1. What is the estimated depth of a Decision Tree trained (unrestricted) on a one million instance training set?

2. Is the Gini impurity of a node usually lower or higher than that of its parent? Is it always lower/greater, or is it usually lower/greater?

3. Explain if its a good idea to reduce max depth if a Decision Tree is overfitting the training set?

4. Explain if its a good idea to try scaling the input features if a Decision Tree underfits the training set?

5. How much time will it take to train another Decision Tree on a training set of 10 million instances if it takes an hour to train a Decision Tree on a training set with 1 million instances?

6. Will setting presort=True speed up training if your training set has 100,000 instances?

7. Follow these steps to train and fine-tune a Decision Tree for the moons dataset:

a. To build a moons dataset, use make moons(n samples=10000, noise=0.4).

b. Divide the dataset into a training and a test collection with train test split().

c. To find good hyperparameters values for a DecisionTreeClassifier, use grid search with cross-validation (with the GridSearchCV class). Try different values for max leaf nodes.

d. Use these hyperparameters to train the model on the entire training set, and then assess its output on the test set. You can achieve an accuracy of 85 to 87 percent.

8. Follow these steps to grow a forest:

a. Using the same method as before, create 1,000 subsets of the training set, each containing 100 instances chosen at random. You can do this with Scikit-ShuffleSplit Learn's class.

b. Using the best hyperparameter values found in the previous exercise, train one Decision Tree on each subset. On the test collection, evaluate these 1,000 Decision Trees. These Decision Trees would likely perform worse than the first Decision Tree, achieving only around 80% accuracy, since they were trained on smaller sets.

c. Now the magic begins. Create 1,000 Decision Tree predictions for each test set case, and keep only the most common prediction (you can do this with SciPy's mode() function). Over the test collection, this method gives you majority-vote predictions.

d. On the test range, evaluate these predictions: you should achieve a slightly higher accuracy than the first model (approx 0.5 to 1.5 percent higher). You've successfully learned a Random Forest classifier!

Answer:

1. The estimated depth of a Decision Tree trained on a one million instance training set is difficult to estimate without knowing other details about the dataset and the hyperparameters of the model. Generally, as the size of the training set increases, the depth of the Decision Tree tends to increase as well.
2. The Gini impurity of a node is usually lower than that of its parent because splitting the node into child nodes tends to reduce impurity. However, this is not always the case, and there can be situations where the impurity of the child nodes is higher than that of the parent node.
3. If a Decision Tree is overfitting the training set, it is generally a good idea to reduce the max depth. This can prevent the model from becoming too complex and overfitting the training data, which can lead to poor generalization performance on new data.
4. Scaling the input features is unlikely to help if a Decision Tree is underfitting the training set because Decision Trees are not sensitive to the scale of the input features.
5. It is difficult to estimate exactly how long it would take to train another Decision Tree on a training set of 10 million instances because the time required for training depends on many factors such as the complexity of the model, the size of the training set, and the computational resources available. However, assuming all other factors are constant, it might take around 10 hours to train a Decision Tree on a training set with 10 million instances.
6. Setting presort=True is unlikely to speed up training if the training set has 100,000 instances because presorting can be computationally expensive and is only beneficial for small datasets.
7. These are the steps to train and fine-tune a Decision Tree for the moons dataset:

a. Use make\_moons(n\_samples=10000, noise=0.4) to create a moons dataset with 10,000 samples and 0.4 noise.

b. Use train\_test\_split() to split the dataset into a training and a test set.

c. Use GridSearchCV to perform a grid search with cross-validation to find good hyperparameter values for a DecisionTreeClassifier. Try different values for max\_leaf\_nodes.

d. Train the model on the entire training set using the best hyperparameters found in step c, and evaluate its performance on the test set. You should achieve an accuracy of 85 to 87 percent.

1. These are the steps to grow a forest:

a. Use ShuffleSplit to create 1,000 subsets of the training set, each containing 100 instances chosen at random.

b. Train one Decision Tree on each subset using the best hyperparameter values found in the previous exercise. Evaluate these 1,000 Decision Trees on the test set.

c. Create 1,000 Decision Tree predictions for each test set case, and keep only the most common prediction using SciPy's mode() function. This gives you majority-vote predictions.

d. Evaluate the predictions on the test set to get a slightly higher accuracy than the first model (approx 0.5 to 1.5 percent higher). This is a Random Forest classifier.