1. Is there any way to combine five different models that have all been trained on the same training data and have all achieved 95 percent precision? If so, how can you go about doing it? If not, what is the reason?

Answer: Try combining them into a voting ensemble, which will often give you even better results. It works better if the models are very different.

2. What's the difference between hard voting classifiers and soft voting classifiers?

Answer:

* A hard voting classifier just counts the votes of each classifier in the ensemble and picks the class that gets the most votes.
* A soft voting classifier computes the average estimated class probability for each class and picks the class with the highest probability.

3. Is it possible to distribute a bagging ensemble's training through several servers to speed up the process? Pasting ensembles, boosting ensembles, Random Forests, and stacking ensembles are all options.

Answer: It is quite possible to speed up training of a bagging ensemble, pasting ensembles and Random Forests by distributing it across multiple servers, since each predictor in the ensemble is independent of the others.

4. What is the advantage of evaluating out of the bag?

Answer:

Each predictor in a bagging ensemble is evaluated using instances that it was not trained on (they were held out). This makes it possible to have a fairly unbiased evaluation of the ensemble without the need for an additional validation set. Thus, you have more instances available for training, and your ensemble can perform slightly better.

5. What distinguishes Extra-Trees from ordinary Random Forests? What good would this extra randomness do? Is it true that Extra-Tree Random Forests are slower or faster than normal Random Forests?

Answer: Random forest uses bootstrap replicas, that is to say, it subsamples the input data with replacement, whereas Extra Trees use the whole original sample. In the Extra Trees sklearn implementation there is an optional parameter that allows users to bootstrap replicas, but by default, it uses the entire input sample. This may increase variance because bootstrapping makes it more diversified.

Another difference is the selection of cut points in order to split nodes. Random Forest chooses the optimum split while Extra Trees chooses it randomly. However, once the split points are selected, the two algorithms choose the best one between all the subset of features. Therefore, Extra Trees adds randomization but still has optimization.

These differences motivate the reduction of both bias and variance. On one hand, using the whole original sample instead of a bootstrap replica will reduce bias. On the other hand, choosing randomly the split point of each node will reduce variance.

In terms of computational cost, and therefore execution time, the Extra Trees algorithm is faster. This algorithm saves time because the whole procedure is the same, but it randomly chooses the split point and does not calculate the optimal one

In the comparison that has been made, the obtained results are practically the same for both algorithms. It is worth noting the difference in the execution time, where the Extra Trees is much faster. Therefore, if you have any doubt about which of the two ensembles to use, it seems a good idea to use Extra Trees since the same result is obtained in a faster way.

6. Which hyperparameters and how do you tweak if your AdaBoost ensemble underfits the training data?

Answer: Try increasing the number of estimators or reducing the regularization hyperparameters of the base estimator, also try slightly increasing the learning rate.

7. Should you raise or decrease the learning rate if your Gradient Boosting ensemble overfits the training set?

Answer: Decreasing the learning rate, early stopping to find the right number of predictors (you probably have too many).