Questions:

1. Could you build two classification models that predict a >50K income level, one using GradientBoostingClassifier from sklearn, the other using CatBoost? Please compare the performances of these two models both in prediction accuracy (or any other appropriate metric) and in computation time.
2. The feature *native-country* has over 40 categories. How did the two models handle this variable, what encoding method did they use, respectively?
3. Using the sklearn model you built in 1) as baseline, can you make changes to the encoding method for the variable native-country and improve the performance of the classifier, again measured in both prediction accuracy (or other metrics) and computation time? Can you explain why the change you made improved (or did not improve) the performance? This is an open-ended question. Your method doesn’t have to improve the performance, but an understanding why it did or did not would be appreciated.

**If was using One-Hot then cutting down to a binary difference would help with computation time.**

1. Create an API that can be invoked to train and predict the two classification models. It should be possible to provide necessary inputs and select which model to use through the interface, as well as retrieve results. How would an additional model be supported in the future (can this be demonstrated)?
2. Create a set of unit tests to prove that this software works as expected.
3. (Bonus) Assuming the limitation of 2 available processing cores, create a simulation of N users invoking the API simultaneously, and demonstrate how processing times scale. Choose one model to use for the simulation. Use N values of (1, 2, 4, 8, 20). Run the same simulation assuming 4 processing cores and compare processing times.