ENSEMBLE MODELS

Outline

- Voting Models
- Bagging
- Random Forests
- Boosting

Ensemble Models

- Ensemble methods improve prediction accuracy by combining the outputs of multiple base models. Bagging (Bootstrap Aggregation) reduces variance by averaging over bootstrapped trees (e.g., Random Forests), while Boosting sequentially builds trees that correct previous errors to reduce bias (e.g., Gradient Boosting).

Use Cases

- Fraud detection
- Customer churn
- Search engine ranking
- Health risk prediction

Strengths

- Excellent predictive performance
- Reduces variance and bias depending on method
- Handles large and high-dimensional datasets well
- Variable importance measures are available

Weaknesses

- Models are harder to interpret
- Computationally intensive
- Requires careful tuning of model hyperparameters

Combining Models

- An analyst may evaluate a set of models to determine the best model
- As it turns out, the other models may not be useless as they may have captured certain patterns that the best model did not.
- Combining the predictions of multiple models will often result in better performance than the best individual model.
- Ensemble models take on a wisdom of the crowds approach by combining predictions from a number of models.
- Two heads are better than one, Many heads are even better.

Ensemble Models

- Combine multiple base models to form a stronger learner
- Bias–Variance Trade-off: decrease variance and/or bias
- Increase predictive accuracy and stability across datasets
- Provide robustness to noisy data and outliers

VOTING MODELS

Voting Approach

- Combine predictions from different models
- Works best when
  - *Models are diverse*
  - *Predictions are independent and uncorrelated*
- Common methods for combining predictions
  - *Hard Voting: Majority Vote. Only works for classification models*
  - *Soft Voting: Average predictions. Applicable to both regression and classification problems*

Ensemble Type

- Voting Models work well for combining predictions of a heterogeneous set of models. For e.g., combining results of logistic regression and trees.
- Next, we will look at ensembles designed for use with the same set of models. E.g., combining a number of tree models.

BAG

Bootstrap AGgregation

Bag Models

- Involve combining a large number of models (of the same type)
- Two models of the same type fitted on the same data will generate identical results. Therefore, rather than using the same data, each model is trained on bootstrapped samples.

Bootstrapping Process

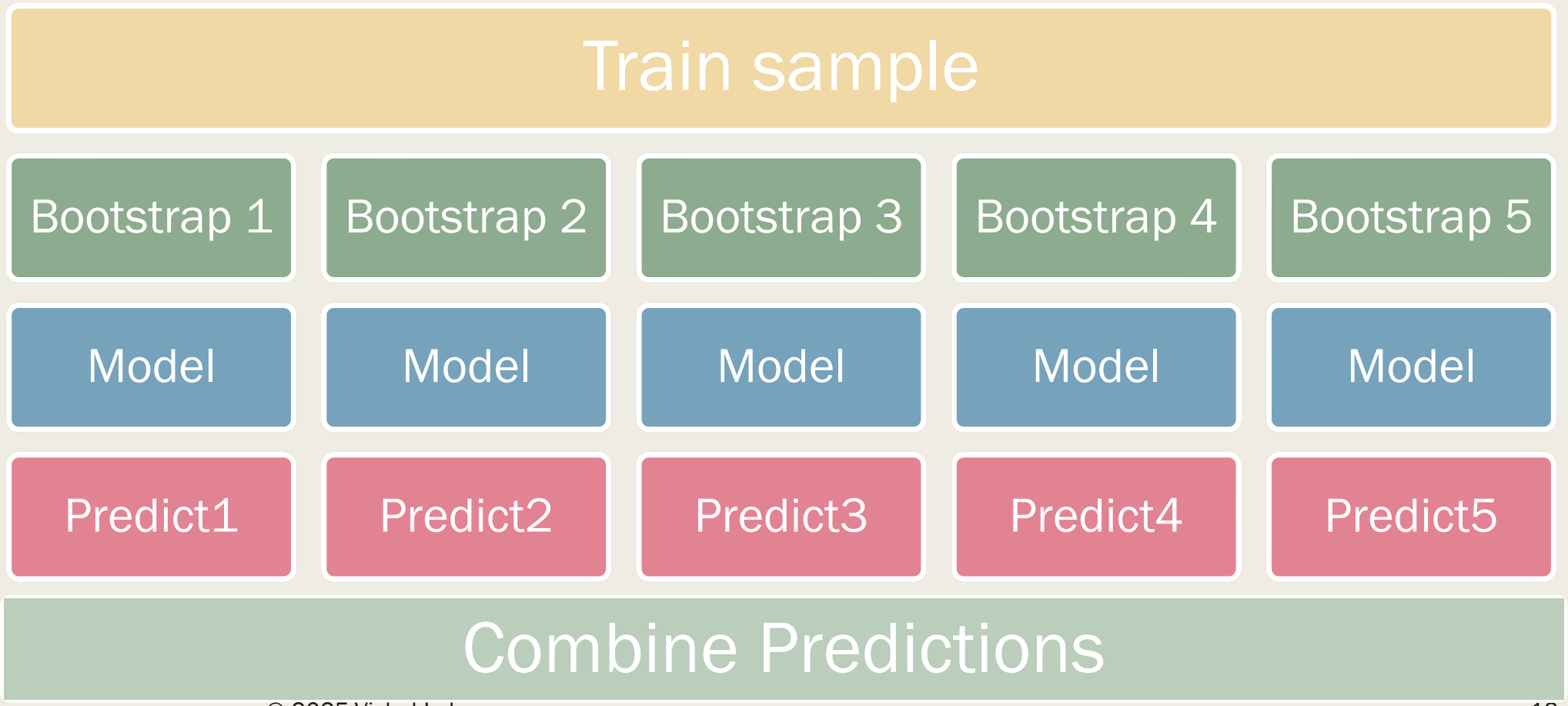
- Draw samples from original data with replacement
- Repeat multiple times

original	bootstrap1	bootstrap2	bootstrap3	bootstrap4	bootstrap5
1	5	2	2	5	3
2	1	3	5	2	3
3	3	3	4	2	5
4	1	2	1	4	5
5	1	2	1	5	2

Bagging

- Bagging involves three steps
  - *Generating a large number of bootstrapped samples from train sample*
  - *Train the model on each sample*
  - *Combining models by averaging (metric outcome) or majority vote (non-metric outcome)*

Bagging Process



Bagging

- Trees constructed in bagging are not pruned, so they tend to be very large trees. But, that is okay, because we are going to average a large number of trees.
- Averaging predictions reduces variance while leaving bias unchanged

RANDOM FORESTS

Bag vs Random Forest

- A special case of Bag Models which use
  - *Tree models as estimators*
  - *Use a subset of features for each model*

Random Forests

- Random forests provide an improvement over bagged trees by way of a small tweak that de-correlates the trees. This reduces the variance when we average the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors.
- A fresh selection of m predictors is taken at each split. The default chosen for m for classification is root $\sqrt{p}$ and for regression is p/3.
- Random Forest with m = p is the same as a Bag model

Why consider a random sample of predictors?

- Suppose that we have a very strong predictor in the data set along with a number of other moderately strong predictor, then in the collection of bagged trees, most or all of them will use the very strong predictor for the first split!
- All bagged trees will look similar. Hence all the predictions from the bagged trees will be highly correlated
- Averaging many highly correlated quantities does not lead to a large variance reduction, and thus random forests “de-correlates” the bagged trees leading to more reduction in variance

BOOSTING

Boosting

- Like bagging, boosting is a general approach that can be applied to many statistical learning methods for regression or classification.
- Recall that bagging involves creating multiple copies of the original training data set using the bootstrap, fitting a separate decision tree to each copy, and then combining all of the trees in order to create a single predictive model.
- Notably, each tree is built on a bootstrap data set, independent of the other trees.
- Boosting works in a similar way, except that the trees are grown sequentially: each tree is grown using information from previously grown trees.

Approach

- Unlike fitting a single large decision tree to the data, which amounts to fitting the data hard and potentially overfitting, the boosting approach instead learns slowly.
- Given the current model, we fit a decision tree to the residuals from the model. We then add this new decision tree into the fitted function in order to update the residuals.
- Each of these trees can be rather small, with just a few terminal nodes, determined by the parameter d in the algorithm.

Types

- Adaboost: Works by increasing weight of incorrectly classified observations in subsequent trees
- Gradient Boosting: Fit small trees to the residuals from the previous tree. This slowly improves performance in area where the model does not perform well. The learning rate can be modified to slow the process down even further, allowing more and different shaped trees to attack the residuals.
- Stochastic Gradient Boosting: Similar to gradient boosting, except each tree is trained on a random subset of the training data. Furthermore, at each node, a random set of features are chosen.
- XGBoost, lightGBM, CatBoost: These boosting algorithms are optimized to scale up to large datasets and run efficiently.

Tuning Parameters

- Boosting models are prone to overfitting.
- Choice of hyperparameters is critical.

Summary

In this module, we examined

- Voting models
- Bag models
- Random Forest models
- Boosting models