**Team Members:** Ankita Kumari, Ashesh Shahi, Uma Sai Madhuri Jetty, Sahithi Priya Gutta

The article “**Rules of Machine Learning: Best Practices for Machine Learning**” is very useful and informative. Out of 43 rules, we have picked some of them which are most important and relatable to our project and have summarizes them:

Keeping the first model simple and getting the infrastructure right is one of the best practices for Machine Learning projects. For our project, we will create simple model first because simple model provides the baseline metrics and baseline behavior that we can use to test more complex models. Also, we should make sure that infrastructure is testable and the training model gives the same score as the testing model so that it will result in a better accuracy. Another important rule is detecting problems before exporting models which is very necessary in case of our project. Because if there is an issue with the model predicting fake job postings, it will create big problem for users. So, it’s better to wait and do sanity check before exporting the model.

Feature column's documentation must be described in detail which tells about what the feature is, its origin, and how much it is relative to our goal. Moreover, we should always start with models that are interpretable like (Linear Regression, Logistic Regression, and Poisson regression) models. These are directly motivated by probabilistic models. Since the prediction is understandable according to probability or an expected value, it makes easier to debug the model rather than those models that uses zero-one loss, various hinge losses etc. Also, we can easily deal with the feedback loops using simple models. Furthermore, rule no 16 is also really important for our project because there always be some new features coming. Therefore, we may combine old features into a new one using regularization parameter. We should also take such thing in consideration like how easy it is to add or remove or recombine features.

As we know that unused features can cause efficiency related issues or technical debt it is always better to remove the features that are no longer used.  We need to always make sure that the infrastructure is clean enough so that the most promising features can be tried as fast as possible. Before removing the feature, one has to make sure if removing it has any effect on data. Never arbitrarily drop data always do an importance-weighting. One is always tempted to drop data when there is too much of it, so instead of just dropping it vaguely (Out of 100 samples picking 1-50) do an importance-weighting probability. Once the cleaning and featuring engineering of the data is done let’s go with the modelling part. Always measure the delta between the models. When a new model is tried compare with the old and try finding any changes, if there is a huge change try identifying if the change is good or bad and always make sure that the system is stable. If there is any unusual behavior, first measure the error and then optimize it. Try creating new features using the patterns discovered while measuring the errors.

Reusability of the code whenever possible is also very important aspect of Machine Learning. We can measure the performance of the data (whether it is clean or not) by sacrificing the performance of the data during serving so that it won’t introduce sampling bias. Also, we can accurately design the model by selecting the features which have positive weights, and which are specific to one or few queries so that the model can differentiate the queries that don’t lead to high regularization. The documentation of features is not appropriate because the query and the related results should be confidential. And also the model should be in the position to measure the difference between the performance of training data and holdout data, and also the performance of holdout data and next day data so that we can tune our regularization in such a way so that it maximizes the next day data performance when compare to holdout data. This can be achieved by taking care of the time sensitive features so that they won’t degrade the model performance.