Lab 05: Description of process

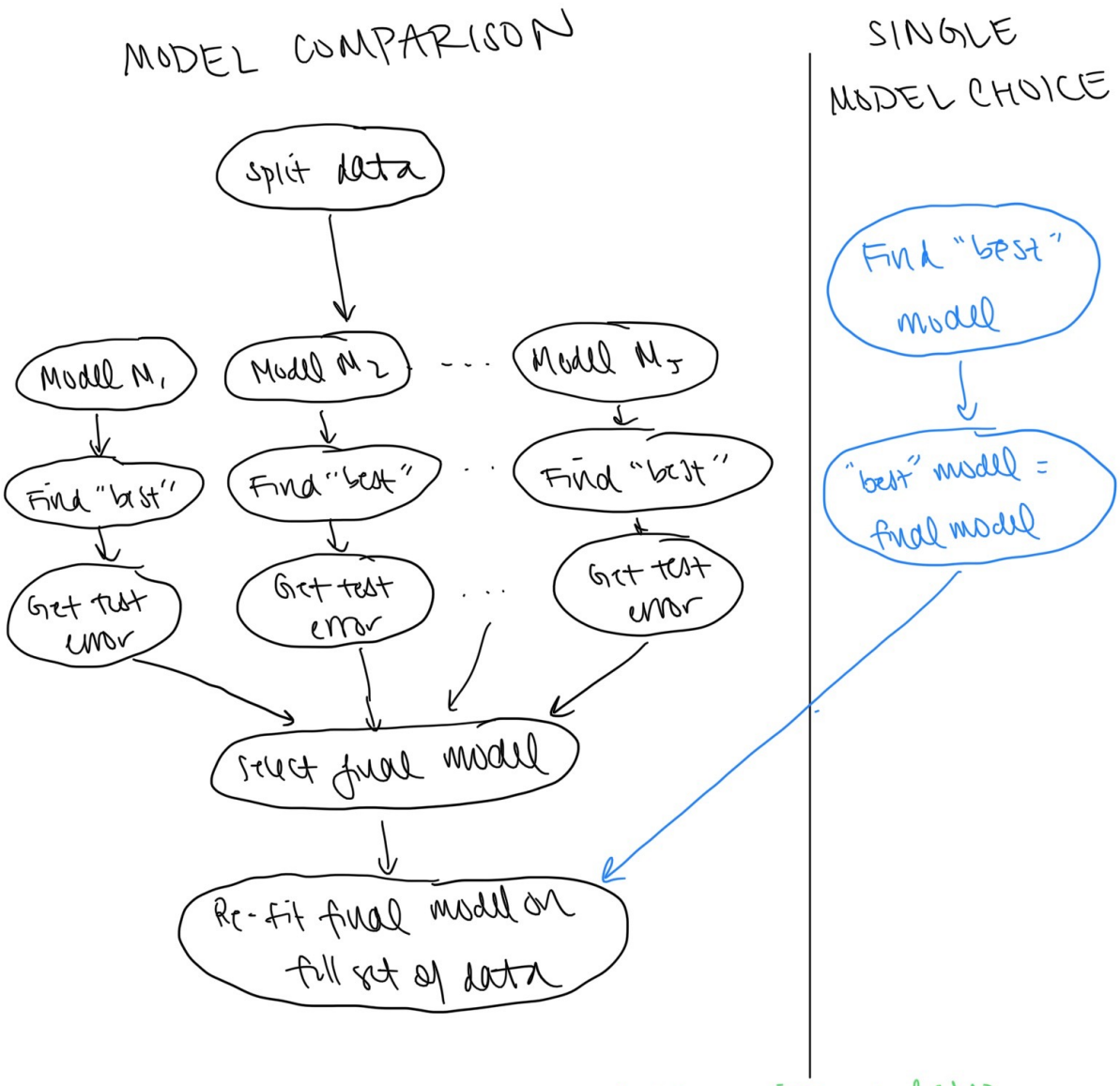
When we want to perform model comparison, we need to compare models using unseen *test* data. This could be achieved any number of ways (validation set approach, LOOCV, k-fold CV, etc.), so long as the same sets of observations are used consistently across the models you are comparing.

In the lab, we wanted to compare ridge and the Lasso models on the `Hitters` data. So we split the data into roughly 50% training, 50% test. However, both of these models also have an extra parameter λ floating around. Therefore, in order to pick the best λ^* for ridge regression, we implement k-fold CV using the set of training observations. Similarly, we pick the best λ^* for Lasso regression using k-fold CV using the same set of training observations. At this point, we have a ridge regression model with an optimal λ^*, and a Lasso model with its optimal λ^*. To compare these two models, we perform predictions on the same set of test data, then compare test errors to choose the "best" model".

If we DID NOT want to compare models, we would not need to split the data into test/train sets. But we we would still have to choose an optimal λ^*. For example, suppose we want to run ridge regression. We could use `cv.glmnet()` on the *full* set of data because the function will automatically split into various test/train folds for us.

In either case, we may be asked to report a final model and its estimated β coefficients. Once we get an optimal λ^*, we should re-fit the ridge (or Lasso) model using `glmnet()` and the chosen λ^*. Why? Any estimates for the coefficients obtained from `cv.glmnet()` do not see all the data at any given time. Thus, in order to minimize uncertainty in our $\hat{\beta}$ estimates, we should run the model using ALL the data.

See the figure on the next page for a more visual/general description!



* Find "best" may be choosing optimal λ^* in ridge & lasso. But if running a linear regression normally, you don't need to choose a "best".