Evaluating Machine Learning Projects

In order to Solve a real a world problem with the help of Machine Learning we need to keep in mind a few basic rules or guidelines which would help us to reach the solution. First things first one should analyze whether the problem actually needs a machine learning solution, not everything in the real world needs a complex Machine Learning solution, certain problems can be well solved much faster and easier using analytical approaches. Secondly once you identify a problem where Machine Learning can be implemented one should always rely on the simple algorithms rather than depending on complex rules or heuristics, for example if we want to heuristically identify cancerous cells from dermatoscopic skin lesion images we have write down thousands of rules. This is a tedious and inaccurate method, ML on the hand adapts from data and gets better with the more data it sees and adapts.

Thirdly it is extremely important to note how you would evaluate the efficiency of the model. A correct choice of evaluation metric is critical to the success of your ML/DL model, not every evaluation metric works well with every given model for example in case of cancer detection accuracy is most the best possible measure because a model with high accuracy might miss out some misclassifications (models detects non-cancerous but it is actually cancerous ) so recall would be a better evaluation metric in this sort of scenario. For our case we are using ROC (Receiver operating Characteristics Curve) for evaluation. Fourthly in order to build a model one should know which feature is to be used (What’s hot and what’s not). Data with innumerable features is good for the models but one should know how to engineer the features for one’s advantage, that’s where feature selection and engineering comes into picture. Feature engineering is an iterative process, it involves combining basic features to produce combined features, choosing features that can be generalized across contexts, choosing specific features for specific scenarios.

Next while training and testing one should have a uniform training and testing datasets, this basically means the distribution in your training dataset should replicate the distribution present in your validation or testing dataset which can be achieve with sampling measures (one can use k-fold cross-validation or leave one out cross validation for this ). Another major learning point should be to keep in mind the randomness of real-world data (it might not always follow a uniform or Gaussian distribution) so you have to keep in mind the presence of outliers (trust me detecting outliers can be a assiduous task ) so one has to account for noise. Subsampling in these situations can come to your rescue. Another important approach to data analysis is asking questions at every step. Of course, the questions you ask can and should evolve as you look at the data. But analysis without a question will end up aimless. While evaluating your model one should be a skeptic and be confident at the same time so one needs to ask questions and come up with answers to them (you are the best critique for your model).

To close off the discussion the three important stages are Validate, Describe and Evaluate. Also these stages do not progress linearly. As you explore the data, you may jump back and forth between the stages, but at any time you should be clear what stage you are in.