MA477: Data Science

Lesson 4 Outline — 15 January 2026

United States Military Academy, West Point
Instructor: MAJ Patrick Kuiper

1 Administrative

- Quiz notes
- Reading will be given based on section headers not page numbers from now on

2 Student Review

3 Lesson Objectives

By the end of this lesson, cadets will be able to:

- Understand the steps in completing an end-to-end machine learning project.
- Understand how to implement data pipelines, clean/standardize data, visualize data, how to handle categorical/text
- Understand and know how to perform cross-validation

4 One-Hot Encoding Example

One-hot encoding + real-valued feature

Let $C \in \{\text{Red}, \text{Blue}, \text{Green}\}$ and define the one-hot vector

$$(c_{\text{red}}, c_{\text{blue}}, c_{\text{green}}) \in \{0,1\}^3, \qquad c_{\text{red}} + c_{\text{blue}} + c_{\text{green}} = 1.$$

Let $x \in \mathbb{R}$ be a real-valued feature. Define

$$\mathbf{x} = (x, c_{\text{red}}, c_{\text{blue}}, c_{\text{green}}), \qquad \mathbf{w} = (w_x, w_{\text{red}}, w_{\text{blue}}, w_{\text{green}}).$$

Then the linear model prediction is

$$\hat{y} = \mathbf{w}^\top \mathbf{x} = w_x x + w_{\text{red}} c_{\text{red}} + w_{\text{blue}} c_{\text{blue}} + w_{\text{green}} c_{\text{green}}.$$

If $C = \text{Blue}$, then $(c_{\text{red}}, c_{\text{blue}}, c_{\text{green}}) = (0,1,0)$ and

$$\hat{y} = w_x x + w_{\text{blue}}.$$

Step	Student checklist for a prediction project
1	**Clarify the question:** Define the target, task type (regression vs. classification), and what "good" performance means in context.
2	**EDA (Exploratory Data Analysis):** Inspect data types, summary statistics, missingness, outliers, class balance, and simple plots/relationships to understand what you have.
3	**Baseline + preprocessing:** Build a simple baseline model; decide if features need **scaling/standardization** (especially for distance/gradient-based models) and encode categorical variables as needed.
4	**Evaluation + regularization:** Choose an appropriate metric/loss; use a validation plan (train/test split or cross-validation); apply **regularization** (e.g., L1/L2) and tune hyperparameters to reduce overfitting.
5	**Overfitting check:** Compare training vs. validation performance; look for instability across folds/splits; simplify model or increase regularization if needed.
6	**Interpret + sanity-check:** Confirm predictions make sense; check feature effects; verify there is no data leakage; summarize limitations and next steps.

K-Fold Cross-Validation (CV)

Idea. Cross-validation estimates how well a model will generalize to new data. Instead of training once on a single train/test split, we repeatedly train and validate on different parts of the data.

Procedure (K-fold CV)

Assume we have n labeled examples and choose an integer K (often 5 or 10).

1. Randomly shuffle the dataset (optional but common).
2. Split the data into K **folds** of (roughly) equal size.
3. For each round $i = 1, 2, \ldots, K$:
   - Use fold i as the **validation set**.
   - Use the other $K - 1$ folds as the **training set**.
   - Train the model on the training set and compute a validation score on fold i.
4. Average the K validation scores to get the cross-validation estimate:

$$\text{CV score} = \frac{1}{K} \sum_{i=1}^{K} s_i,$$

   where s_i is the chosen metric (accuracy, MSE, etc.) on validation fold i.

Why it helps. Each example is used for validation exactly once and for training $K - 1$ times, so the estimate is typically more stable than a single split.

A simple visual (5-fold CV)

Legend: **T** = training fold, **V** = validation fold

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Round 1:	V	T	T	T	T
Round 2:	T	V	T	T	T
Round 3:	T	T	V	T	T
Round 4:	T	T	T	V	T
Round 5:	T	T	T	T	V

Common variations

- **Stratified K-fold:** keeps class proportions similar in each fold (important for classification).
- **Repeated K-fold:** repeats K-fold CV multiple times with different shuffles.
- **Leave-one-out (LOOCV):** $K = n$ (each fold has one example); can be expensive.