Overview of Debugging ML Models 🔖

You **debug** your ML model to make the model work. Once your model is working, you **optimize** the model's quality for production-readiness. This section covers both the debugging and optimization steps.

How is ML Debugging Different?

Before diving into ML debugging, let's understand what differentiates debugging ML models from debugging typical programs. Unlike typical programs, poor quality in an ML model does not imply the presence of a bug. Instead, to debug poor performance in a model, you investigate a broader range of causes than you would in traditional programming.

For example, here are a few causes for poor model performance:

- Features lack predictive power.

- Hyperparameters are set to nonoptimal values.

- Data contains errors and anomalies.

- Feature engineering code contains bugs.

Debugging ML models is complicated by the time it takes to run your experiments. Given the longer iteration cycles, and the larger error space, debugging ML models is uniquely challenging.

ML Model Development Process

If you follow best practices for developing your ML model, then debugging your ML model will be simpler. These best practices are as follows:

1. Start with a simple model that uses one or two features. Starting with a simple, easily debuggable model helps you narrow down the many possible causes for poor model performance.

2. Get your model working by trying different features and hyperparameter values. Keep your model as simple as possible to simplify debugging.

3. Optimize your model by iteratively trying these changes:

- adding features

- tuning hyperparameters

- increasing model capacity

4. After each change to your model, revisit your metrics and check whether model quality increases. If not, then debug your model as described in this course.

5. As you iterate, ensure you add complexity to your model slowly and incrementally.

For certain domains like NLP or image recognition, you should start with a complex model.