The "Pipeline-First" Syllabus (Advanced)

Rigorous ML Engineering for Econometrics

Based on ISLP & Géron

December 2025

Resource Key

- **[ISLP]** *An Introduction to Statistical Learning with Applications in Python* (2023).
- **[Géron]** *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd Ed).

---

1 Week 1: The Engineering Backbone (Pipelines)

Focus: Reproducibility and Data Leakage Prevention.

Day 1 | Data Cleaning & Imputation

> **Readings:** **[Géron]** Ch 2, Section "Data Cleaning".
> **[ISLP]** Ch 3.3.3 "Potential Problems" (Missing Data).

Task:

1. Load a dirty dataset (e.g., Titanic).
2. Instantiate `SimpleImputer(strategy='median')`.
3. **Verify:** Check that `fit()` computes stats on Train only, and `transform()` applies them to Test.

Day 2 | Handling Categorical Data

> **Readings:** **[Géron]** Ch 2, Section "Handling Text and Categorical Attributes".
> **[ISLP]** Ch 3.3.1 "Qualitative Predictors".

Task:

1. Use `OneHotEncoder(handle_unknown='ignore')`.
2. Contrast this with `pandas.get_dummies()` (which fails in production pipelines).

Day 3 | Feature Scaling

> **Readings:** **[Géron]** Ch 2, Section "Feature Scaling".
> **[ISLP]** Ch 6.2 "The Lasso" (Scaling for Regularization).

Task:

1. Implement `StandardScaler` (Z-score).
2. Observe the effect on coefficients in a linear model (magnitude interpretation).

Day 4-5 | The ColumnTransformer

> **Readings:** **[Géron]** Ch 2, Section "Transformation Pipelines".

Task:

1. Build the "Production Template":
   - **Numeric Pipe:** Imputer → Scaler.
   - **Categorical Pipe:** Imputer → OneHotEncoder.
2. Combine them using `ColumnTransformer`.

2 Week 2: Plugging in the Models

Focus: Tree-based Ensembles (The Econometrician's Non-Parametric Tool).

Day 1 | Decision Trees

> **Readings: [ISLP]** Ch 8.1 "The Basics of Decision Trees".
> **[Géron]** Ch 6 "Decision Trees".

Task:

1. Attach `DecisionTreeRegressor` to your pipeline.
2. Visualize the tree using `export_graphviz`.
3. Tune `min_samples_leaf` to control variance.

Day 2 | Random Forests

> **Readings: [ISLP]** Ch 8.2.2 "Random Forests".
> **[Géron]** Ch 7 "Ensemble Learning".

Task:

1. Implement `RandomForestRegressor`.
2. Extract `feature_importances_` to understand what drives the model.

Day 3-4 | Gradient Boosting

> **Readings: [ISLP]** Ch 8.2.3 "Boosting".
> **[Géron]** Ch 7 "Gradient Boosting" and "XGBoost".

Task:

1. Implement `GradientBoostingRegressor`.
2. Understand the "Learning Rate" (λ) vs. "Number of Trees".

Day 5 | Grid Search (Tuning)

> **Readings: [Géron]** Ch 2 "Fine-Tune Your Model".
> **[ISLP]** Ch 5.1 "Cross-Validation".

Task:

1. Wrap the pipeline in `GridSearchCV`.
2. Tune preprocessing params (imputation strategy) AND model params (trees) simultaneously.

3 Week 3: Capstone (Rigorous Double Machine Learning)

Focus: Causal Inference with Uncertainty Quantification.

Day 1 | The DML Pipeline Architecture

Statistician's Note: Standard ML regularization introduces bias. We use the FWL Theorem to orthogonalize D and Y against X.

Task: Define two separate pipelines:

- `pipe_y`: Predict Outcome Y (Gradient Boosting).
- `pipe_d`: Predict Treatment D (Gradient Boosting).

Day 2 | Cross-Fitting & Residualization

Task: Write a K-Fold loop (e.g., 5 splits). For each fold:

1. Train both pipes on **In-Fold** data.
2. Predict on **Out-of-Fold** data.
3. Store residuals: $\tilde{Y} = Y - \hat{Y}$ and $\tilde{D} = D - \hat{D}$.

Day 3 | Validation: The Orthogonality Check

Statistician's Note: If your ML models worked, the residuals $\tilde{D}$ should contain NO information about confounders X.

Task:

1. Run a check regression: $\tilde{D} \sim X$ (all confounders).
2. Calculate R^2. It should be very close to 0.
3. If $R^2 > 0.05$, your ML model is underfitting the confounders. Tune it.

Day 4 | Inference: The Bootstrap

Readings: **[ISLP]** Ch 5.2 "The Bootstrap".

Task:

1. Standard errors from the final OLS step are slightly biased because they ignore the ML uncertainty.
2. Implement a Bootstrap loop (100 iterations):
   - Resample the original data with replacement.
   - Run the full DML process.
   - Store $\hat{\theta}$.
3. Calculate the SD of the stored $\hat{\theta}$s. This is your robust Standard Error.

Day 5 | Visualization & Reporting

Task:

1. **Partial Regression Plot:** Scatter plot $\tilde{Y}$ vs $\tilde{D}$. Add a regression line. This visualizes the causal effect "net of confounders."
2. Compare your DML estimate to a naive OLS ($Y \sim D$). Document the difference (the magnitude of the selection bias you fixed).