CAI 4104/6108 – Machine Learning Engineering:

Performance Evaluation

Prof. Vincent Bindschaedler

Spring 2024

Administrivia: Midterm

- Reminder: Midterm is coming up!
  - Monday 2/19 and Wednesday 2/21 during class time (10:40 - 11:30) in **FLG 0220**
    - Topics: everything until 2/16
  - Duration: 50 minutes
  - Schedule:
    - **CAI6108**: take the exam on **2/19**
    - **CAI4104**: take the exam on **2/21**
    - Lecture that week (topic: Unsupervised Learning) will be pre-recorded
  - Format: **closed-books**, (blank) scratch paper, and physical calculator are allowed (no phones!)
  - Questions: short answers + problems

- Sample Midterm (Practice Questions):
  - On Canvas — 50 minutes (18 short answers) + 3 problems.
  - This is for you to practice. Do **not** overfit!

Administrivia: Homework 2

- **Homework 2** is out
  - Topic: regression and ensembles
  - Due **2/23** (delayed to avoid conflict with the midterm)
  - Do **not** submit the data file, only submit the notebook (.ipynb)
  - Advice: start early

Reminder: Training, Test, Validation

- Before training a model (or looking at the data ideally)
  - Divide the dataset into **three disjoint parts**
    1. Training dataset
    2. Test dataset
    3. Validation dataset
- Why? Why do we need the validation dataset? Why not just training and test?
  - We need it for hyperparameter optimization!
- What proportion of the dataset to allocate to each?
  - (Rule of thumb:) For small datasets (i.e., < 100k examples): 70% training, 15% validation, 15% test
  - For large datasets (e.g., deep learning): 95% training, 2.5% validation, 2.5% test
  - For very small datasets (e.g., <1000 examples): check the raw numbers (e.g., how many examples is 10%?)
  - What if you don't have enough data to afford leaving some aside for validation/testing?
    - Use k-fold validation: divide the data into k equal parts, then train on k-1, test on the remaining part, repeat k times and average!

Model Performance Evaluation

- What do we care about?
  - Out-of-sample predictions quality
  - In other words, we want our model to perform well on the test set
    - If it does, we say it **generalizes** well
  - What about overfitting?
- We need a metric
  - Different metrics for different tasks (e.g., classification vs regression, binary vs multiclass)
  - Q: Why don't we use the loss/cost function that we train the model with?

Metrics for Regression

- Mean Squared Error (MSE):
  - $\mathrm{MSE}(h) := 1/n \sum_i [h(\boldsymbol{x}_i) - y_i]^2$
  - Root Mean Squared Error (RMSE): $\mathrm{RMSE}(h) := [\mathrm{MSE}(h)]^{1/2}$
- Mean Absolute Error (MAE):
  - $\mathrm{MAE}(h) := 1/n \sum_i |h(\boldsymbol{x}_i) - y_i|$
- Median Absolute Error (MedAE):
  - $\mathrm{MedAE}(h) := \mathrm{median}(\{ |h(\boldsymbol{x}_i) - y_i| \}_{i=1,2,\ldots,n})$
- Almost Correct Predictions Error Rate (ACPER):
  - τ: threshold of percentage error that is acceptable (e.g., 1%)
  - $\mathrm{ACPER}(h, \tau) = 1/n \,|\, \{ \boldsymbol{x}_i : \ |h(\boldsymbol{x}_i) - y_i| \,/\, y_i \leq \tau) \}_{i=1,2,\ldots,n} \,|$
    - Proportion of predictions within τ % of the target

- What about baselines?
  - Mean model: always predict the mean value/target of the training data

Metrics for Classification

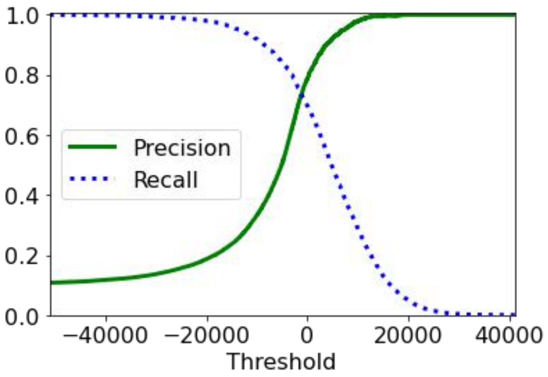
- Confusion Matrix
  - Applicable to classification in general
- Focus on the binary classification case:
  - One of the classes is designated as "*positive*" (the other is "*negative*")
  - False positives are Type I errors
    - Think of them as "false alarms"
  - False negatives are Type II errors
    - Think of them as "missed detections"
  - Prevalence (aka base rate):
    - Proportion of positive examples
- Baselines?
  - Statistical mode prediction
  - Random guessing

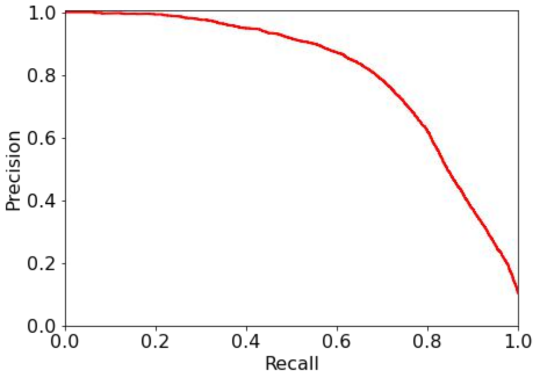
		Actual	
		+	-
Predicted	+	True Positive (TP)	False Positive (FP)
Predicted	-	False Negative (FN)	True Negative (TN)

- Accuracy: (TP + TN) / (TP + FP + TN + FN)
- Recall: TP / (TP + FN)
  - Also called: True positive rate (TPR), Sensitivity
- False negative rate (FNR): FN / (TP + FN) = 1 - TPR
- False positive rate (FPR): FP / (FP + TN)
- True negative rate (TNR): TN / (FP + TN) = 1 - FPR
  - Also called Specificity and Selectivity
- Precision: TP / (TP + FP)
  - Also called: Positive predictive value (PPV)
- False discovery rate (FDR): FP / (TP + FP) = 1 - Precision

Precision-Recall Tradeoff

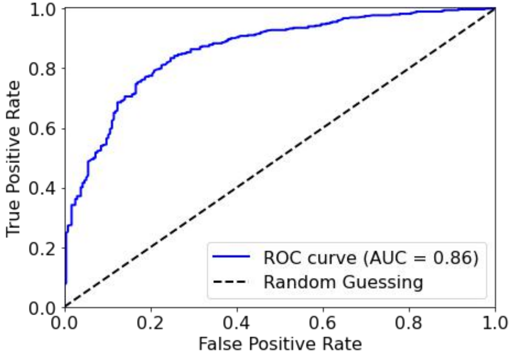
- Decision Threshold / Function
  - (In most cases) when you train a classification model, you actually train a family of classifiers!
    - The model assigns scores (or probabilities) to examples
    - Make predictions based on scores using a specific decision threshold
- The Tradeoff
  - The threshold provides a tradeoff between Type I and Type II errors
  - A popular way to express this tradeoff is Precision versus Recall
    - We need to choose between high precision and high recall
    - Q: What if we need to achieve precision above a specific value (e.g., 90%)?
- Optimizing and Statisficing
  - One way of navigating the tradeoff is to set a cutoff for precision or recall
  - E.g.: Pick the model with precision ≥95% that maximimizes recall



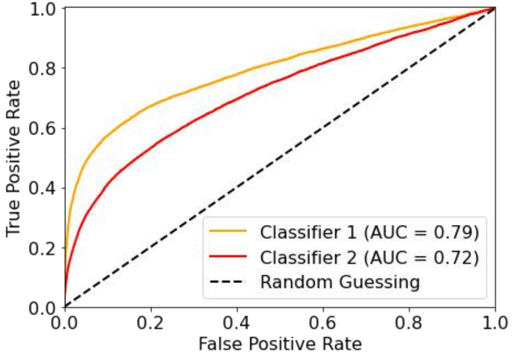


Performance Curves

- Receiver Operating Characteristics (ROC)
  - Plots true positive rate (TPR) versus false positive rate (FPR)
    - How? Vary the threshold
  - Each point on the curve (a valid pair (TPR,FPR)) is a valid tradeoff point
    - E.g.: Equal Error Rate (EER) — point where FPR and FNR are equal



- Area Under Curve (AUC)
  - This is exactly the area under the ROC curve
    - AUC = 0 means the worst possible classifier
    - AUC = 0.5 is a random classifier
    - AUC = 1.0 is a perfect classifier



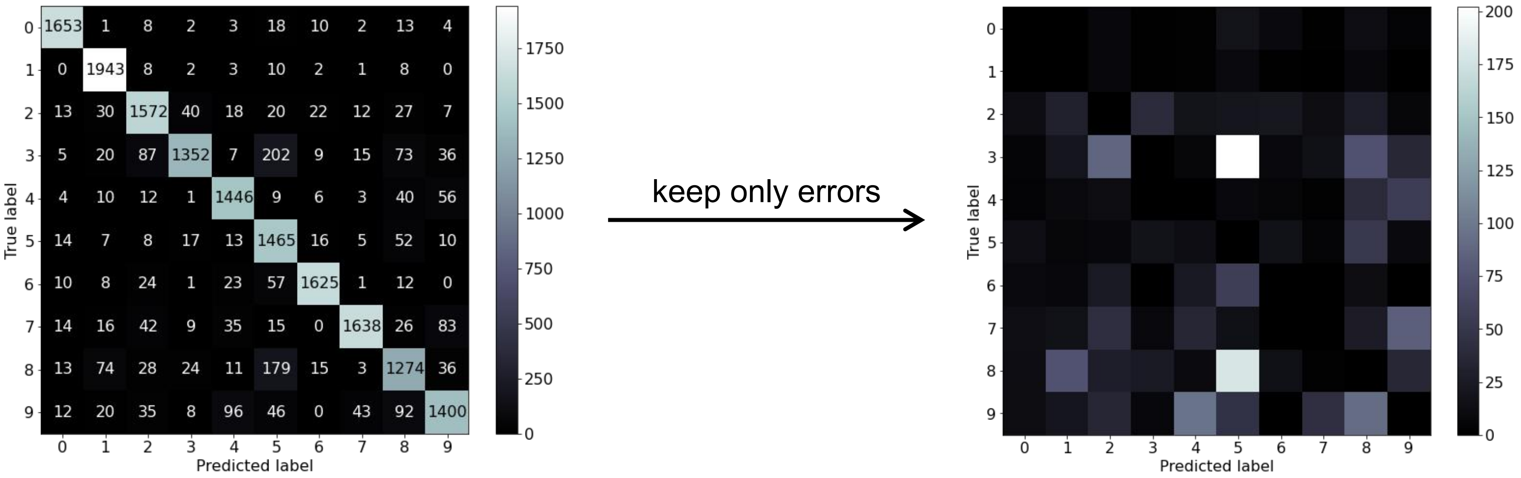
- Note: there are other performance curves
  - For example: Detection Error Tradeoff (DET) curves
    - False positive rate (FPR) versus the false negative rate (FNR), usually a log-log plot

Accuracies & F-measure

- Accuracy
  - Definition: the proportion of examples correctly classified (out of all examples)
    - If $\boldsymbol{M}$ is the confusion matrix, then $\mathrm{accuracy} = \mathrm{Tr}(\boldsymbol{M}) \,/\, \sum_{i,j} \boldsymbol{M}_{i,j}$
      - where $\mathrm{Tr}(\boldsymbol{M})$ is the trace of $\boldsymbol{M}$, i.e., $\mathrm{Tr}(\boldsymbol{M}) = \sum_i \boldsymbol{M}_{i,i}$ (the sum of diagonal elements)
  - Cost-sensitive accuracy: assign a cost to both FP and FN; multiply by those costs to compute accuracy
    - E.g.: (TP + TN) / (TP + α FP + TN + β FN), where α>0 is the cost for FP and β>0 is the cost for FN
  - Per-class accuracy: calculate accuracy for each class separately then average the individual accuracies
    - Suitable for multiclass problems; Useful when the dataset is imbalanced but all mistakes are costly
    - Warning: do **not** use this if some classes have very few examples! **Why?**
    - Example for the binary case: (balanced accuracy) BA = (TPR + TNR) / 2
- F-measure (aka F-score)
  - Combination of precision and recall into a single measure
  - $\mathrm{F}_\beta = (1 + \beta^2)\ \mathrm{precision} \cdot \mathrm{recall} \,/\, (\beta^2 \cdot \mathrm{precision} + \mathrm{recall}) = (1 + \beta^2)\ \mathrm{PPV} \cdot \mathrm{TPR} \,/\, (\mathrm{PPV} + \mathrm{TPR})$
  - F_1-score: $\mathrm{F}_1 = 2\ \mathrm{precision} \cdot \mathrm{recall} \,/\, (\mathrm{precision} + \mathrm{recall}) = 2\ \mathrm{TP} \,/\, (2\ \mathrm{TP} + \mathrm{FP} + \mathrm{FN})$

Multiclass Classification

- Can we use precision and recall?
  - Yes: designate one class as “positive” and the rest as “negative”
- What else can we do?
  - Accuracy and per-class accuracy
  - Error analysis: explore the confusion matrix to understand the kinds of error that your model makes!



Model Calibration

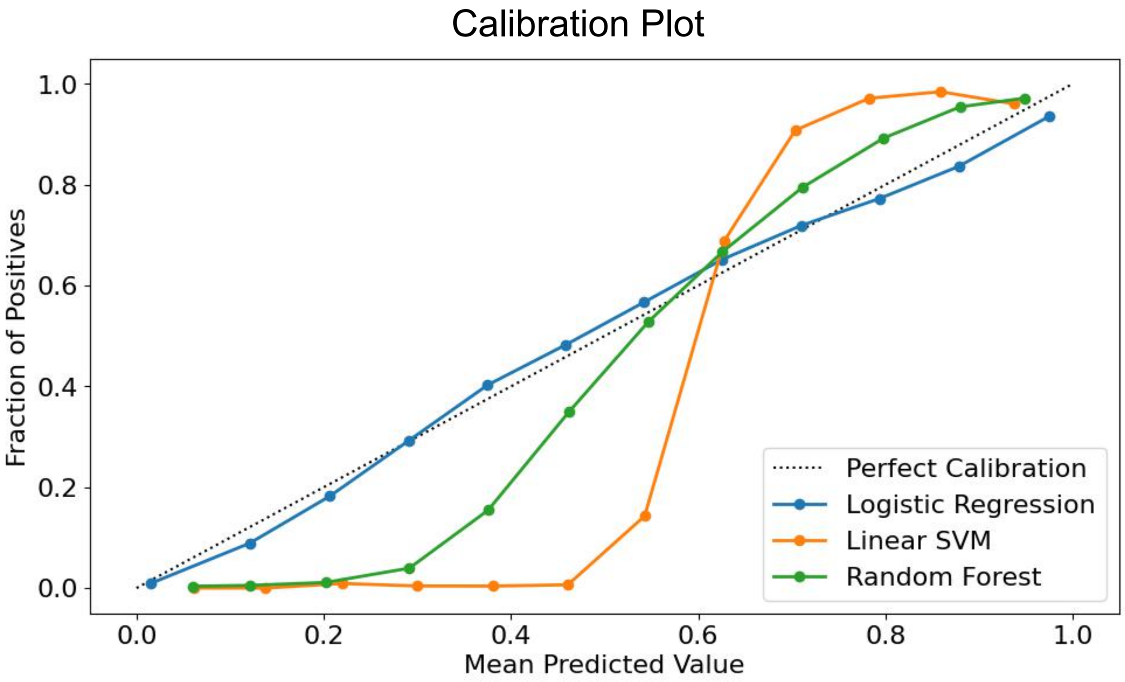
- Classification:
  - Given an input example $\boldsymbol{x}$, the model returns a prediction class label y'
    - But many models also return a **score** or a **probability** p associated with the prediction
  - A model is **well-calibrated** if we can ***interpret*** the score/probability as the **true probability**

- Most models are **not** well-calibrated by default
  - Especially deep neural networks but also SVM, etc.

- Calibration techniques:
  - Train a calibration model ("sigmoid scaling" aka "Platt scaling" or "isotonic regression")
  - Many regularization techniques help with calibration
    - For example: L_1 and L_2 regularization, label smoothing, temperature scaling, etc.
  - Data augmentation

Model Calibration

- A model is **well-calibrated** if we can ***interpret*** the score/probability as the **true probability**
- Is your model well-calibrated?



Base Rate Fallacy

- Example:
  - Suppose you have a very accurate classifier (e.g., 99% accurate) to predict whether a person suffers from a specific disease D from features of their blood
  - Q: Should we test everyone in the world? Why or why not?
    - It depends on the base rate!

- Base rate fallacy / base rate neglect
  - Error in reasoning: confusing a classifier's prior probability of correct prediction and the posterior probability of a true positive
  - Suppose we have a classifier that has a false positive rate of 2% (i.e., 2% of the time it predicts '+' when the true label is '-') and a true positive rate of 100% (i.e., it never fails to detect '+' instances)
  - What is the probability that if the classifier predicts '+' the true label is in fact '+'?
    - If the base rate is 0.5 (i.e., 50% of instances are '+') then it is: ~98%
    - If the base rate is 0.001 (i.e., 0.1% of instances are '+') then it is: **~4.8%**

Base Rate Fallacy

- Base rate fallacy / base rate neglect
  - Suppose we have a classifier that has a false positive rate of 2% (i.e., 2% of the time it predicts '+' when the true label is '-') and a true positive rate of 100% (i.e., it never fails to detect '+' instances)
  - What is the probability that if the classifier predicts '+' the true label is in fact '+'?
    - If the base rate is 0.5 (i.e., 50% of instances are '+') then it is: ~98%
    - If the base rate is 0.001 (i.e., 0.1% of instances are '+') then it is: ~4.8%

		Actual	
		+	-
Predicted	+	5000	100
	-	0	4900

Pr(Actual + | Predicted +) = 5000/5100 ≈ 0.98

		Actual	
		+	-
Predicted	+	10	200
	-	0	9790

Pr(Actual + | Predicted +) = 10/210 ≈ 0.0476

- Note: there are other (similar) errors in reasoning
  - Examples: Prosecutor's fallacy, Simpson's paradox, etc.

Next Time

- Wednesday (2/14): Lecture

- Upcoming:
  - Homework 2 is out (due 2/23) by 11:59pm
  - Midterm on 2/19 and 2/21