. EasyEnsemble

* Random Undersampling + Bagging of AdaBoost

1. BalanceCascade

* EasyEnsemble but instead of parallel building, we build sequentially

A diagram of a diagram

AI-generated content may be incorrect.

# Wrapping up

1. Scalability

* Methods based on KNN do not scale well
* In ensemble we repeat the resampling in each bag or at each round of boosting
* Ensemble with random over/undersampling are the least costly

1. Comparison (see videos)

* Most ensemble algorithms have similar performance, but speed can differ 🡪 choose one that is least costly to scale

1. Optimize the metric of interest

* Methods with best recall will not have F1 score