Erwin Cheng

L03_Reflection Journal

ITAI 1371

Viswanatha Rao

A Deeper Dive into the Machine Learning Workflow

Before this lab, my perception of machine learning was admittedly nebulous. I pictured it as a kind of digital alchemy where data was fed into a complex algorithm, and answers magically emerged. The process felt monolithic and inaccessible. The initial overview of the multi-step machine learning workflow seemed to confirm my fears; it looked like a long, intimidating checklist. However, working through the wine classification project transformed that checklist into a logical narrative. The ML workflow, which seemed overwhelming initially, revealed itself as a series of interconnected, foundational pillars, where each step builds logically toward the ultimate goal of creating a model that can make reliable predictions on new, unseen data.

My first real insight came from understanding the relationship between Exploratory Data Analysis (EDA) and the subsequent modeling steps. At first, creating visualizations felt like a preliminary, almost skippable, formality. But as I looked at the distribution of wine classes and the correlation between chemical features, I realized EDA is not just about making pretty charts; it’s an intelligence-gathering mission. It’s where you form hypotheses. Seeing that class_1 was the most frequent might influence how I interpret the model's errors later. Noticing a strong correlation between two features might lead me to question if both are necessary. This initial exploration directly informs feature selection and even the choice of model, turning what could be blind guesswork into a more strategic, evidence-based process.

This led to a new appreciation for the most crucial step in the workflow: the train-test split. The concept seemed simple enough—hold back some data for a final exam. Yet, its profound importance didn't fully land until I began to think about the problem of overfitting. I struggled with the concept of overfitting until the practical implications became clear. A model that isn't tested on unseen data is like a student who memorizes the answers to a specific practice test. They might get 100% on that test, but they haven't truly learned the material and will fail when faced with new questions. The test set is our unbiased proctor; it's the only way to know if our model has genuinely learned the underlying patterns in the data or if it has simply created a complex, brittle map of the training data's noise and quirks. The use of stratify=y was another subtle but powerful detail, ensuring our test set was a fair representation of the real world and not skewed by random chance.

The lab's structure also solidified my understanding of the fundamental types of machine learning. The distinction between supervised and unsupervised learning finally clicked when I realized it all comes down to whether you have the "answer key" during training. It’s not just a difference in algorithms; it's a fundamental difference in the *question you are asking*. With the wine dataset, we had the answer key (the wine_class), so we were in the supervised world, asking the model to *predict* a specific outcome. If we had been given the same data without the wine classes and asked to "find natural groupings of wines based on their chemical profiles," we would have been in the unsupervised world, asking the model to *discover* hidden structures. This "prediction vs. discovery" framework is far more intuitive to me than simply memorizing definitions.

This conceptual clarity made the practical model comparison between Logistic Regression and the Decision Tree far more meaningful. The Decision Tree performed better, but the real lesson wasn't just in the accuracy score. It was in understanding *why*. Logistic Regression, being a linear model, is trying to draw straight lines to separate the wine classes in the data. The Decision Tree, on the other hand, can create complex, non-linear boundaries by asking a series of "if-then" questions. The fact that it performed better suggests that the relationship between a wine's chemical makeup and its class isn't simple and linear.

This was a pivotal moment. It demonstrated that choosing a model isn't about finding the "best" one in a vacuum, but about matching the model's capabilities to the inherent complexity of the data. By setting max_depth=3 for the tree, we were actively preventing it from overfitting—we were telling it not to get too specific and to learn more generalizable rules. This single parameter provided a tangible lever to control the balance between learning and memorizing, making the abstract concept of the bias-variance tradeoff much more concrete.

Ultimately, this lab transformed my view of machine learning from a single action to an iterative, thoughtful process. It's a craft that involves not just coding but also critical thinking, investigation, and a healthy dose of skepticism. The goal isn't just to achieve the highest accuracy but to build a model that is understandable, reliable, and truly useful for its intended purpose. I now see the workflow not as a rigid set of instructions, but as a flexible blueprint for methodical problem-solving in a domain of fascinating complexity.