Ensemble Learning
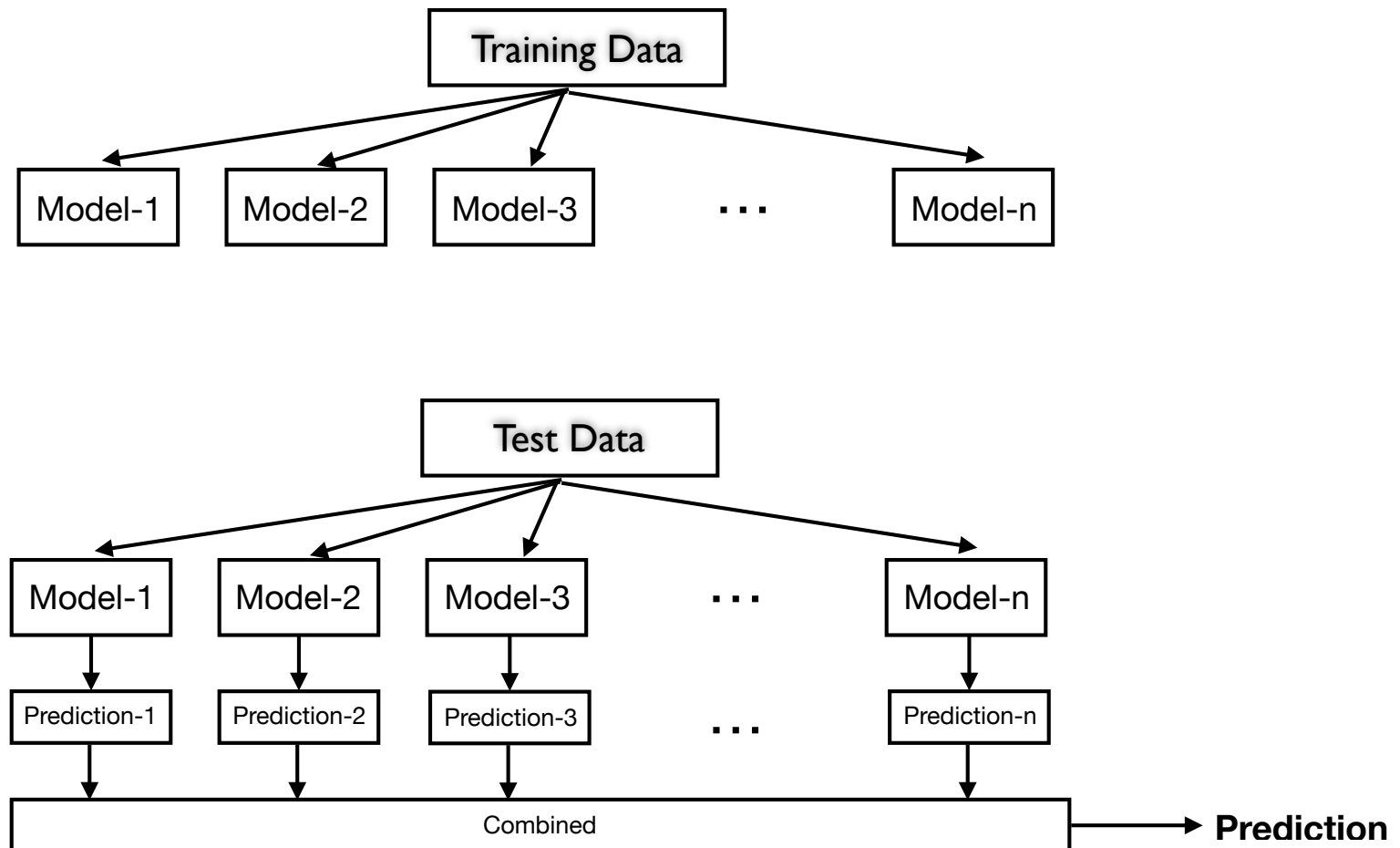
Ensemble Methods

- Ensembles are machine learning methods for combining predictions from multiple separate models.

- The central motivation is rooted under the belief that a committee of experts working together can perform better than a single expert.

```
                        ┌──────────────┐
                        │ Training Data │
                        └──────────────┘
        ┌────────┬──────────┼──────────┬─────────────┐
   ┌─────────┐ ┌─────────┐ ┌─────────┐         ┌─────────┐
   │ Model-1 │ │ Model-2 │ │ Model-3 │  . . .  │ Model-n │
   └─────────┘ └─────────┘ └─────────┘         └─────────┘


                        ┌──────────────┐
                        │  Test Data   │
                        └──────────────┘
        ┌────────┬──────────┼──────────┬─────────────┐
   ┌─────────┐ ┌─────────┐ ┌─────────┐         ┌─────────┐
   │ Model-1 │ │ Model-2 │ │ Model-3 │  . . .  │ Model-n │
   └─────────┘ └─────────┘ └─────────┘         └─────────┘
        │           │           │                   │
   ┌──────────┐ ┌──────────┐ ┌──────────┐      ┌──────────┐
   │Prediction-1│ │Prediction-2│ │Prediction-3│  . . . │Prediction-n│
   └──────────┘ └──────────┘ └──────────┘      └──────────┘
        │           │           │                   │
   ┌──────────────────────────────────────────────────────┐
   │                     Combined                          │──→ Prediction
   └──────────────────────────────────────────────────────┘
```

- Both Regression and Classification can be done using Ensemble learning

- Combining the individual predictions can be done by using either voting or averaging

- The individual ensemble learners need to be:

  - Different from each other (independent errors)

  - Can be weak (slightly better than random): Because of the number of models in an Ensemble method, computational requirements are much higher than that of evaluating a single model. So ensembles are a way to compensate for poor models by performing a lot of extra computation.

Common Ensemble Techniques

- Bagging (Bootstrap Aggregation)

  - Reduced chances of over fitting by training each model only with a randomly chosen subset of the training data. Training can be done in <u>parallel.</u>

  - Essentially trains a large number of "strong" learners in parallel (each model is an over fit for that subset of the data)

  - Combines (averaging or voting) these learners together to "smooth out" predictions.

- Boosting

  - Trains a large number of "weak" learners in sequence. A weak learner is a simple model that is only slightly better than random (eg. One depth decision tree).

  - Miss-classified data weights are increased for training the next model. So training has to be done in <u>sequence</u>.

  - Boosting then combines all the weak learners into a single strong learner.

<u>Bagging</u> uses complex models and tries to "smooth out" their predictions, while <u>Boosting</u> uses simple models and tries to "boost" their aggregate complexity.

- **AdaBoosting** (Adaptive Boosting)

  - In AdaBoost, the successive learners are created with a focus on the ill fitted data of the previous learner

  - Each successive learner focuses more and more on the harder to fit data i.e. their residuals in the previous tree

- **Gradient Boosting**

  - Each learner is fit on a modified version of original data (original data is replaced with the x values and residuals from previous learner

  - By fitting new models to the residuals, the overall learner gradually improves in areas where residuals are initially high