Master of Science in Business Analytics

Machine Learning I
OPAN 6602

Week 4 Live Session

Instructor:
Tommy Jones, PhD

GEORGETOWN UNIVERSITY / McDONOUGH SCHOOL of BUSINESS

“All models are wrong. Some Models are useful.”

– George Box

“An approximate answer to the right question is worth far more than a precise answer to the wrong one.”

– John Tukey

Bottom Line Up Front

We are covering 4 topics

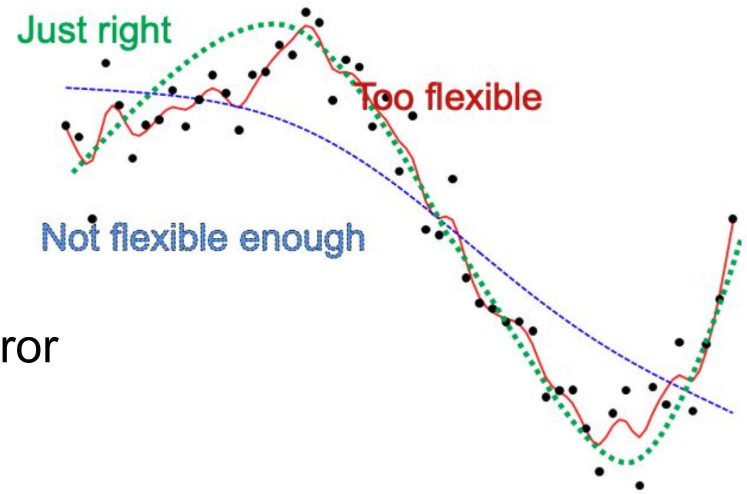
1. The bias-variance tradeoff in machine learning models (conceptual, but very important)
2. K-fold cross validation
3. Imputation of missing values
4. More feature engineering
(Emphasis: standardization and normalization)

The Bias Variance Tradeoff

$$E(Y_0 - f(X_0))^2 =$$

$$\left.\begin{array}{l} \mathrm{Var}(f(X_0)) + \\ \\ [\mathrm{Bias}(f(X_0))]^2 + \end{array}\right\} \text{Reducible error}$$

$$\mathrm{Var}(\epsilon) \quad \text{Irreducible error}$$



Good fit/ robust
As model becomes less complex, bias increases (systematically miss signals)
As model becomes more complex, variance increases in our estimates

Why do we care?

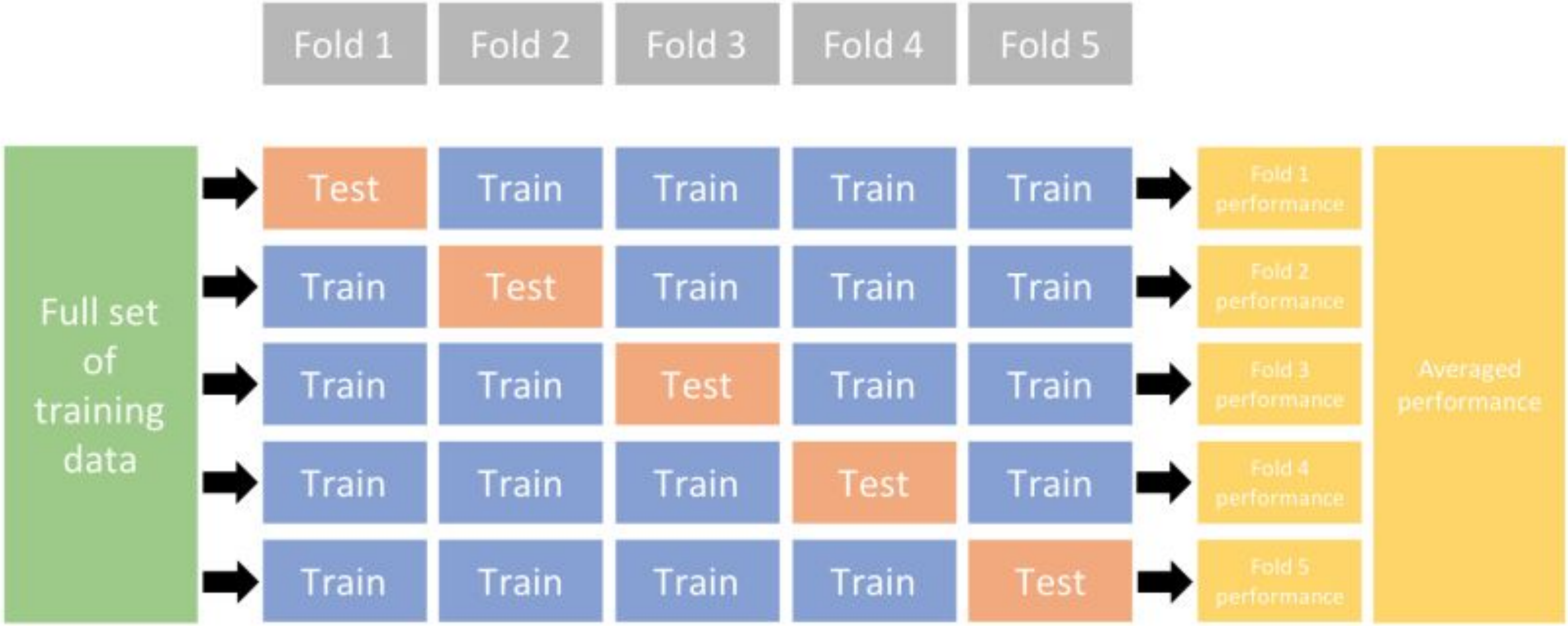
Many modeling choices involve choosing between this tradeoff.

Informs why certain models behave as they do.

K-Fold Cross Validation

Method that randomly divides the training data into k groups (aka folds) of approximately equal size. The model is fit on k−1 folds and then the remaining fold is used to compute model performance.

Typically, one chooses K = 5 or k = 10.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
Full set of training data	Test	Train	Train	Train	Train	Fold 1 performance	Averaged performance
	Train	Test	Train	Train	Train	Fold 2 performance	
	Train	Train	Test	Train	Train	Fold 3 performance	
	Train	Train	Train	Test	Train	Fold 4 performance	
	Train	Train	Train	Train	Test	Fold 5 performance	

Graphic by Sudipta Dasmohapatra

Many uses for K-Fold CV, but…

Use primarily on your training set to

- compare models,
- hyperparameter choices,
- design decisions,
- etc.

Imputation: Filling in Missing Data (Quick Overview)

- **Always fit imputation on the training set only** — then apply those same parameters (mean, median, neighbors, etc.) to the test set to avoid data leakage.

- **Mean, median, or most-frequent imputation** is simple, fast, and often “good enough,” but **can shrink variability and bias relationships**.

- **Model-based imputation** (e.g., KNNImputer in scikit-learn) uses patterns across other variables to predict missing values — **often more accurate, but slower**.

- **Imputation is about making the dataset usable, not perfect** — the key is consistency and **avoiding leakage from future information**.

Scaling

Standardization

Re-scale a variable, *X*, so its units are measured in **standard deviations from its mean**.

$$x_i^{(new)} = \frac{x_i - E[X]}{\sqrt{V[X]}}$$

Min/Max Scaling (Normalization)

Re-scale a variable, X, so its **range is between 0 and 1**.

$$x_i^{(new)} = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)}$$

Binning

Change a continuous variable into a discrete one. i.e., Turn a numeric variable into an ordered categorical variable.

$$x_i^{(new)} = \begin{cases} A, & x_i \ge a \\ B, & a < x_i < c \\ C, & x_i \le c \end{cases}$$

Where you cut and how many bins can significantly affect your results.

Transforming

Generally, transform your variable using a mathematical function.

Scaling, Binning, and Encoding are subsets of transforming.

Simple common transforms

Log transform $x_i^{(new)} = log(x_i)$

Square root transform $x_i^{(new)} = \sqrt{x_i}$

Cubed root transform $x_i^{(new)} = x_i^{1/3}$

Encoding

Many names: Dummy / Binary / Indicator / “One Hot Encoding”

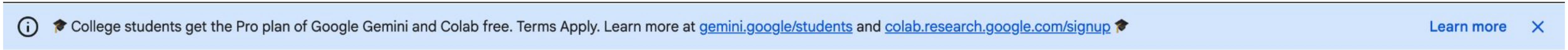
Turns categorical data into numeric data.

X
Apple
Banana
Pear

→

Apple	Banana	Pear
1	0	0
0	1	0
0	0	1

Free Google Colab Pro for Students (and Teachers)



🎓 College students get the Pro plan of Google Gemini and Colab free. Terms Apply. Learn more at gemini.google/students and colab.research.google.com/signup 🎓

See: https://colab.research.google.com/signup

End