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?

Ans: Yes, we can do that by using Voting ensemble algorithm

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

1. **Hard Voting:**
   1. Hard voting classifier classifies input data based on the **mode** of all the predictions made by different classifiers. The majority voting is considered differently when weights associated with the different classifiers are equal or otherwise.
   2. In hard voting, the predicted output class is a class with the highest majority of votes i.e the class which had the highest probability of being predicted by each of the classifiers. Suppose three classifiers predicted the *output class (A, A, B)*, so here the majority predicted *A* as output. Hence *A* will be the final prediction.
2. **Soft Voting:**
   1. oft voting classifier classifies input data based on the **probabilities** of all the predictions made by different classifiers.
   2. In soft voting, the output class is the prediction based on the average of probability given to that class. Suppose given some input to three models, the prediction probability for class *A = (0.30, 0.47, 0.53)* and *B = (0.20, 0.32, 0.40)*. So, the average for class *A is 0.4333* and *B is 0.3067*, the winner is clearly class *A* because it had the highest probability averaged by each classifier.

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.

**Yes, it is possible to speed up** the training time of bagging by distributing them on multiple servers because all of them are independent of each other.

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

Better Predictive Model: OOB\_Score **helps in the least variance** and hence it makes a much better predictive model than a model using other validation techniques.

Less Computation: It requires less computation as it allows one to test the data as it is being trained.

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?

* **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.

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

If your AdaBoost ensemble underfits the training data, you can try **increasing the number of estimators or reducing the regularization hyperparameters of the base estimator**. You may 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?

If your Gradient Boosting ensemble overfits the training set, **you should try decreasing the learning rate**. You could also use early stopping to find the right number of predictors (you probably have too many).