LING 506 - TOPICS IN COMPUTATIONAL LINGUISTICS

Introductory Machine Learning

Yan Tang

Department of Linguistics, UIUC

Week 9-10

Last week…

- The holdout method
  - Training, validation and testing
- Cross-validation
  - K-fold cross-validation
  - Stratified k-fold cross-validation
  - Leave-one-out cross-validation
  - Shuffle-split cross-validation

Last week…

- Grid search
  - A brute-force exhaustive search approach
  - Grid search with cross-validation
- Randomised search
  - Only evaluates a specific number of random hyperparameter combinations
  - May achieve similar performance as grid search
  - Allows for efficient computation when search space is large

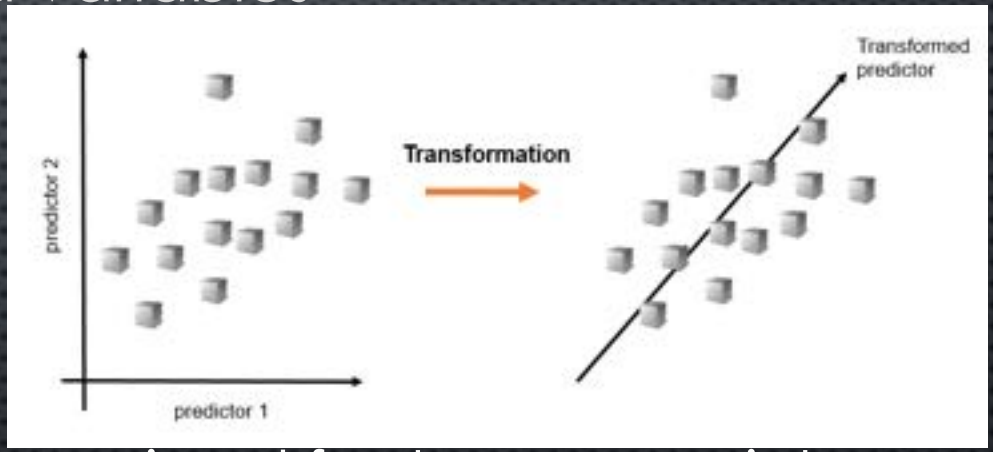
Feature reduction

- ML problems may involve high-dimensional data with a tremendous number of variables
  - Computationally expensive – computing resources and time

- Motivation of reducing the number of variables
  - Simplify the model - better generalised, easier to interpret
  - Expedite model training and evaluation
  - Boost prediction speed

Feature reduction techniques: Feature transformation

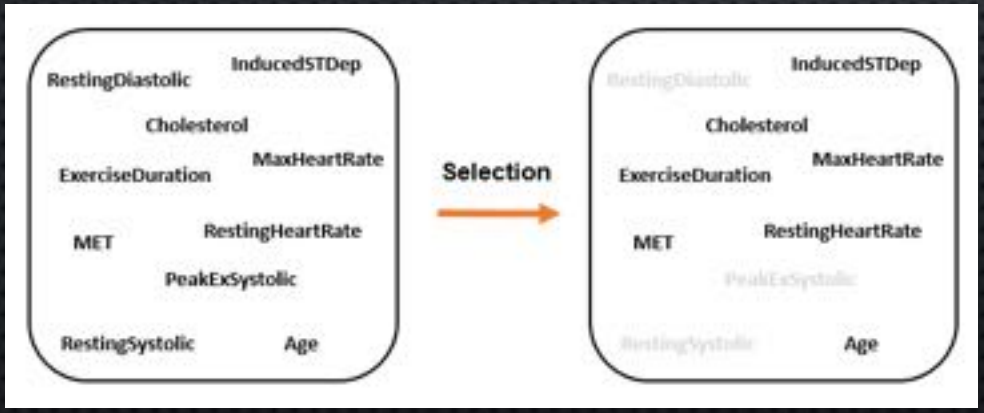
- Feature transformation transforms the coordinate space of the observed variables
  - e.g. PCA

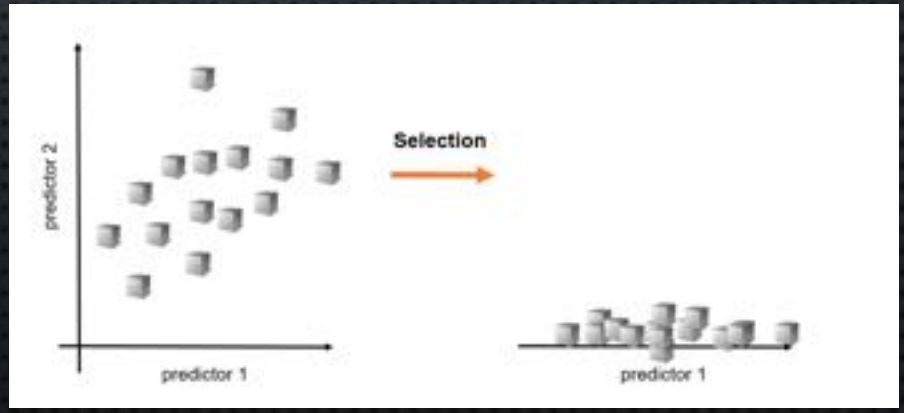


- PCA transforms an n-dimensional feature space into a new n-dimensional space of orthogonal components
  - The components are ordered by the variation explained
  - Discarding the components beyond a chosen threshold of explained variance

Feature reduction techniques: Feature selection

- Data often contains variables that are irrelevant with the response
- Data may also contain highly correlated variables; no need to include all of them in the model
- Choose a subset of the observed variables





Feature reduction techniques: Feature selection

- Some classifiers provide their own methods or attributes of feature selection
  - e.g. The implementation of Decision Trees in *sklearn* offers "*feature_importances_*"

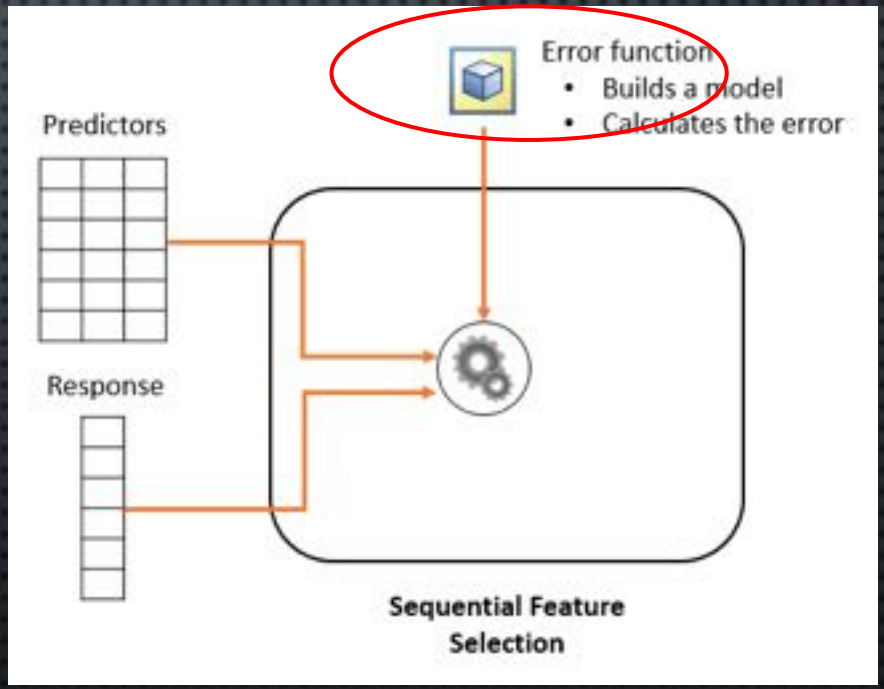
- Weighted impurity (as Gini importance):

$$I_i = \frac{N_i}{N}\left(G_i - \frac{N_i^L G_i^L}{N_i} - \frac{N_i^R G_i^R}{N_i}\right)$$

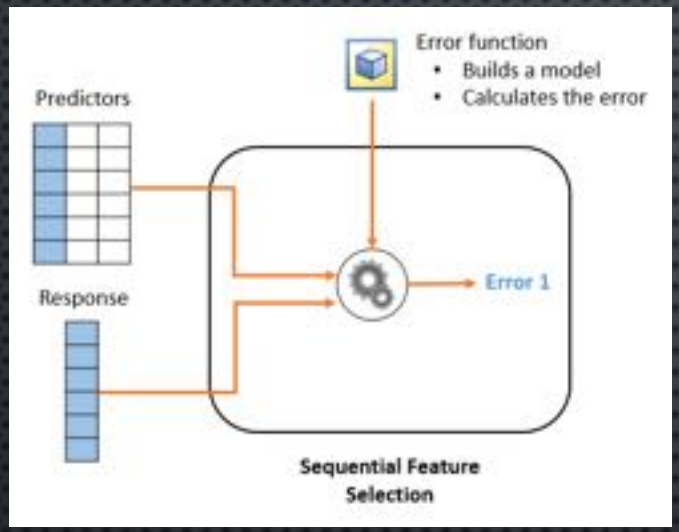
N_i: Number of samples at node i
G_i: Gini impurity of node i
N: Number of total samples
N_i^L, N_i^R : Number of samples in the left/right branch
G_i^L, G_i^R : Gini impurity of the left/right branch

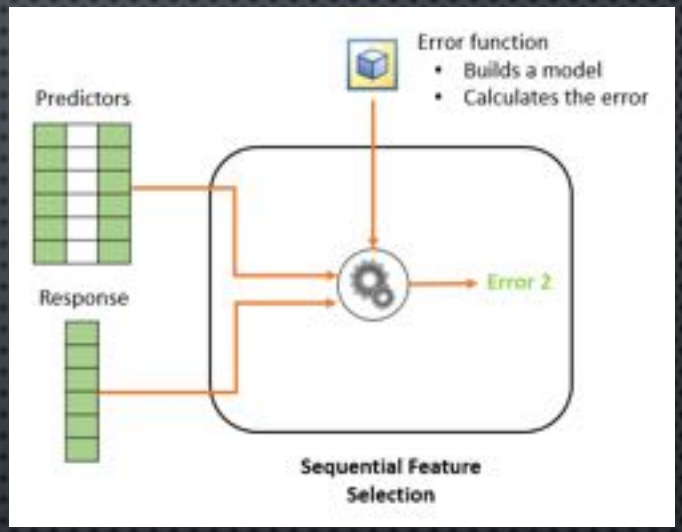
Feature reduction techniques: Feature selection

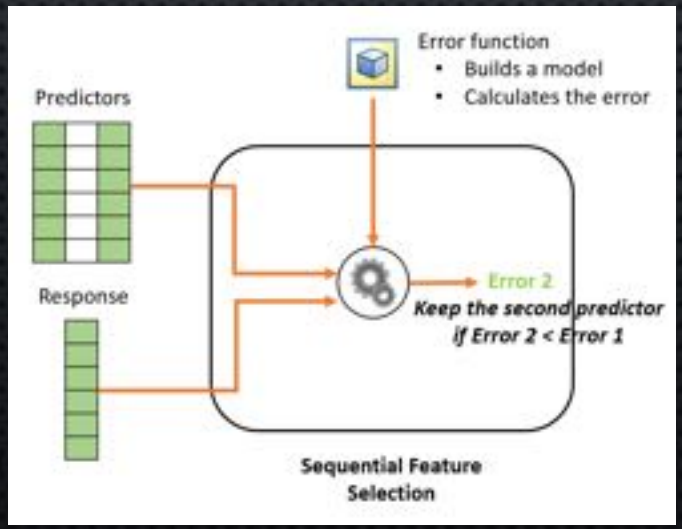
- Sequential feature selection
  - Can be used with any learning algorithm
- Idea: incrementally add variables to the model if prediction error keeps reducing
  - Requires an error function that builds a model and calculates the prediction error



Feature reduction techniques: Feature selection





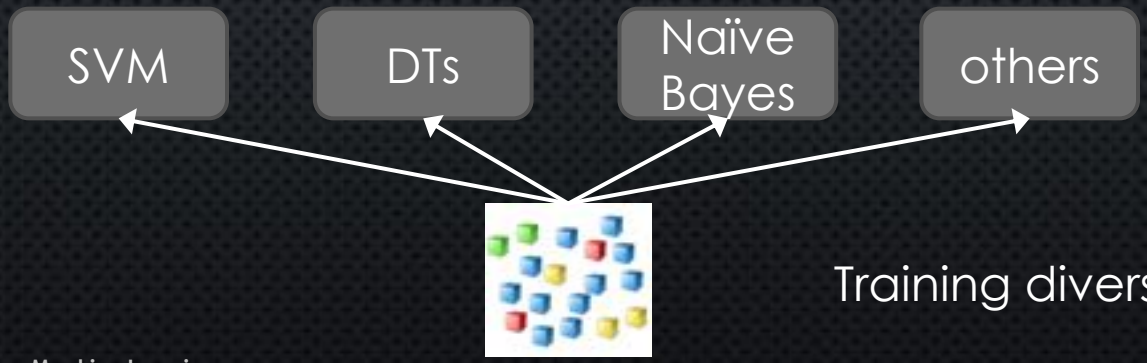


Feature reduction techniques: Other selection techniques

- Recursive feature elimination
  - In the opposite to sequential feature selection
  - Starting with all features, discards the least important feature, until only a desired number of features left
- Univariate statistics (e.g. ANOVA)
  - Compute whether there is a statistically significant relationship between each variable and target
  - Only consider each variable individually
  - Choose features with highest confidence
  - However, independent of the ML model

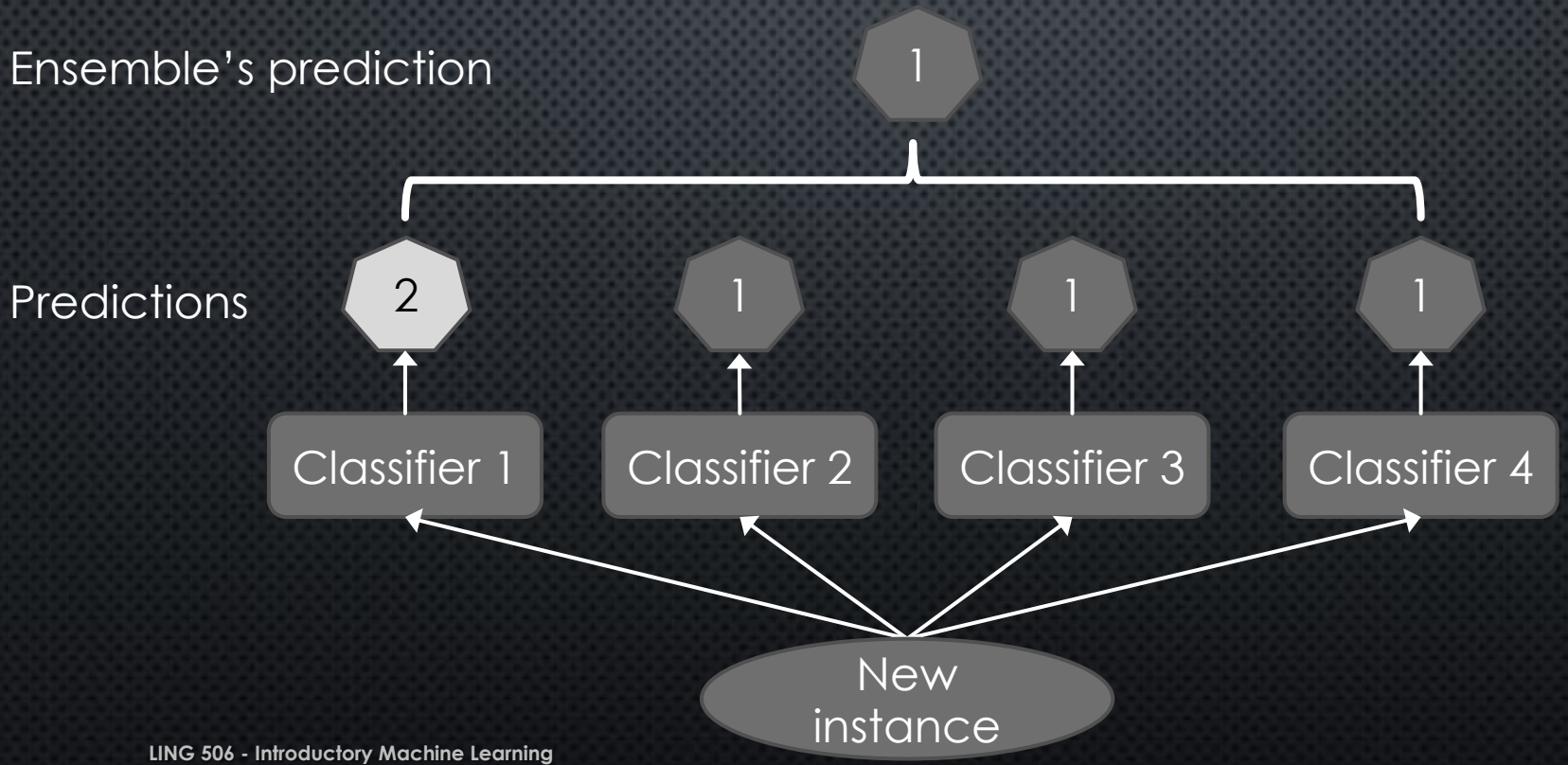
Ensemble learning

- *Wisdom of the crowd*
  - Individuals: weak learner; ensemble: strong learner
  - Aggregate the predictions of a group of classifiers/regressors to achieve better accuracy than any best individuals
  - A group of predictors (e.g. KNN, Naïve Bayes, SVM and DTs)



Voting classifier

- Aggregate the predictions of each predictor; predict the class gets the most votes
  - A **hard-voting** classifier

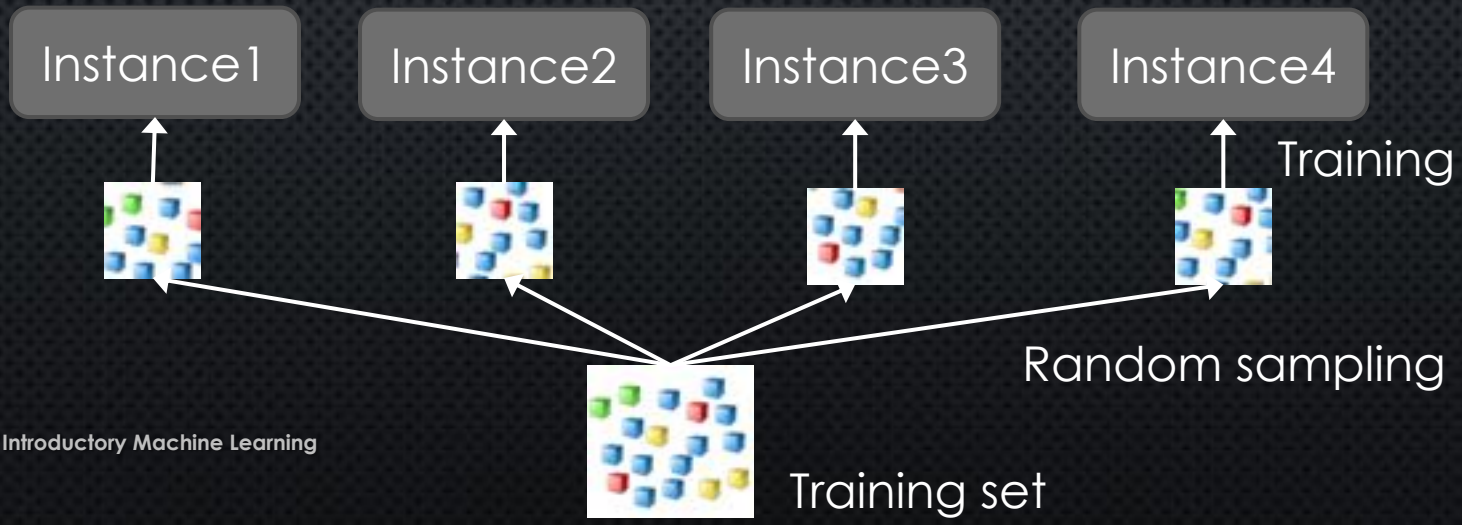


Voting classifier – soft voting

- **Soft voting** is available if the predictors can estimate class posterior propagability
  - e.g. naïve Bayes, SVC and DTs
  - Average over all individual predictors
  - May achieve better accuracy than hard voting

Bagging and pasting

- Train several **instances** of a predictor using different random subsets of training set.
  - Ensemble methods work best when the predictors are as independent from one another as possible
- Rationale: Increases the chance to make different type of errors, resulting in better ensemble's accuracy



Bagging and pasting

- **Bagging**: sampling the training data with replacement
- **Pasting**: sampling the training data without replacement
- **Both** allow a training instance to be sampled many times for multiple predictors; only bagging allows a training instance to be sampled several times for a same predictor

Bagging and pasting

- **Bagging**: introduce more diversity in subsets for training
  - Higher bias than pasting; less correlated between predictors - reduced ensemble's variance
  - Generally preferred for better models
- **Pasting**: might lead to a model that is more robust in external validations
- Advantage: **both** allow predictors to be trained concurrently using parallel computing

Out-of-bag evaluation

- Bagging may lead to some data not being sampled at all
  - Approximately 63% of training samples are used for each model instance. i.e. $1 - \exp(-1) \approx 63.21\%$
  - Unused samples: out-of-bag(OOB) samples

- OOB samples can be used to evaluate the predictor!
  - The performance of an ensemble may be measured as the average OOB evaluations of each predictor

Random Forests (RF)

- A Random Forest (RF) is an ensemble of **Decision Trees**
  - DTs are often trained using bagging method
  - Number of samples/bag: size of the training set

- RF introduces extra randomness when growing trees
  - DT searches for very best feature when generating branches; RF seeks through a random subset of features
  - Greater tree diversity; higher bias, lower variance
  - An overall better model

Extremely Randomised Trees

- **Extra-Trees (ET)** for short
- More random procedure when splitting for branches in a tree
  - RF seeks for the *best* possible criteria among a random subset of features; ET use random thresholds
  - Faster to train than RF
  - Even higher bias, lower variance

DT, RF and ET

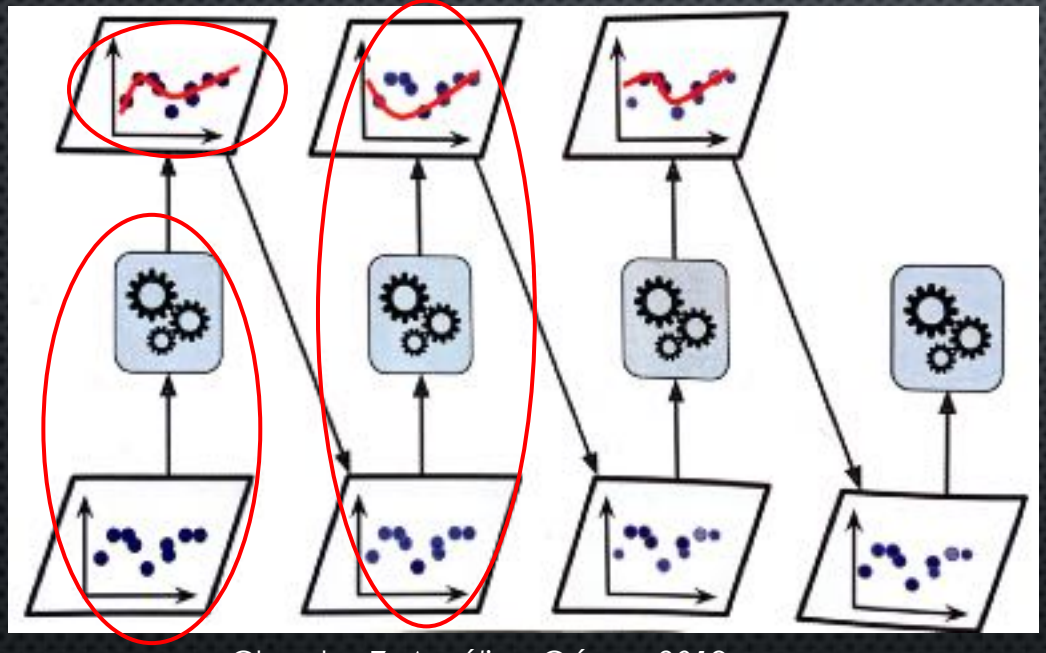
	No. of Trees	Features for branching	Criteria for branching
DT	single	All	Optimal
RF	multiple	Random subset	Optimal
ET	multiple	Random subset	Random

Recall the lecture of Week 6: an optimal criterion is usually sought by minimising the sum of the weighted Gini impurity of the latent left and right splits

Boosting

- Hypothesis boosting – ***boosting*** conventionally
  - An ensemble method that creates a strong classifier by combining several weak classifiers
- Boosting is to train predictors in sequence, each attempts to improve over its predecessor

Boosting: Adaptive Boosting



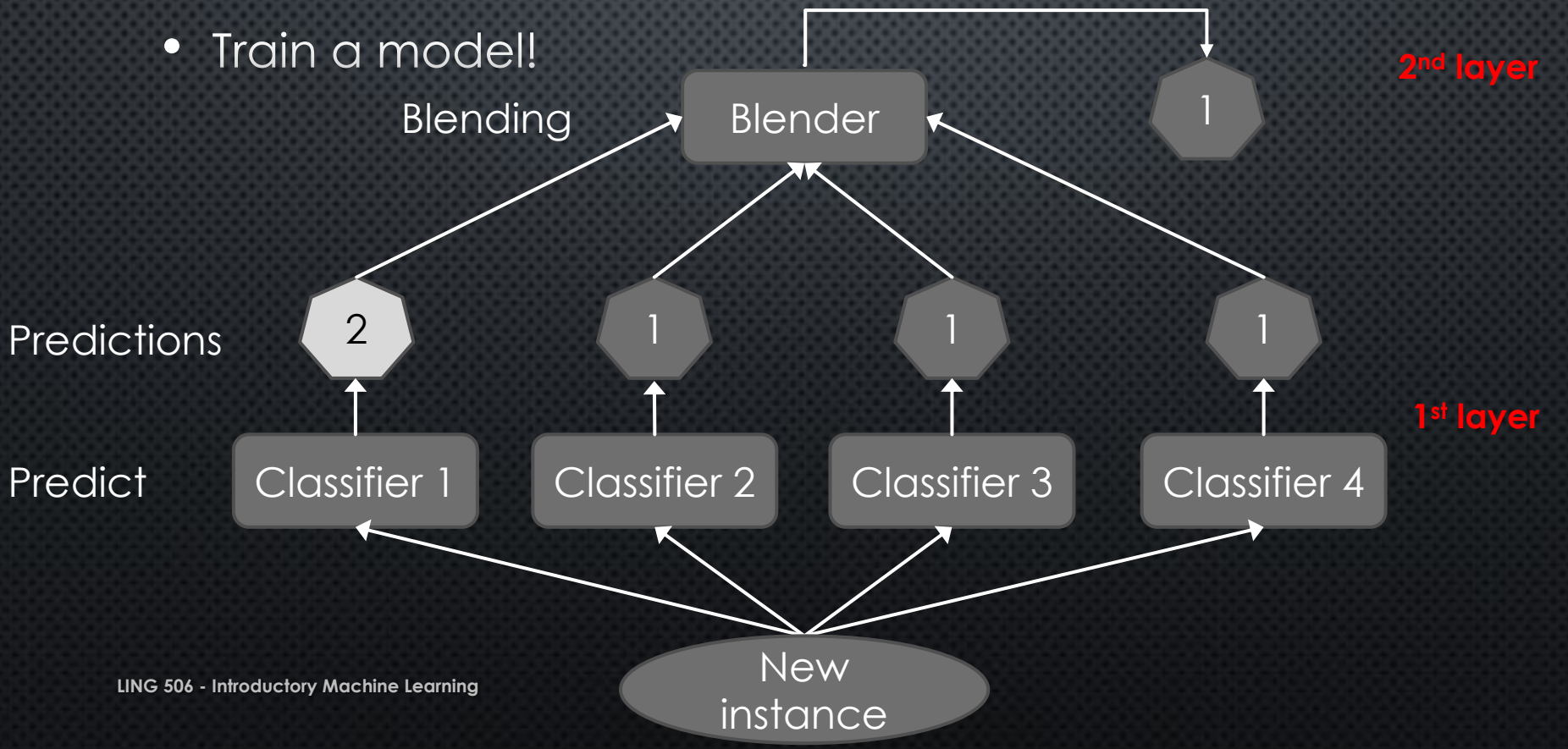
Chapter 7, *Aurélien Géron, 2019*

- **AdaBoost** for short

1. Train a base predictor to make predictions on training set
2. Increase the weight of misclassified training samples
3. Train the next predictor using the updated weights, and make predictions; further update weights
4. Repeat 3 until desired number of predictors are trained

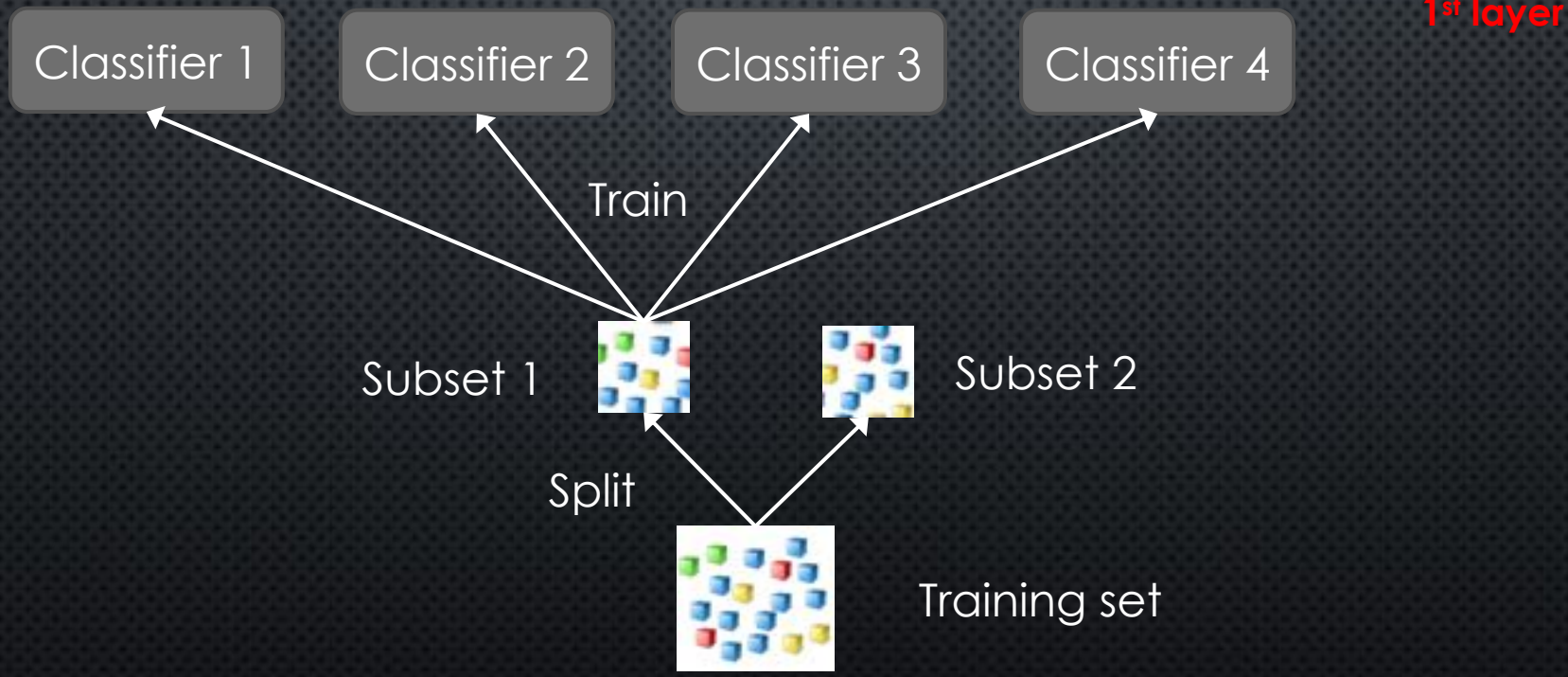
Stacking: stacked generalisation

- Is there any other way, apart from hard/soft voting, that can be used to aggregated the predictions of all predictors in an ensemble?
  - Train a model!



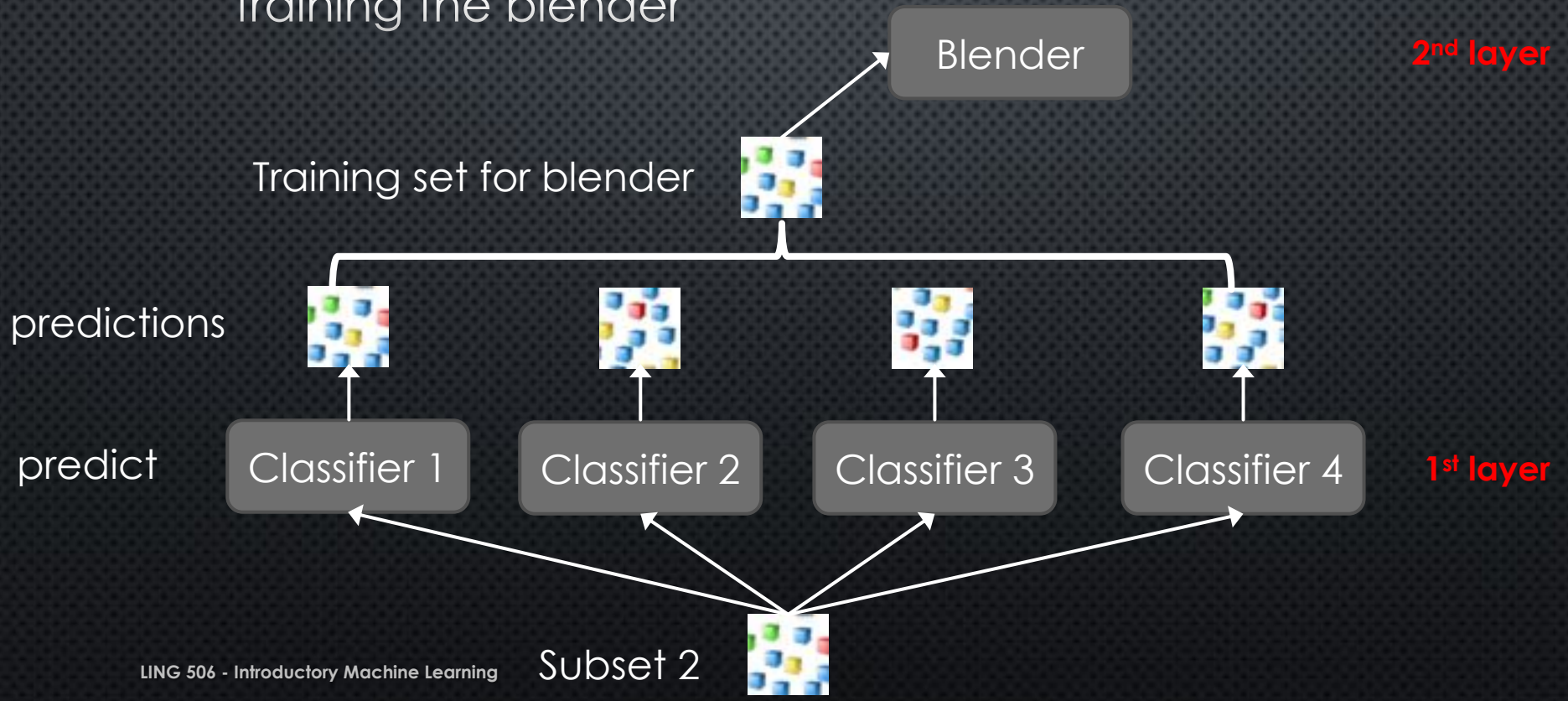
Stacking: Training 1st layer predictors

- Partition the training data into two
  - Use the first set to train the predictors in the first layer



Stacking: Training 2nd layer blender

- Use 1st-layer predictors to make predictions on the second set
  - Combine the predictions to form the input features for training the blender



Stacking: Further extension

- Multi-blender layer!
  - Partition the training data into three subsets, each of which is used for training each layer.