MS-Level Comprehensive Syllabus: Dual-Track (ISLP + HOML)
Dec 27, 2025 to Jan 31, 2026 (Asia/Manila)

Methods Mastery (ISLP) + Production Engineering (HOML)

Compilation instruction

If you want the Noto Sans look, compile with **XeLaTeX** or **LuaLaTeX**. If you compile with **pdfLaTeX**, the document will still compile using a fallback font.

Contents

1 Purpose

Primary goal: Master statistical learning methods using **ISLP (James et al.)** for theory and evaluation, synchronized with **HOML (Geron)** for modern engineering pipelines, Scikit-Learn internals, and production-ready code structure.

Secondary goal (optional): Apply selected labs to the SBFP table as practice only. No attendance outcomes are used.

2 How to use this syllabus

Non-negotiable operating rules

1. **Reproducibility first:** Run notebooks from a clean kernel.
2. **No leakage:** Preprocessing stays inside the cross-validation loop using `Pipeline`.
3. **Tuning is not evaluation:** Tuning happens on CV within the training split; final performance is estimated separately.
4. **Dual-Track Study:**
   - Read **ISLP** for the "Why" (Statistical justification, Bias-Variance).
   - Read **HOML** for the "How" (Classes, Transformers, APIs).

Daily workflow card

1. **Theory (ISLP):** Read assigned sections. Focus on: *What are the assumptions? When does this fail?*
2. **Engineering (HOML):** Read assigned sections. Focus on: *Which sklearn class do I use? What arguments matter?*
3. **Code:** Run the ISLP lab (reproduce outputs) AND snippets from HOML (reproduce pipeline logic).
4. **Check:** Update `concept_checks.md` (solve/fail/knobs/diagnostics).

3 Definition of Done

3.1 Minimum deliverables

- `islp_labs/`: One runnable notebook per completed topic.
- `homl_practice/`: Snippets showing you can implement the "Engineering" side (e.g., custom transformers).
- `concept_checks.md`: Four headings per chapter:
  1. What problem this method solves (Stat view)
  2. How to implement it robustly (Eng view)
  3. When it fails and why
  4. What knobs matter (hyperparameters)

4 Weekly Detailed Outlines

Week 0 (Dec 27–31): Foundations + Pipelines (The Setup)

Theme: The "Life Cycle" of a Model.
Specific Learning Objectives (What to know exactly):

- **ISLP Theory (Ch 2):**
  - Distinguish **Parametric** vs **Non-parametric** methods.
  - Define **Irreducible Error** and why we can never eliminate it.
  - Explain the **Bias-Variance Tradeoff** (The most important concept in the course).
  - Why **Mean Squared Error (MSE)** is used for regression.
- **HOML Engineering (Ch 2):**
  - Build a custom **Transformer** (inheriting from 'BaseEstimator', 'TransformerMixin').
  - Master the `Pipeline` class to chain preprocessing and modeling.
  - Use `ColumnTransformer` to treat numerical and categorical data differently.
  - Understand why we **never** call '.fit()' on the test set.

Readings:

- ISLP Ch 2 (Stat Learning).
- HOML Ch 2 (End-to-End Project) – *Read this twice. It is the blueprint.*

Build Artifact:

- A 'clean$_p$*ipeline.ipynb*'*thattakesarawCSV*, *cleansit*, *scalesit*, *andfitsadummymodelusing*'*sklearn.pipeline.*

Week 1 (Jan 1–7): Resampling, Regularization, Optimization

Theme: How to select the right model and find the best parameters.
Specific Learning Objectives:

- **ISLP Theory (Ch 5 & 6):**
  - **Cross-Validation (CV):** Why K-Fold is better than a single Split. LOOCV vs K-Fold variance.
  - **The Bootstrap:** What it is used for (estimating standard errors, not prediction).
  - **Ridge (L2) vs Lasso (L1):** Geometric interpretation. Why Lasso yields sparse models (feature selection) and Ridge shrinks coefficients.
  - **Tuning:** Why we need a "Validation" set to pick λ (alpha).
- **HOML Engineering (Ch 4):**
  - **Gradient Descent (SGD):** The difference between 'LinearRegression' (Normal Equation) and 'SGDRegressor' (Iterative).
  - **Learning Curves:** Plotting training error vs validation error to diagnose overfitting/underfitting.
  - **Regularization in Scikit:** Using 'Ridge', 'Lasso', and 'ElasticNet' classes.
  - **Early Stopping:** How to stop training when validation error spikes.

Readings:

- ISLP Ch 5 (Resampling), Ch 6 (Regularization).
- HOML Ch 4 (Training Models).

Build Artifact:

- A notebook comparing ‘Ridge‘ (Analytical) vs ‘SGDRegressor(penalty='l2')‘. Show they reach similar coefficients.

Week 2 (Jan 8–14): Classification & Advanced Metrics

Theme: Handling "Yes/No" problems and Imbalanced Data.
Specific Learning Objectives:

- **ISLP Theory (Ch 4):**
  - **Logistic Regression:** Log-odds interpretation. Why Linear Regression fails for probabilities.
  - **LDA/QDA:** The Bayes Theorem approach. Assumption of Gaussian distributions.
  - **KNN:** The non-parametric alternative. The "Curse of Dimensionality."
- **HOML Engineering (Ch 3):**
  - **Confusion Matrix:** True Positives, False Positives, FN, TN.
  - **Precision vs Recall:** The tradeoff. When to care about one over the other.
  - **F1 Score:** The harmonic mean.
  - **ROC Curve vs PR Curve:** Rule of thumb: use PR curve when the positive class is rare (e.g., fraud, disease).
  - **Implementation:** Using ‘cross$_val_predict$‘to get" clean" predictions for the confusion matrix.

Readings:

- ISLP Ch 4 (Classification).
- HOML Ch 3 (Classification).

Build Artifact:

- A ‘classification$_p$laybook.ipynb‘that plots the ROC curve and calculates the AUC for a Logistic Regression model.

Week 3 (Jan 15–21): Beyond Linearity + SVMs

Theme: Bending the lines (flexibility) and Geometric Margins.
Specific Learning Objectives:

- **ISLP Theory (Ch 7 & 9):**
  - **Polynomial Regression:** The simplest non-linear approach.
  - **Step Functions Splines:** Local regression (knots). Why Splines are smoother than polynomials.
  - **SVM:** The "Maximal Margin Classifier." Support Vectors. The "Kernel Trick" (mapping to higher dimensions).
- **HOML Engineering (Ch 5):**
  - **Polynomial Features:** Using ‘PolynomialFeatures‘ inside a Pipeline *before* a linear model.
  - **SVM Scaling:** Why SVMs **must** have scaled features (StandardScaler).
  - **Kernels in Sklearn:** ‘SVC(kernel='rbf')‘ vs ‘SVC(kernel='poly')‘. The ‘gamma‘ and ‘C‘ hyperparameters.

Readings:

- ISLP Ch 7 (Non-linear), Ch 9 (SVM).
- HOML Ch 5 (SVM).

Build Artifact:

- A notebook visualizing the decision boundary of an RBF Kernel SVM with different 'gamma' values.

Week 4 (Jan 22–28): Trees and Ensembles

Theme: The Power of the Crowd (Bagging Boosting).

Specific Learning Objectives:

- **ISLP Theory (Ch 8):**
  - **Trees:** Recursive Binary Splitting. Pruning.
  - **Bagging:** Bootstrap Aggregation. Reduces Variance. (Parallel learning).
  - **Random Forests:** Decorrelating the trees by splitting on a subset of features.
  - **Boosting:** Learning from mistakes. Reduces Bias. (Sequential learning).
- **HOML Engineering (Ch 6 & 7):**
  - **Visualizing Trees:** Using 'export$_{g}raphviz$'.
  - **Feature Importance:** Extracting 'feature$_{i}mportances$' $fromaRandomForest.$
  - **Gradient Boosting:** Using 'GradientBoostingRegressor' or 'HistGradientBoostingRegressor' (optimized for speed).
  - **Voting Classifiers:** Combining different models (LogReg + SVM + RF) into a 'VotingClassifier'.

Readings:

- ISLP Ch 8 (Trees).
- HOML Ch 6 (Decision Trees), Ch 7 (Ensemble Learning).

Build Artifact:

- An 'ensemble$_{b}akeoff.ipynb$' $comparingatunedRandomForestvsatunedGradientBoostingmodel.$

Week 5 (Jan 29–31): Deep Learning Sprint (PyTorch Focus)

Theme: Neural Networks and Unsupervised Learning.

Specific Learning Objectives:

- **ISLP Theory (Ch 10):**
  - **Neural Networks:** Layers, Neurons, Activation Functions (ReLU).
  - **CNN/RNN:** Brief overview of architectures for images/sequences.
  - **Unsupervised:** PCA (Variance maximization) vs Clustering (K-Means).
- **HOML Engineering (Ch 10 & 8):**
  - **The Perceptron:** Understanding the simplest unit.
  - **Backpropagation:** The intuition of how gradients flow backward.
  - **Dimensionality Reduction:** Using 'PCA' in Scikit-Learn to reduce features while keeping 95% variance.
- **PyTorch Specifics (ISLP Labs):**
  - Tensors vs Arrays.

  - The ‘.forward()‘ method.
  - Defining a ‘Sequential‘ model in torch.

Readings:

- ISLP Ch 10 (Deep Learning) - **Primary for Code (PyTorch)**.
- HOML Ch 10 (ANNs) - **Primary for Concepts**.
- HOML Ch 8 (Dim Reduction).

Build Artifact:

- **Capstone Memo:** Final 1-page summary of your best model from the previous weeks.

5 Required resources

- ISLP labs hub: `https://intro-stat-learning.github.io/ISLP/labs.html`
- HOML 3rd Ed Github: `https://github.com/ageron/handson-ml3`
- **PyTorch Note:** Use the ISLP labs for PyTorch code. Use HOML text for understanding Neural Network architecture.