Ensembles

Ensemble is a group of complementary models all addressing the same problem. Try to combine the predictions of the models in the group to come up with a better prediction than any individual model.

- **No Free Lunch Theorem** - *If a model performs well for a set of problems, then it will suck at all the remaining problems* - **there is no model that is the best for all problems**;
- Instead of learning a single complex model, we can **learn many simpler models and combine them**;
- Train each model, and then, in the presence of a new record to classify, **each model gives its classification**, which plays the role of a **vote**.

There are several **voting schemes**:

- **Unanimous Vote** - all models must agree on the classification;
- **Majority Vote** - the classification with the most votes is chosen;
- **Weighted Majority Vote** - each model has a weight, and the classification with the most weighted votes is chosen;

Ensembles can be seen as the **decomposition of a complex problem** into multiple simpler ones. To do so, we need a **set of weak, independent, and diverse models**:

- **Weak model** - model that performs slightly better than random guessing;
- **Diverse models** - models that output different predictions.

Advantages

- Increase **accuracy** and **robustness** of single models;
- With **diverse models** combined, **random errors cancel each other out**, and the **correct decisions are reinforced**.

Train Diverse Models

- Use different learning approaches;
- Use different parameters;
- Use different training sets;
- Describe the data with different features.

There are two main approaches to build ensembles:

- **Bagging - B**ootstrap **agg**regat**ing -** train a set of models with **samples from the original dataset**;
  - **Random Forests**;
- **Boosting -** train a set of models with **samples from the original dataset**, but **each sample is re-weighted**;
  - **Gradient Boosting**.

Bagging

> **Bagging - B**ootstrap **agg**regat**ing -** train a set of models with **samples from the original dataset**.

- D - **training set**;
- k - **number of models**;
- c - **target class**;
- A **bagging ensemble** picks k **samples** with **replacement** of n records and trains a model over each sample; The ensemble to deliver is the **union of the** k **models - diversity** is achieved by **different training sets**;
- Bagging uses **majority voting** to classify a new record, since it is optimal;
- Bagging tends to use **resampling rather than re-weighting**, giving equal importance to all models;

Random Forests

- **Random Forests - bagging** with **decision trees**;
- Ensemble of decision trees trained over a **re-sample with replacement** of the original dataset, and using a random selection of the variables to determine the split;
- In classification, the **majority vote** is used to classify a new record;
- k - **number of trees**;
- n - **number of records**;
- d - **number of variables**;
- **Maximum number of samples**: $n \times 2^(d+1)$ for binary classification;
  - In general: $nx \prod_{i=1}^{d+1} |V_i|$;
  - $|V_i|$ - number of values of the i-th variable;
- **Maximum number of trees**: $k \leq 2^d$ for binary classification;
  - In general: $k \leq 2 \times \sum_{i=1}^{d} |V_i| - 1$.

Advantages

- **Fast** to train - by choosing a random subset of variables, the trees are smaller and faster to train;
- **Efficiency** increases with feature selection, which by dropping out redundant features, reduces the number of splits;
- **Accuracy** increases.

Boosting

Boosting - learn a set of models that **depend on each other**, which then vote according to a **weighted majority vote**.

- Each model is learned on the original dataset, where each record has a **weight** associated with it;
- The weight given to each model depends on its **performance**;
- In every iteration, the **weights of the records are updated**;
- **Diversity** is achieved by **re-weighting** the training set;

- Like bagging, it also bases its model in training a set of **weak models**;
- It may fail when:
  - there is not enough data;
  - the data is too noisy;
  - the weak learners are too weak.

Gradient Boosting (XGBoost)

- **Gradient Boosting - boosting** with **decision trees**;
- **Records** represented as conjunctions of propositions;
- **Training**:
  - Learn k complementary decision trees from **different weighted datasets**;
  - Create a new model from **combining the previous and last one**, weighting the latter;
- **Classify** a new record by **combining the predictions** of all models, weighting each one by its performance;
- Combines **gradient descent and logistic regression -** used to assign weights to the records;
- **Pseudo-residuals - error made in the classification of each record**;
  - Instead of assigning them to the records as their new weights, they are going to be used as the target variable to learn the new model.