Evaluating Machine Learning Projects: Forty-Three Rules of Machine Learning

All the rules given by Google ML guidelines are important for designing and deploying end-to-end ML Pipelines, however, the following is the summary of the rules we found useful and related to our Machine Learning Project.

First, design and implement metrics, since we are working on predicting the route that is less prone to accidents, we need Historic data to better design the ML pipeline. Historic data provides us with an objective as to what the system can accomplish if we implement the ML policy.

Keep the first model simple and get the infrastructure right, as we are working on a cloud platform, infrastructure is the first thing we need to manage and getting that right is an essential part of building further data pipelines. In addition to this, **Testing the infrastructure independently from machine learning** is also necessary. As we build a data pipeline, starting from data ingestion to modelling, we need to test the flow of infrastructure and connection between different components of infrastructure viz. Storage bucket, Big Query, Jupyter Notebook Compute VMs, etc. As we are going to use large infrastructure to run our pipeline, it's better to **Detect problems before exporting models**, because it will affect end-users directly and better to do end-to-end testing before exporting and deploying it to production.

Heuristics methods were widely used before Machine Learning and those work well with rule-based solutions. Sometimes it's better to have trustable heuristics in our Machine learning pipeline as they work and give expected results, that's why **Turn heuristics into features, or handle them externally.** In Feature Engineering, it's suggested to **Give feature columns owners and documentation,** as in our project we have around 50 feature columns and we are also doing Feature Generation by combination and grouping of few features, documentation is must-have for all newly generated columns. This also explains that **Combining and Modifying existing features to create new features in human-understandable ways,** as we generated Day of Week features from Start Time of the accident, it’s useful to see the number of accidents on each given weekday. **Using very specific features when you can**, because some of the features are directly correlated to the output label and we can leverage that to improve the accuracy of the model. Also **Cleaning up features you are no longer using**, and **Looking for patterns in the measured errors, and creating new features** are required to make features more impactful to the target label, if provided features are not being useful directly then creating new features would help.

In ML Pipeline, **choosing a simple, observable and attributable metric for your first objective**, is needed to start the process at some point, then we can move to more complex implementation. Metrics are always helpful to analyze the performance of the model and using multiple types of metrics gives direction for improvement. **Reusing code between your training pipeline and your serving pipeline whenever possible,** as we need to keep feature cleaning, generation, and transformation the same for both the pipelines.

If you produce a model based on the data until January 5th, test the model on the data from January 6th and after, this is most important in our project of predicting a particular outcome or suggesting something to someone. We need the ML system to be built on the latest data updated to date as it provides better training and testing and covers all the possibilities since there exists daily activities that might have an impact on the ML scores.

Don't waste time on new features if unaligned objectives have become the issue. In our current development, we have created a new feature **Day_Of_Week** from an existing date column which will help us in a better understanding of the data. Likewise, we should only create new features that will be relevant to the ML model.