**Gradient Boosting (GB)**

Gradient Boosting is a popular boosting algorithm that has a proven track record of winning many machine learning competitions.

**2. Gradient Boosted Trees**

In gradient boosting, each predictor in the ensemble corrects its predecessor's error. In contrast to AdaBoost, the weights of the training instances are not tweaked. Instead, each predictor is trained using the residual errors of its predecessor as labels. In the following slides, you'll explore the technique known as gradient boosted trees where the base learner is a CART.

**3. Gradient Boosted Trees for Regression: Training**

To understand how gradient boosted trees are trained for a regression problem, take a look at the diagram here. The ensemble consists of N trees. Tree1 is trained using the features matrix X and the dataset labels y. The predictions labeled y1hat are used to determine the training set residual errors r1. Tree2 is then trained using the features matrix X and the residual errors r1 of Tree1 as labels. The predicted residuals r1hat are then used to determine the residuals of residuals which are labeled r2. This process is repeated until all of the N trees forming the ensemble are trained.

**4. Shrinkage**

An important parameter used in training gradient boosted trees is shrinkage. In this context, shrinkage refers to the fact that the prediction of each tree in the ensemble is shrinked after it is multiplied by a learning rate eta which is a number between 0 and 1. Similarly to AdaBoost, there's a trade-off between eta and the number of estimators. Decreasing the learning rate needs to be compensated by increasing the number of estimators in order for the ensemble to reach a certain performance.

**5. Gradient Boosted Trees: Prediction**

Once all trees in the ensemble are trained, prediction can be made. When a new instance is available, each tree predicts a label and the final ensemble prediction is given by the formula shown on this slide. In scikit-learn, the class for a gradient boosting regressor is GradientBoostingRegressor. Though not discussed in this course, a similar algorithm is used for classification problems. The class implementing gradient-boosted-classification in scikit-learn is GradientBoostingClassifier.

**6. Gradient Boosting in sklearn (auto dataset)**

Great! Now it's time to get your hands dirty by predicting the miles per gallon consumption of cars in the auto-dataset. Note that the dataset is already loaded. First, import GradientBoostingRegressor from sklearn.ensemble. Also, import the functions train\_test\_split and mean\_squared\_error as MSE as shown here. Then split the dataset into 70%-train and 30%-test.

**7. Gradient Boosting in sklearn (auto dataset)**

Now instantiate a GradientBoostingRegressor gbt consisting of 300 decision-stumps. This can be done by setting the parameters n\_estimators to 300 and max\_depth to 1. Finally, fit gbt to the training set and predict the test set labels. Compute the test set RMSE as shown here and print the value. The result shows that gbt achieves a test set RMSE of 4-dot-01.

**Stochastic Gradient Boosting (SGB)**

**2. Gradient Boosting: Cons**

Gradient boosting involves an exhaustive search procedure. Each tree in the ensemble is trained to find the best split-points and the best features. This procedure may lead to CARTs that use the same split-points and possibly the same features.

**3. Stochastic Gradient Boosting**

To mitigate these effects, you can use an algorithm known as stochastic gradient boosting. In stochastic gradient boosting, each CART is trained on a random subset of the training data. This subset is sampled without replacement. Furthermore, at the level of each node, features are sampled without replacement when choosing the best split-points. As a result, this creates further diversity in the ensemble and the net effect is adding more variance to the ensemble of trees.

**4. Stochastic Gradient Boosting: Training**

Let's take a closer look at the training procedure used in stochastic gradient boosting by examining the diagram shown on this slide. First, instead of providing all the training instances to a tree, only a fraction of these instances are provided through sampling without replacement. The sampled data is then used for training a tree. However, not all features are considered when a split is made. Instead, only a certain randomly sampled fraction of these features are used for this purpose. Once a tree is trained, predictions are made and the residual errors can be computed. These residual errors are multiplied by the learning rate eta and are fed to the next tree in the ensemble. This procedure is repeated sequentially until all the trees in the ensemble are trained. The prediction procedure for a new instance in stochastic gradient boosting is similar to that of gradient boosting.

**5. Stochastic Gradient Boosting in sklearn (auto dataset)**

Alright, now it's time to put this into practice. As in the last video, we'll be dealing with the auto-dataset which is already loaded. Perform the same imports that were introduced in the previous lesson and split the data.

**6. Stochastic Gradient Boosting in sklearn (auto dataset)**

Now define a stochastic-gradient-boosting-regressor named sgbt consisting of 300 decision-stumps. This can be done by setting the parameters max\_depth to 1 and n\_estimators to 300. Here, the parameter subsample was set to 0-dot-8 in order for each tree to sample 80% of the data for training. Finally, the parameter max\_features was set to 0-dot-2 so that each tree uses 20% of available features to perform the best-split. Once done, fit sgbt to the training set and predict the test set labels.

**7. Stochastic Gradient Boosting in sklearn (auto dataset)**

Finally, compute the test set RMSE and print it. The result shows that sgbt achieves a test set RMSE of 3-dot-95.