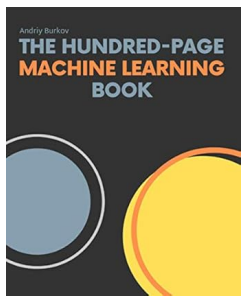
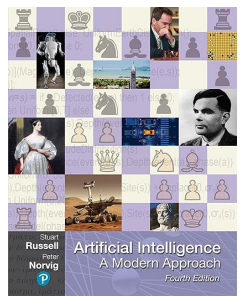
Lesson 02.6: ML Techniques - How to Do Research

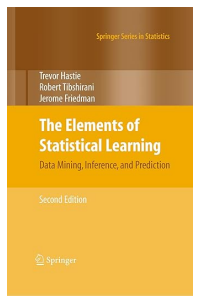
Instructor: Dr. GP Saggese - gsaggese@umd.edu

References:

- Burkov: *"The Hundred-Page Machine Learning Book"* (2019)
- Russell et al.: *"Artificial Intelligence: A Modern Approach"* (4th ed, 2020)
- Hastie et al.: *"The Elements of Statistical Learning"* (2nd ed, 2009)







- ***How to Do Research***

Occam's Razor

- **Simple is better**
  - *"The simplest model that fits the data is also the most plausible"* (Occam)
  - *"An explanation of the data should be as simple as possible, but not simpler"* (Einstein?)
- Corollary:
  - Trim the model to the bare minimum necessary to explain the data
- What does **simple** mean?
  - A model is simple when it is one of few possible objects
- What does **better** mean?
  - A model is better means better out of sample performance
- **Why it's true**?
  - Less likely to fit a given data by coincidence
  - An event is more significant if it is unlikely (formalized in terms entropy)

What is a "simple model"?

- An object is **simple** when it is one of few possible objects
- **Examples**:
  - Polynomial of order 2 is simpler than order 17
    - More polynomials of order 17 than order 2, though both are infinite sets
  - SVM (Support Vector Machine) characteristics:
    - Separating hyperplane appears wiggly, but defined by few support vectors
- **Measures of complexity**
  - Complexity of a single hypothesis h
    - E.g., polynomial order, MDL (describe hypothesis in bits), Kolmogorov complexity
  - Complexity of a hypothesis set $\mathcal{H}$
    - E.g., VC dimension of the model
  - Complexity of h and $\mathcal{H}$ related by counting:
    - If you need l bits to specify h, then h is one of 2^l elements of set $\mathcal{H}$

Model Soundness

- **Model should tell a story**
  - You cannot blindly accept results of modeling
  - Ask: *"What criticisms would you give if the model was presented for the first time?"*
- **Benchmark models**: compute performance if model outputs:
  - Always 0 or 1
    - E.g., long-only model for stock predictions
  - Random results
    - E.g., bootstrap of null hypothesis "no prediction power"
- **Perfect fit can mean nothing**, e.g.,
  - Get 2 data points on a plane
    - Fit data with a linear relationship
    - Perfect fit
  - Means nothing since:
    - Always a line between 2 points
    - Data cannot falsify hypothesis
  - The model (line) is too complex for data set (only 2 points)

Sampling Bias (1/2)

- **Sampling bias** occurs when data is not representative of the intended population
  - Biased sampling leads to biased outcomes
  - Model views world through training data
  - Hoeffding's hypothesis: training and testing distributions are the same
- **Causes of Sampling Bias**
  - *Non-random sampling*: Favors certain outcomes or groups
  - *Undercoverage*: Inadequate representation of some population members
  - *Survivorship bias*: Includes only successful subjects
  - *Self-selection bias*: Participants choose to participate, introducing bias
- **Consequences**
  - Systematic errors distort relationships or predictions
  - Models trained on biased samples perform poorly on real-world data
  - Leads to incorrect or unfair decisions

Sampling Bias (2/2)

- **Examples**
  - Training facial recognition on lighter-skinned faces reduces accuracy on darker-skinned faces
  - Predicting income using data only from employed individuals omits unemployed people
- **Mitigation**
  - Compare sample statistics with population statistics
  - Use stratified sampling or resampling to balance data
  - Apply re-weighting or bias correction techniques
  - If data points have zero probability ($\Pr = 0$), no remedies possible

Data Snooping (1/2)

- **Data snooping** occurs when test set information improperly influences model-building
- **Typical Sources**
  - *Train-test contamination*: Test data leaks into training
    - E.g., Using test data during feature engineering, model selection, or tuning
  - *Multiple hypothesis testing*: Trying many models inflates performance
- **Consequences**
  - Overly optimistic evaluations
  - Models perform well in validation but fail on unseen data
- **Why It's Dangerous**
  - Misleading results
  - False confidence in model quality
  - Common pitfall in machine learning workflows

Data Snooping (2/2)

- **Examples**
  - Choosing features based on the full dataset before splitting gives foresight not present in deployment
  - E.g., demean the entire data and then split in train / test
- **Preventive Strategies**
  - Keep training, validation, and test sets separate
  - Use cross-validation properly
  - Fit preprocessing steps only on training data, then apply to test data

"Burning the Test Set"

- **Problem**
  - Repeated test set use during development leaks into training
  - Overfitting to test data
  - *"If you torture the data long enough, it will confess whatever you want"* (Coase, 1982)
- **Symptoms**
  - Test accuracy improves; real-world performance stagnates or degrades
  - Model evaluation becomes untrustworthy
- **Solutions**
  - *One-time use principle*: Evaluate on test set once, post-tuning
  - *Use a validation set*: For model development and tuning
  - *Statistical adjustments*:
    - Adjust p-values for multiple testing
    - Use Bonferroni or False Discovery Rate corrections
  - *Theoretical tools*:
    - VC dimension reflects learning capacity, including repeated trials
    - Minimum Description Length (MDL) penalizes complex models and repeated fitting

How to Achieve Out-Of-Sample Fit

- **Goal**: Choose an hypothesis $g \in \mathcal{H}$ approximates the unknown target function f
  - What does $g \approx f$ mean?

$$g \approx f \iff E_{out}(g) \approx 0$$

- **Solution**:
  - Achieve:
    - Good in-sample performance $E_{in}(g) \approx 0$
    - Good generalization $E_{out}(g) \approx E_{in}(g)$
  - Together $\implies$ good out-of-sample performance $E_{out}(g) \approx 0$

What If Out-Of-Sample Fit Is Poor?

- The model performs well in sample ($E_{in} \approx 0$) but poorly out of sample ($E_{out} \gg E_{in}$)
- **What does it mean?**
  - In-sample performance is optimistic
  - Model is overfitted and fails to generalize
- **What do you do?**
  - Run diagnostics before long-term projects
  - Gain insight on what works/doesn't to improve performance
    - E.g., bias-variance curves and learning curves
- **How to fix?**
  1. Training data
     - Get more training data $\iff$ fixes high variance
  2. Features
     - Remove features $\iff$ fixes high variance
     - Add features $\iff$ fixes high bias
     - Add derived features (e.g., polynomial features) $\iff$ fixes high bias
  3. Regularization
     - Decrease regularization λ $\iff$ fixes high bias
     - Increase regularization λ $\iff$ fixes high variance

Why Using a Lot of Data?

- **Several studies** show that:
  - Different algorithms/models have remarkably similar performance
  - Increasing training set improves performance
- **Consequence**
  - High capacity model + massive training set = good performance
- **Why**
  - Using a high capacity model with many parameters (e.g., neural network)

$$E_{in} \approx 0$$

    due to low bias (and high variance)
  - A massive data set helps avoid overfitting

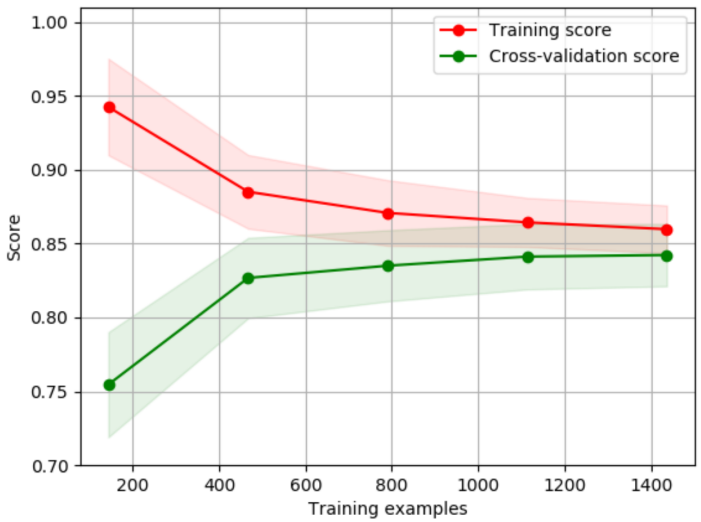
$$E_{out} \approx E_{in}$$

  - These two conditions together

$$E_{out} \approx E_{in} \approx 0 \implies E_{out} \approx 0$$

What to Do When You Have Lots of Data?

- You have $m = 100$M examples in data set: great!
- **Cons**
  - Scalability issue (slow, lots of compute)
  - Require work on infrastructure
- **First step**:
  - Plot learning curves (in-sample and out-of-sample performance) for increasing amount of data $m = 1k, 10k, 100k, 1M, ...$
- **Next step**:
  - If model has large bias
    - Training and validation performance are similar at $1M$
    - Next step: use more complex model rather than training on $100M$ instances
  - If model has large variance
    - Next step: use all $100M$ instances to train the model



SCIENCE ACADEMY

Why We Do Things?

- **Always, always, always:**
  - Understand the purpose of the task
    - Ask: *"Why are we doing something?"*
  - Clarify goals and outcomes of the task
    - Ask: *"What do we hope to determine by performing the task?"*
  - Think about actions with the bigger picture in mind
  - Avoid going through the motions
  - Prioritize tasks by importance and impact
- E.g., when conducting a customer survey:
  - Q: *"Why is the feedback being collected?"*
    - A: To improve product features and customer service
  - Q: *"What is the desired outcome?"*
    - A: To identify areas for improvement and innovation
- E.g., before starting a marketing campaign:
  - Q: *"Why is this campaign run?"*
    - A: To increase brand awareness or drive sales
  - Q: *"What are the specific goals?"*
    - A: Set target number of new leads or click-through rates

Summary of the Results, Next Steps

- **Always have a summary of results**
  - High-level map of discoveries
    - E.g., "smoothing model coefficients helps"
  - Highlight major findings
  - Interpret results
    - E.g., *"Increase in sales likely due to new marketing strategy."*
  - Conclusions
    - Summarize data suggestions or confirmations
    - E.g., *"Hypothesis that user engagement increases retention is supported"*
- **Always have a reference to detailed results**
  - Provide quick insights before details
- **Always have next steps**
  - What do you expect to happen?
    - Like thinking n moves ahead in chess
  - E.g., *"Next, conduct detailed analysis on demographics contributing most to sales growth"*
    - Outline potential experiments or analyses to validate findings further

Example: Spam Filter Classification

- You use $N = 5$ words in an email to distinguish spam / non-spam emails using logistic regression
  - Words are `buy`, `now`, `deal`, `discount`, `<your name>`
- **Goal**: improve classifier performance
  1. **Collect more data**
     - Honeypot project: fake email account to collect spam
  2. **Use better features**
     - Email routing info (spammers use unusual accounts, mask emails as legitimate)
  3. **Use better features from message body**
  4. **Detect intentional misspellings**
     - Spammers use misspelled words (e.g., `w4tch` for `watch`)
     - Use stemming software

Right and Wrong Approach to Research

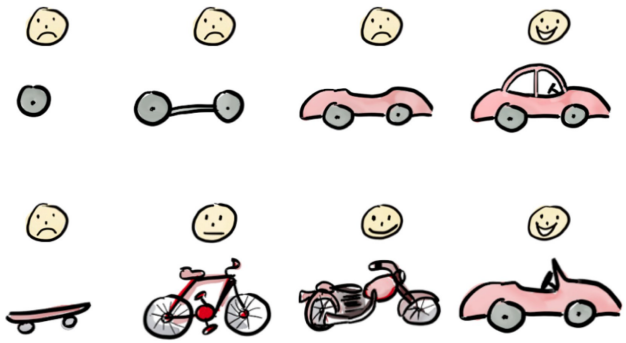
- It is not clear how to prioritize the different possible tasks
- **Bad**
  1. Use gut feeling to choose a task
  2. Complete task
  3. Re-evaluate performance
- **Good**
  1. Build a simple algorithm
     - Within 1 day
  2. Set up performance evaluation
     - Single number and bounds to improve
     - Use cross-validation
  3. Set up diagnostic tools
     - Compute learning and bias-variance curves
     - Understand issues before fixing (avoid premature optimization)
  4. Review misclassified emails manually
     - Identify features to improve performance
     - E.g., what are types of misclassified emails?
- Sometimes you just need to try approaches to see if they work
  - E.g., use stemming software to equate words

Incremental vs Iterative

- **Incremental Development**
  - Each increment adds functional components
  - Require upfront planning to divide features meaningfully
  - Integration of increments can be complex
- **Iterative Development**
  - Each increment delivers usable system
  - Refine and improve product through repeated cycles
  - Get feedback
  - Uncover and adjust for unknown requirements
- **Incremental $\gg$ Iterative**

_Incremental

Iterative



Incremental vs Iterative