CS229: Evaluation Metrics

Jeremy Irvin
Slides by Anand Avati, Jeremy Irvin
Oct 20, 2023

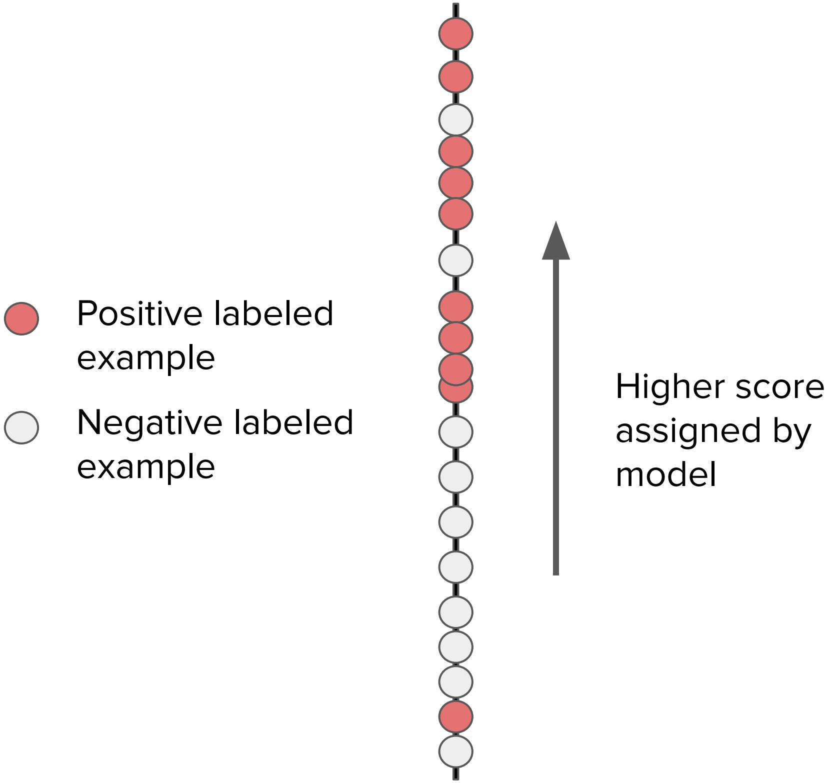
Why should we care about evaluation metrics?

- Quantitatively define a **real world objective**
  - Training objective (loss function) is typically only a *proxy* for this objective
  - Ideally this objective matches the real world objective as closely as possible
- Help organize ML team effort toward that target
  - Generally by trying to improve the metric on the *validation set*
- Quantify the gap between
  - Baseline and desired performance (initial difficulty estimate)
  - Current performance and desired performance
- Help **debug** (bias vs. variance)

Binary Classification Setting

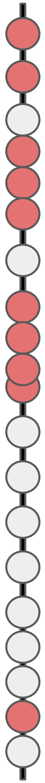
- Input ***X***, *binary* output $\boldsymbol{y \in \{0, 1\}}$
- Two types of models:
  - Models that output a *categorical class directly* (K Nearest neighbor, Decision tree)
  - Models that output a *real valued score* (Logistic Regression, SVM, NN)
    - Score could be margin (SVM) or a probability (LR, NN)
    - We'll focus on this type for now, although many of the metrics we'll discuss apply to the other type as well

Score-based Models



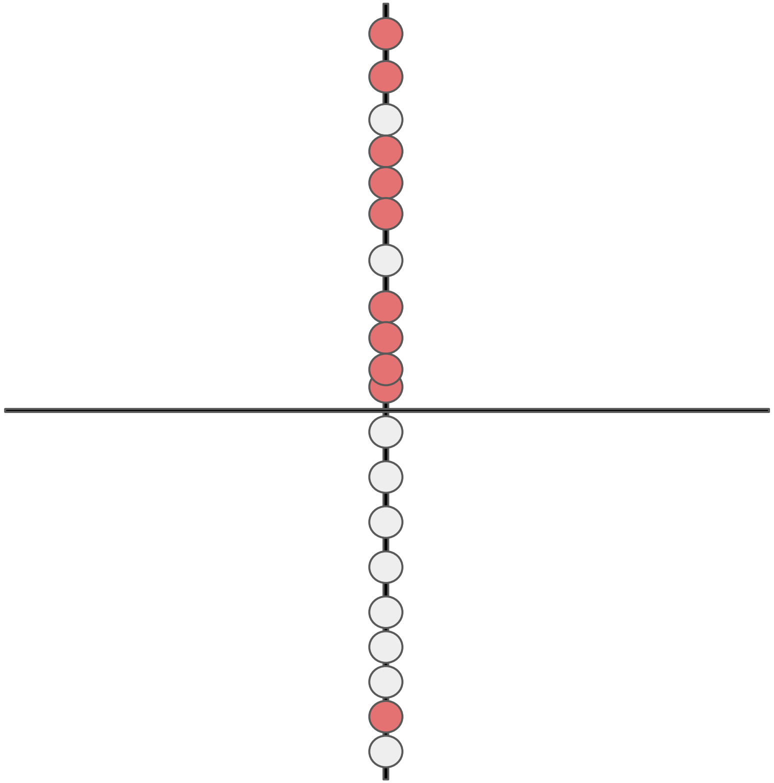
- Score output by a logistic regression model or SVM
- For many metrics, **only the order** of positive/negative examples matters
- *Prevalence* (# positives / # total) determines class imbalance
- If too many examples, can plot a histogram (binned by scores) instead
- Rank view helpful for error analysis (look at topmost negative or bottommost positive)

Classifier



- Need to pick a threshold to have a classifier

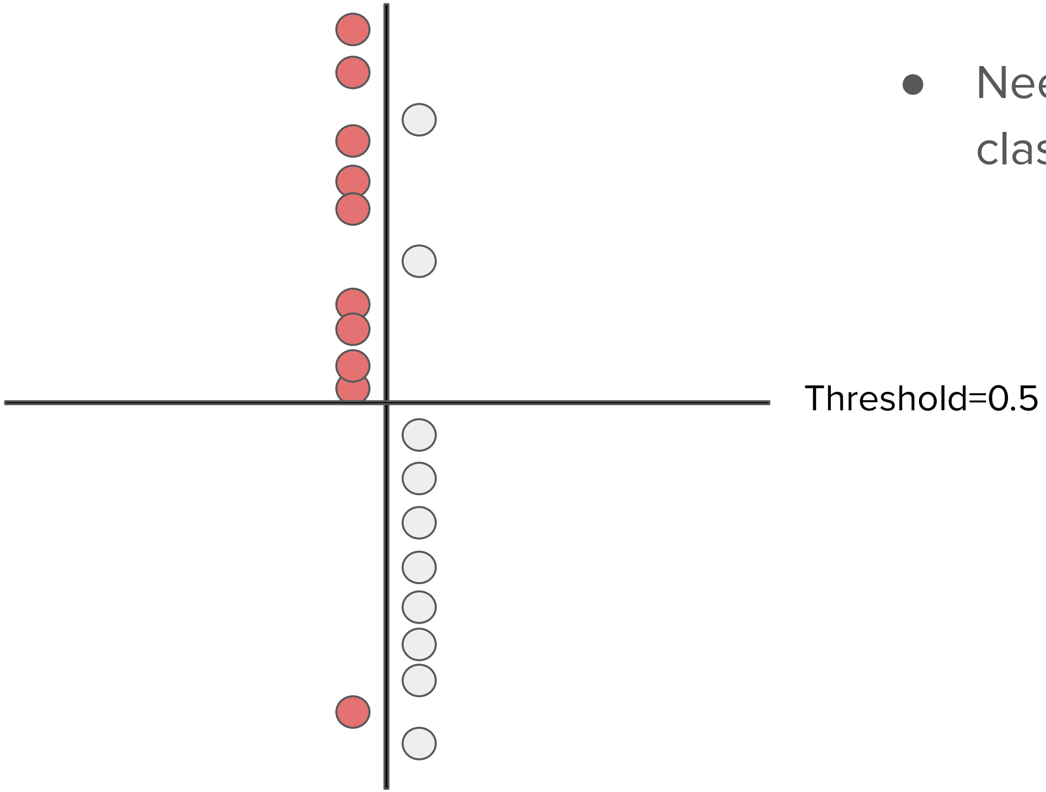
Classifier



Threshold=0.5

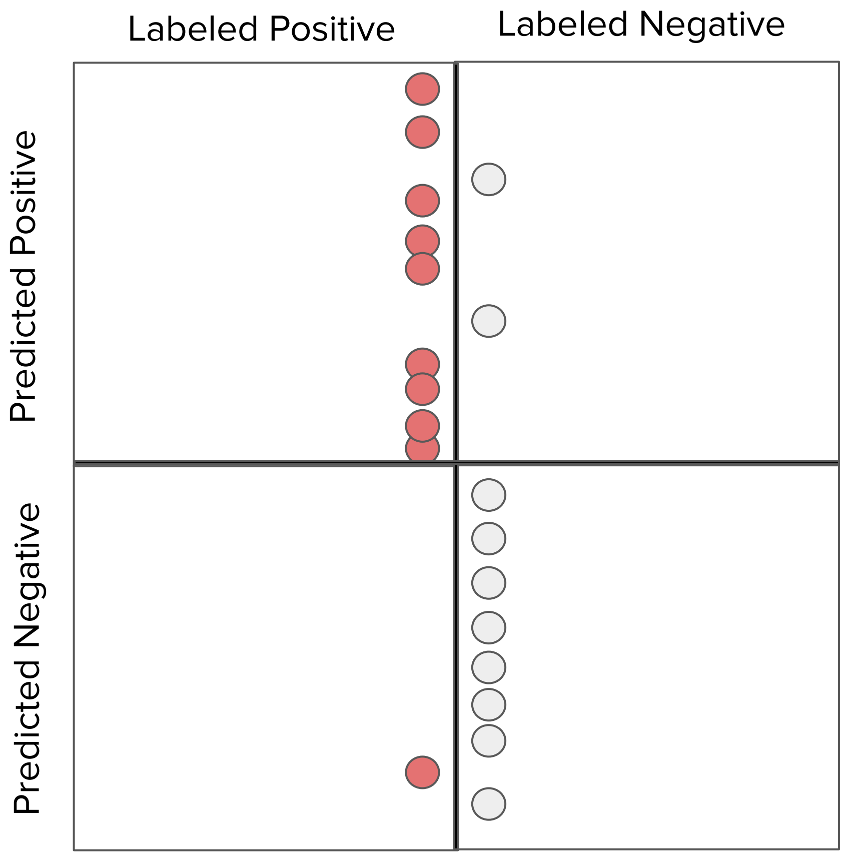
- Need to pick a threshold to have a classifier

Classifier



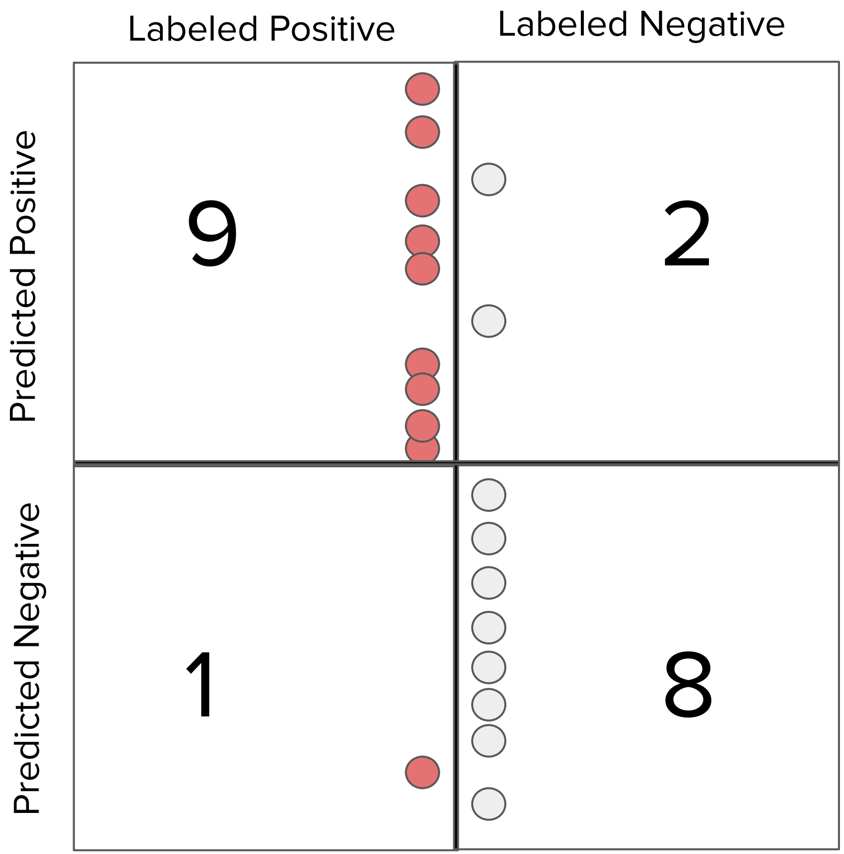
- Need to pick a threshold to have a classifier

Confusion Matrix



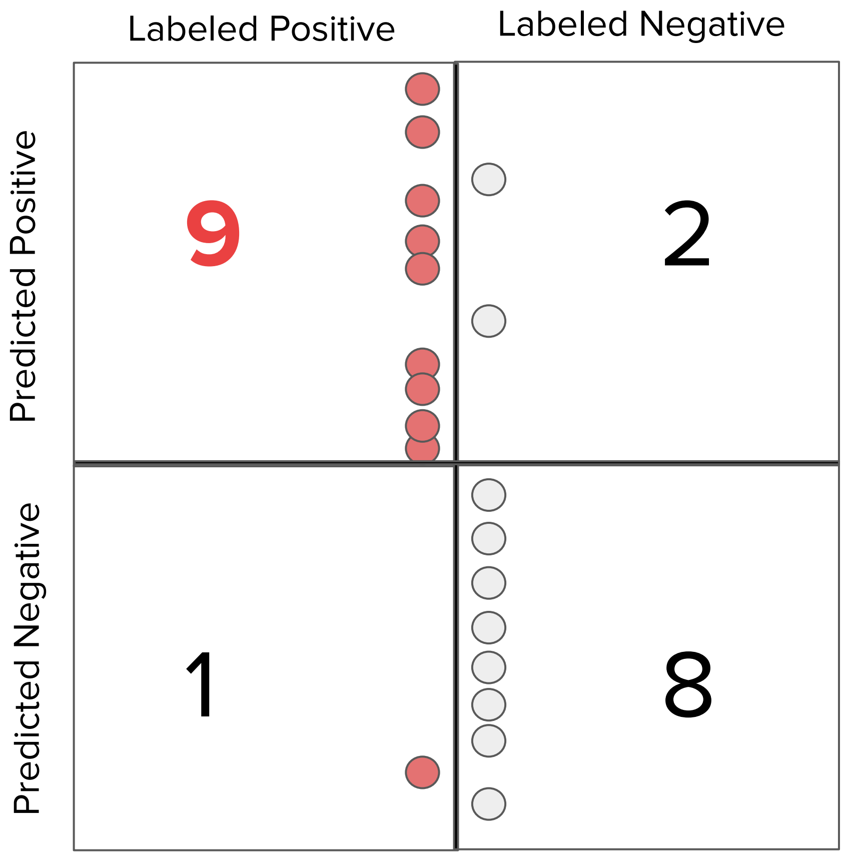
- Need to pick a threshold to have a classifier
- Once we have a threshold, can create a **confusion matrix**

Confusion Matrix

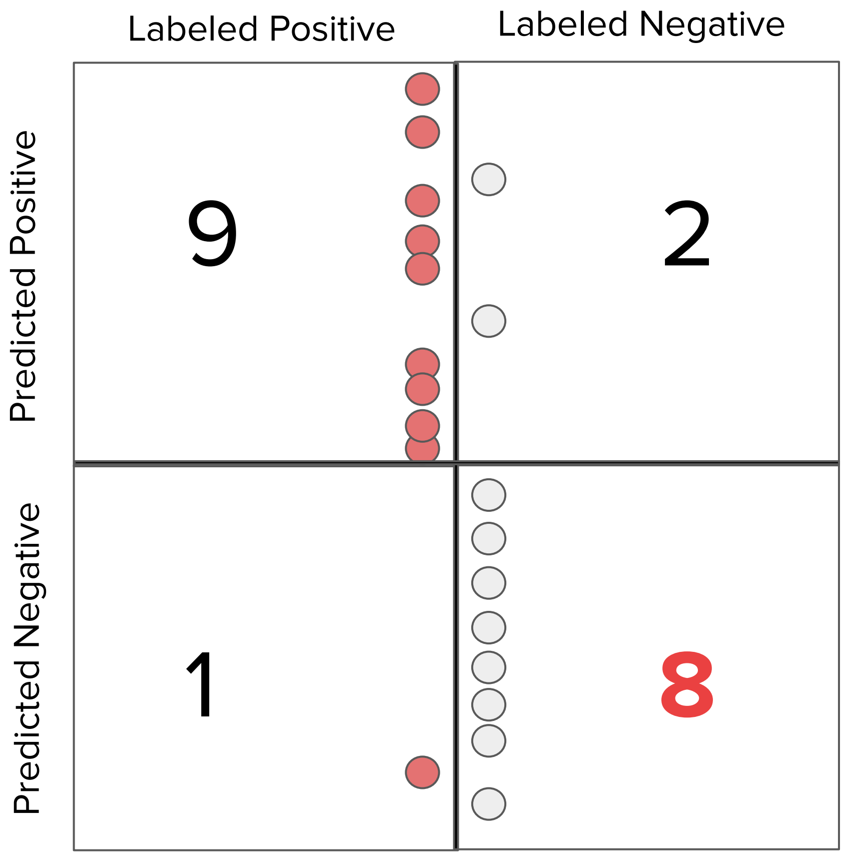


- Properties
  - Total sum is fixed (population)
  - Column sums are fixed (class-wise population)
  - Threshold determines how to split into rows
  - Want diagonal entries to be large and off-diagonal small

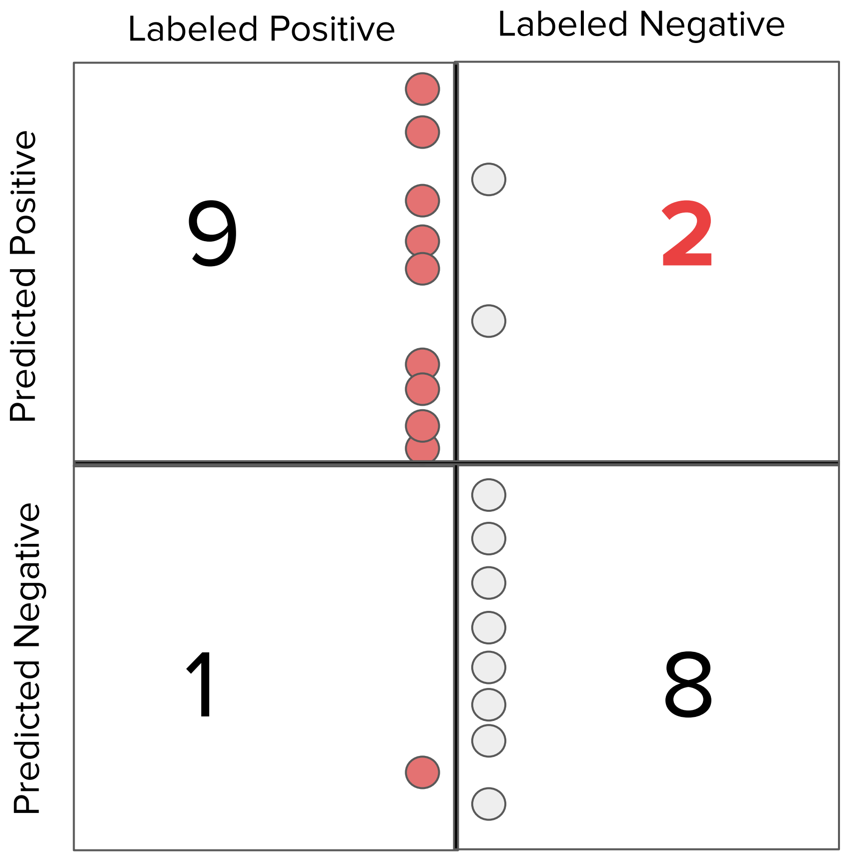
Point Metrics: True Positives



Point Metrics: True Negatives

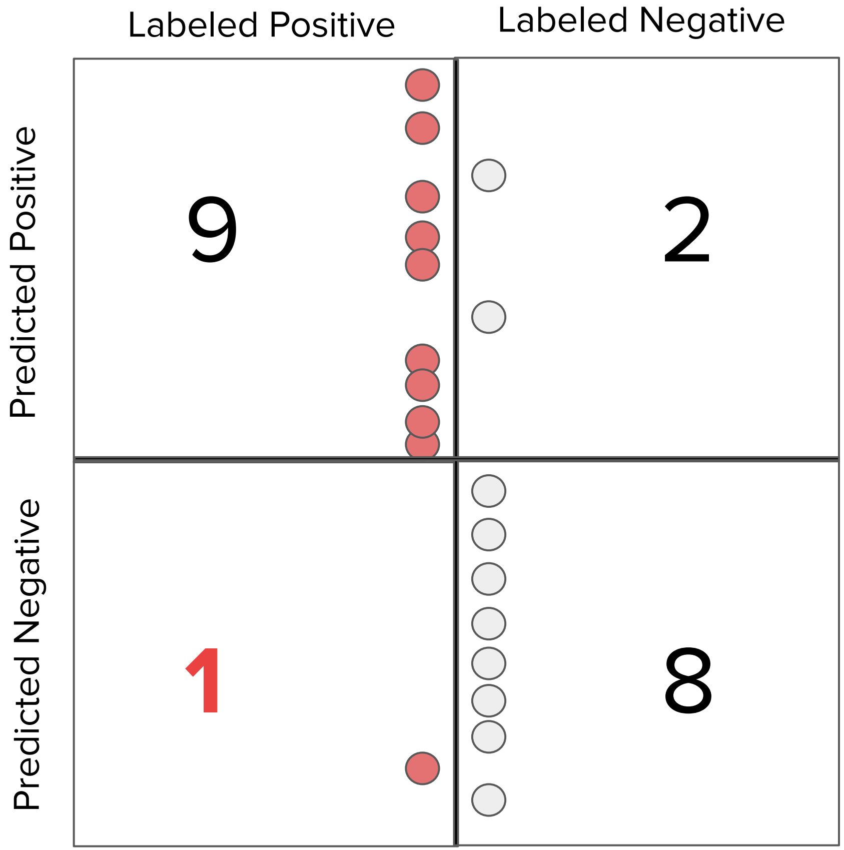


Point Metrics: False Positives



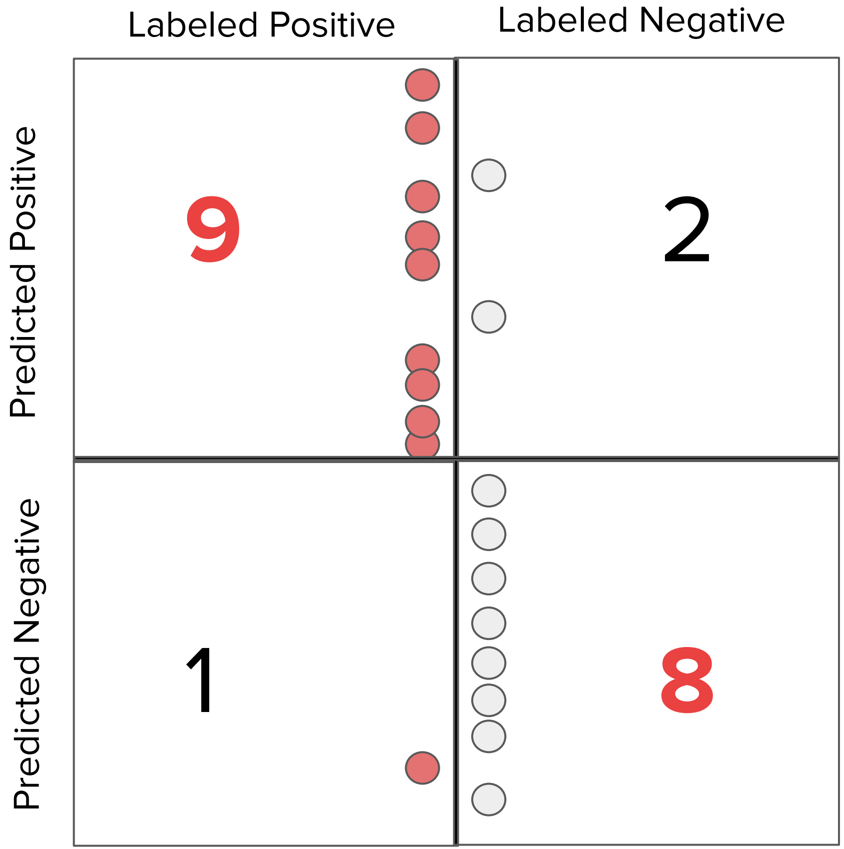
- Type-I error

Point Metrics: False Negatives



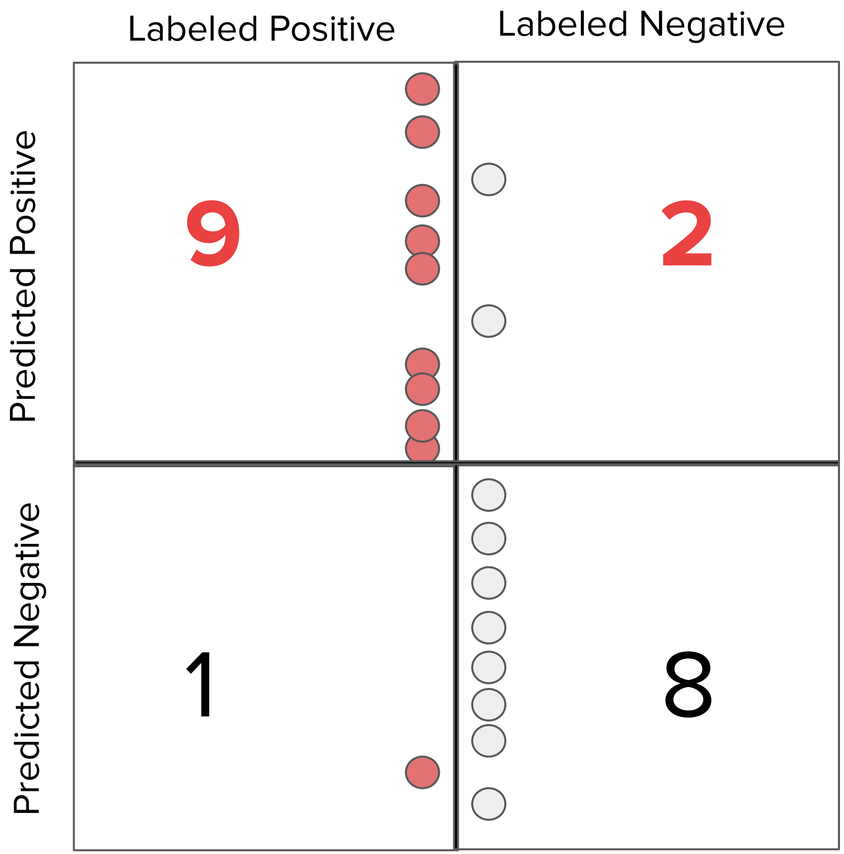
- Type-II error

Point Metrics: Accuracy



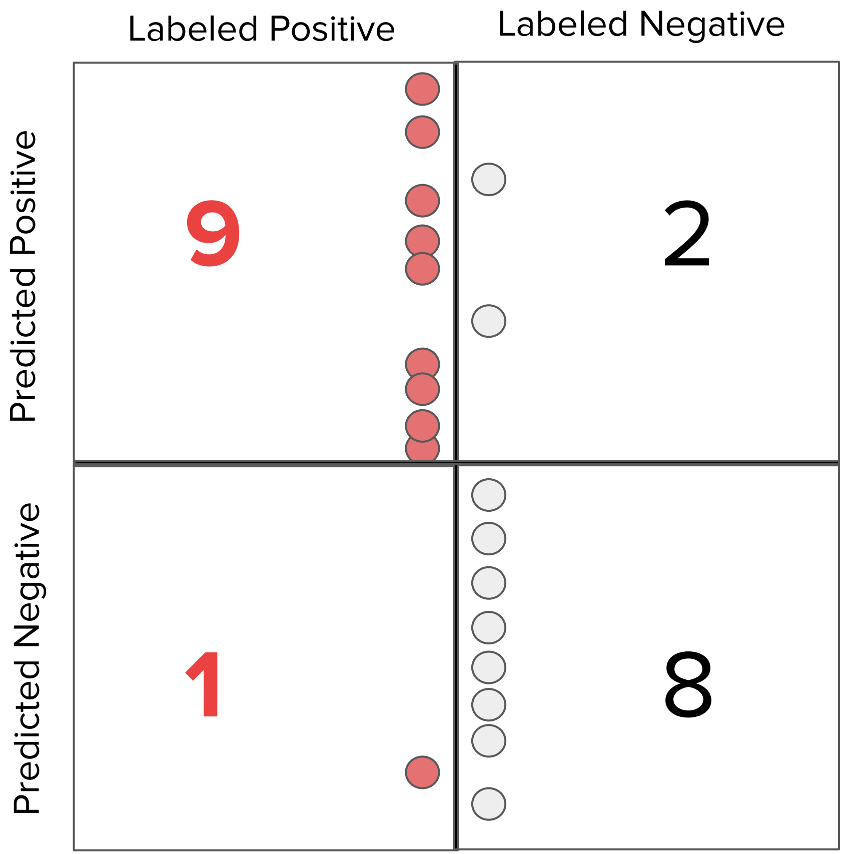
- Accuracy = (TP + TN) / Total
  - (9 + 8) / 20 = 0.85
- Equivalent to 0-1 loss.

Point Metrics: Precision



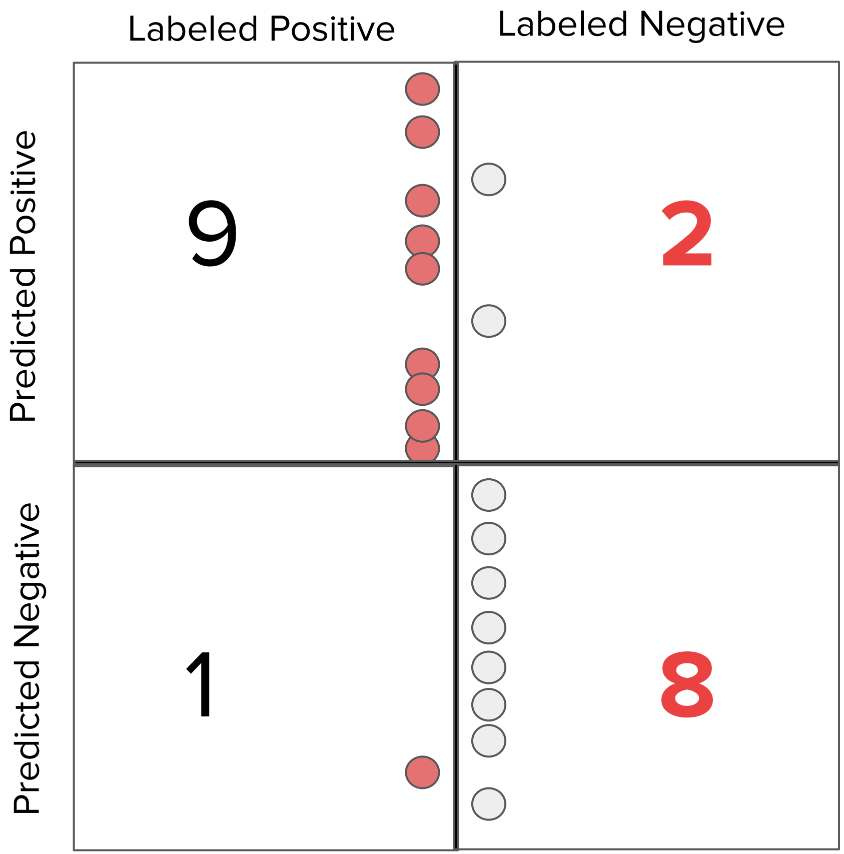
- Precision = TP / (TP + FP)
    - 9 / (9 + 2) = 0.82
- **Out of model predicted positives, how many were actually positive?**
    - p(y_true = 1 | y_pred = 1)
- Also called *PPV* (Positive Predictive Value)
- Trivial 100% precision: threshold to right before topmost labeled positive (if possible)

Point Metrics: Positive Recall



- Recall = TP / (TP + FN)
  - 9 / (9 + 1) = 0.9
- **Out of all the positives, how many did the model predict positive?**
  - p(y_pred = 1 | y_true = 1)
- Also called *sensitivity*
- Trivial 100% recall: super small threshold
- Good precision with 100% recall: push lowest red up
- Good recall with 100% precision: push highest grey down

Point Metrics: Negative Recall



- Negative Recall = TN / (TN + FP)
  - 8 / (8 + 2) = 0.8
- **Out of all the negatives, how many did the model predict negative?**
  - p(y_pred = 0 | y_true = 0)
- Also called *specificity*
- Sensitivity and specificity operate in different sub-universes

Point Metrics: F-score

- Summarize **precision** and **recall** into a single score

Point Metrics: F-score

- Summarize **precision** and **recall** into a single score
- *F1-score*: Harmonic mean of precision and recall
  - Why harmonic mean?

Point Metrics: F-score

- Summarize **precision** and **recall** into a single score
- *F1-score*: Harmonic mean of precision and recall
  - Why harmonic mean?

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{1}{\frac{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}{2}}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Point Metrics: F-score

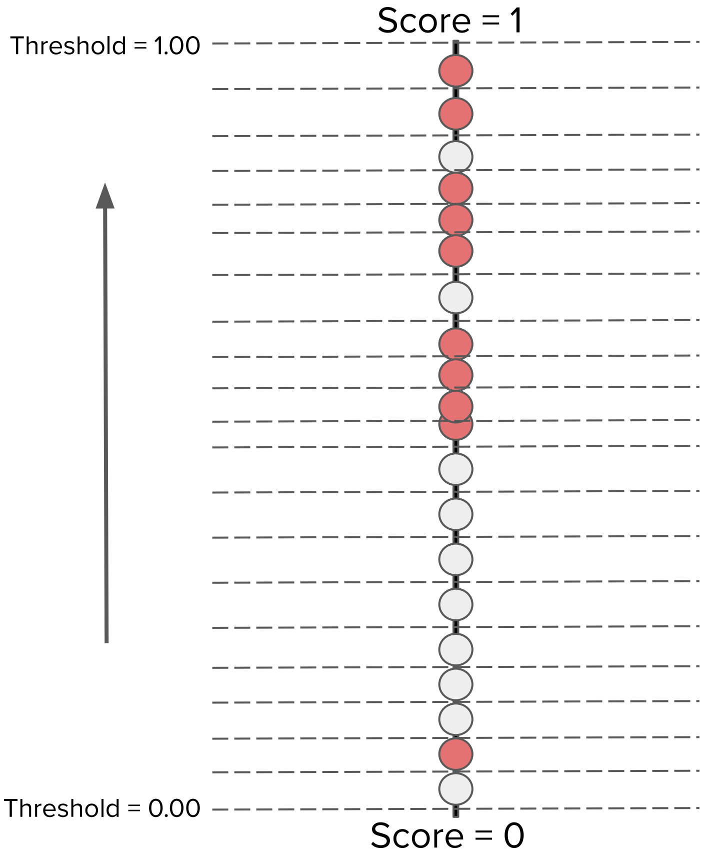
- Summarize **precision** and **recall** into a single score
- *F1-score*: Harmonic mean of precision and recall
  - Why harmonic mean?
- *Fβ-score*:

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}$$

Point Metrics: Varying the threshold

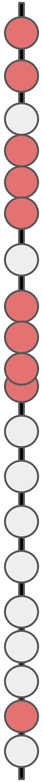
- Changing the threshold can result in a **new confusion matrix**, and new values for some of the metrics
- Many threshold values are redundant (between two consecutively ranked examples)
  - Number of effective thresholds = # examples + 1

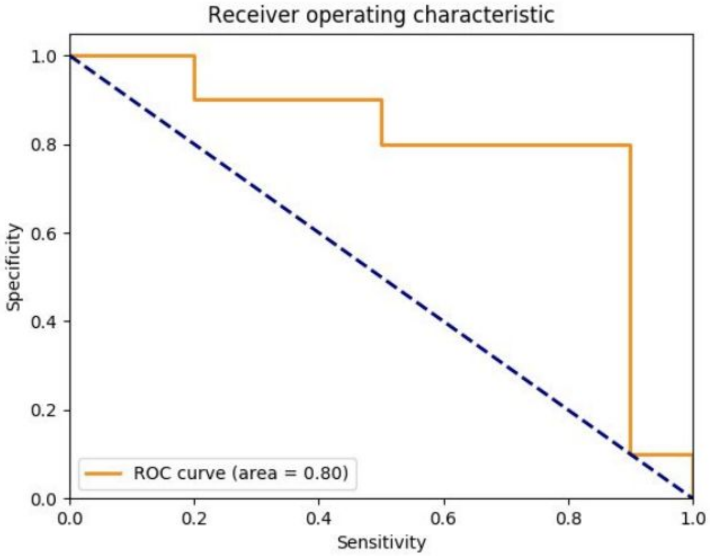
Point Metrics: Varying the threshold



Threshold	TP	TN	FP	FN	Accuracy	Precision	Recall	Specificity	F1
1.00	0	10	0	10	0.50	1	0	1	0
0.95	1	10	0	9	0.55	1	0.1	1	0.182
0.90	2	10	0	8	0.60	1	0.2	1	0.333
0.85	2	9	1	8	0.55	0.667	0.2	0.9	0.308
0.80	3	9	1	7	0.60	0.750	0.3	0.9	0.429
0.75	4	9	1	6	0.65	0.800	0.4	0.9	0.533
0.70	5	9	1	5	0.70	0.833	0.5	0.9	0.625
0.65	5	8	2	5	0.65	0.714	0.5	0.8	0.588
0.60	6	8	2	4	0.70	0.750	0.6	0.8	0.667
0.55	7	8	2	3	0.75	0.778	0.7	0.8	0.737
0.50	8	8	2	2	0.80	0.800	0.8	0.8	0.800
0.45	9	8	2	1	0.85	0.818	0.9	0.8	0.857
0.40	9	7	3	1	0.80	0.750	0.9	0.7	0.818
0.35	9	6	4	1	0.75	0.692	0.9	0.6	0.783
0.30	9	5	5	1	0.70	0.643	0.9	0.5	0.750
0.25	9	4	6	1	0.65	0.600	0.9	0.4	0.720
0.20	9	3	7	1	0.60	0.562	0.9	0.3	0.692
0.15	9	2	8	1	0.55	0.529	0.9	0.2	0.667
0.10	9	1	9	1	0.50	0.500	0.9	0.1	0.643
0.05	10	1	9	0	0.55	0.526	1	0.1	0.690
0.00	10	0	10	0	0.50	0.500	1	0	0.667

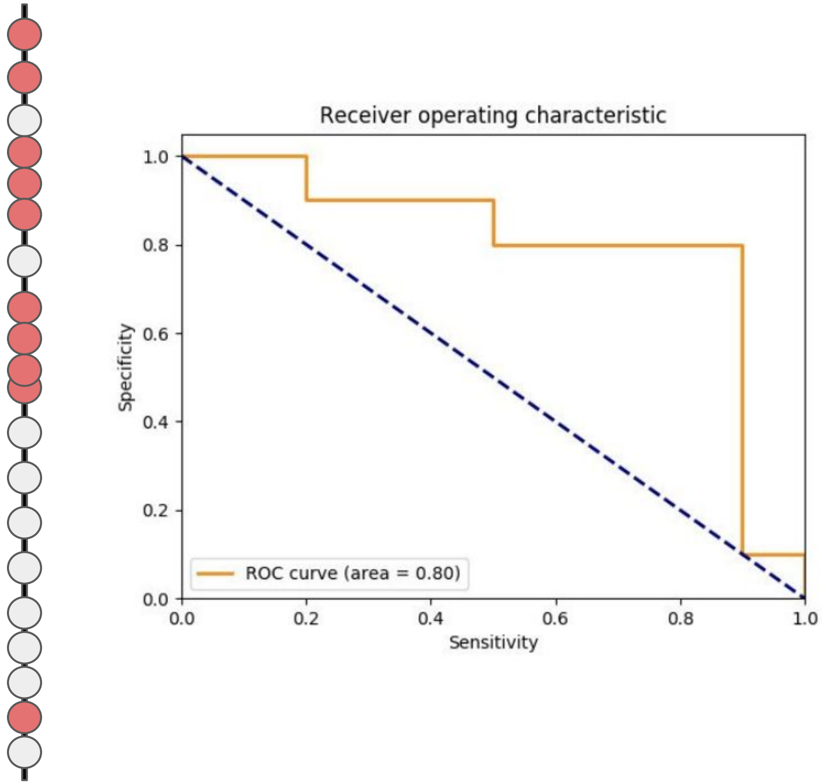
Summary Metrics: ROC Curve





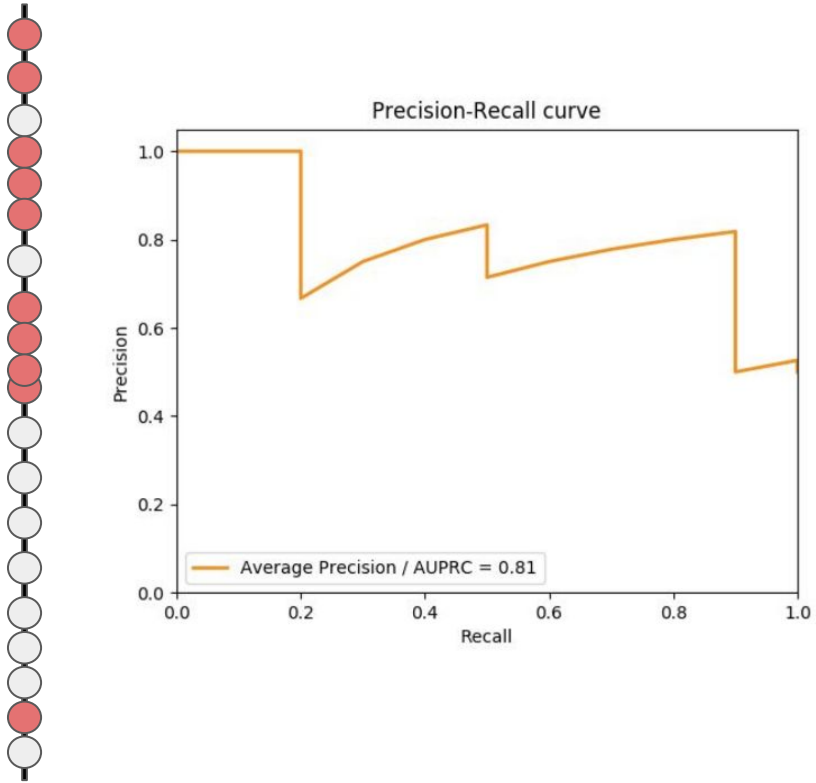
- Each point defined by a threshold, and corresponds to point metrics
- Most people plot FPR (1 - Specificity) against TPR (Recall, Sensitivity)
- Diagonal line = *random guessing*
- How do you select a threshold?
  - Youden's J
  - F1
  - High sensitivity/specificity
  - ...
  - **Always use the validation set!**

Summary Metrics: AUC



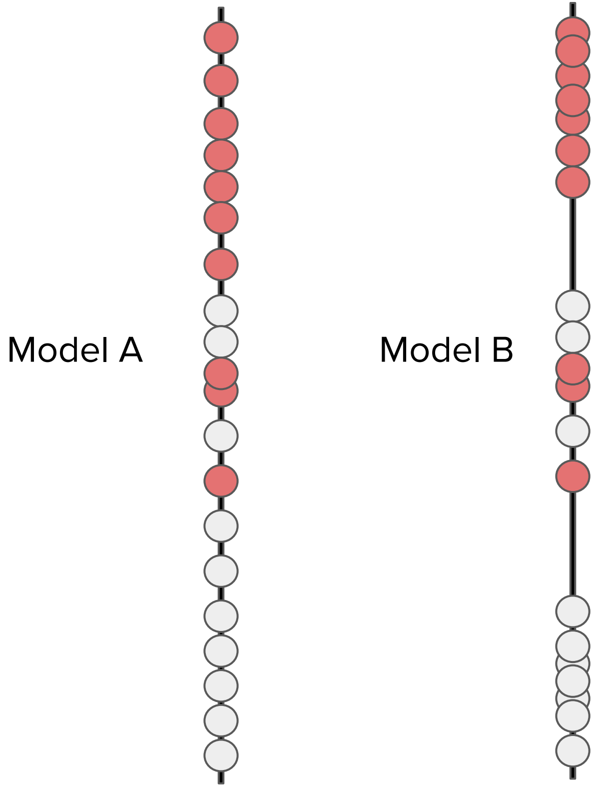
- *Area Under the Curve* (AUC)
  - AKA Concordance statistic/index (C-statistic, C-index)
- Rank-statistic, i.e. **only depends on the ordering** of the positives/negatives.
  - This is known as *discrimination*
- If you pick positive and negative by random, **probability that the positive ranked higher than the negative**

Summary Metrics: PR Curve



- End of curve at right cannot be lower than prevalence - why?
- *Area under PRC* (AUPRC) = average precision
  - By randomly picking the threshold, what's the **expected precision**?

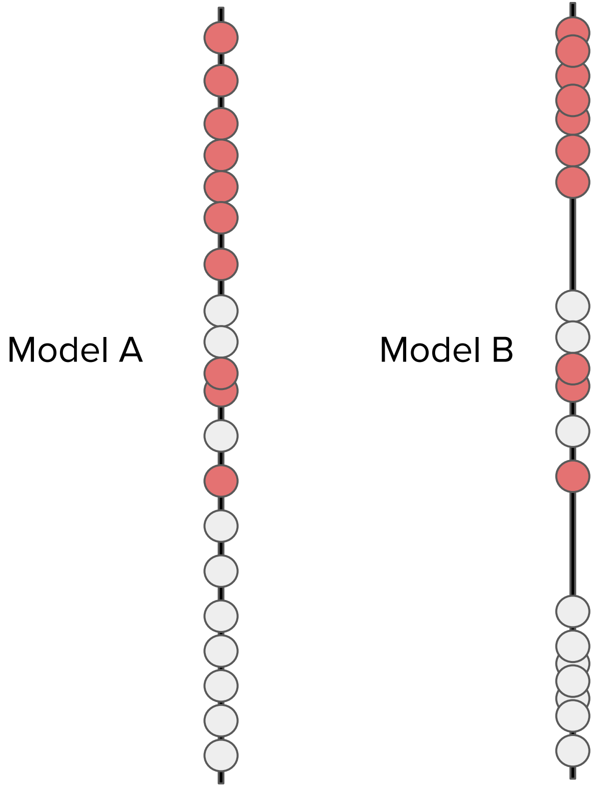
Summary Metrics: Log-loss



Model scores on the same dataset.

Which is better?

Summary Metrics: Log-loss

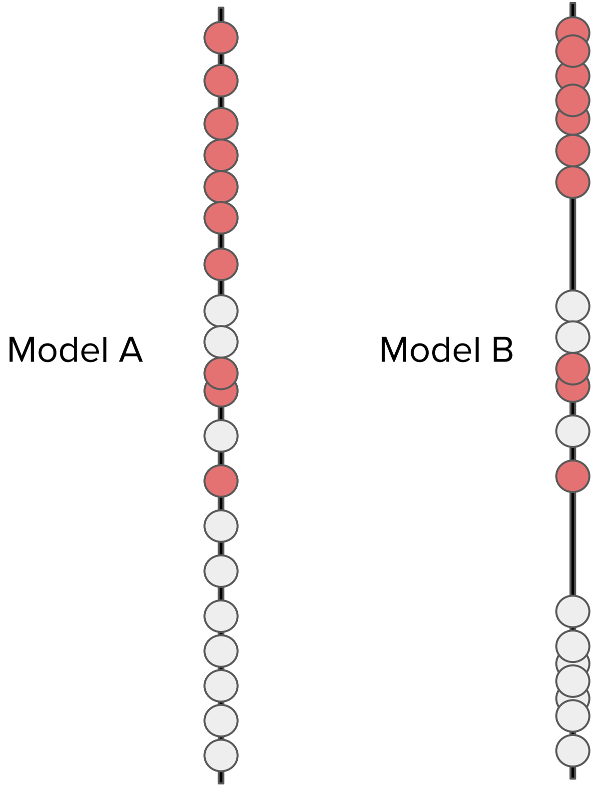


- Same AUROC, AUPRC, point metrics etc. (same *discrimination*)
- Log-loss (cross entropy) rewards confident correct predictions and **heavily penalizes** confident incorrect predictions.

$$-(y\log(p) + (1-y)\log(1-p))$$

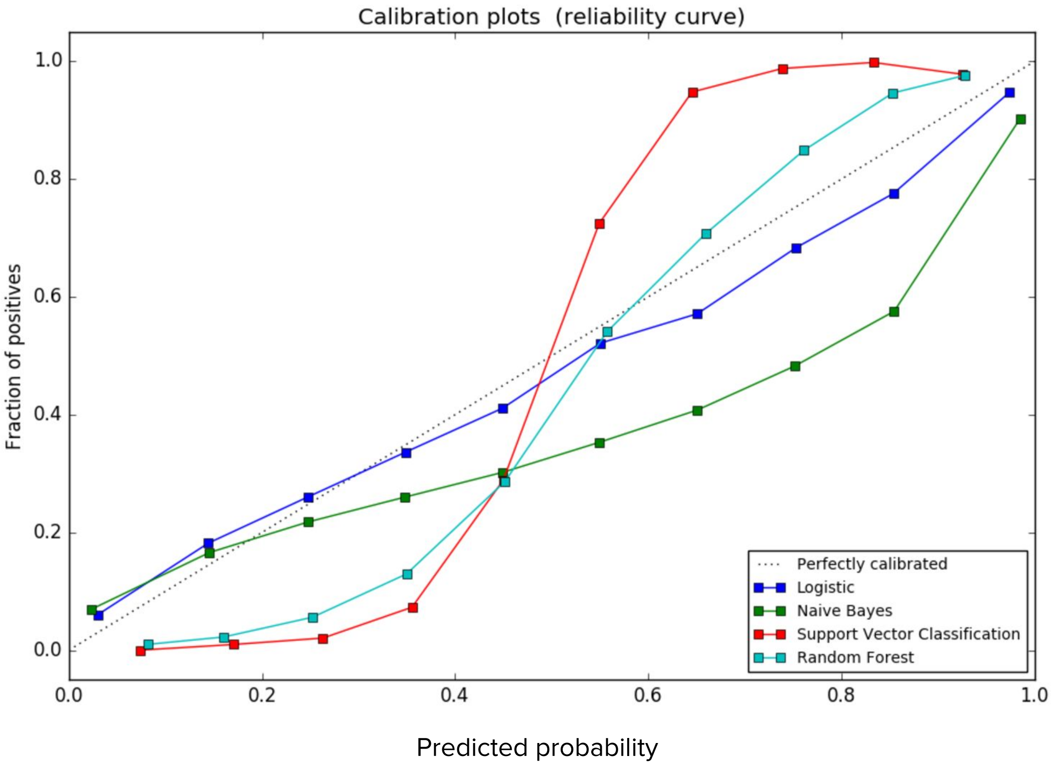
- All 0.5 predictions:
  - -log(0.5) = 0.69
  - For C > 2 classes, log(C) is random/uniform loss.

Summary Metrics: Log-loss



- So log-loss captures more than just discrimination
- It also captures *calibration*, i.e. how well the model's predictions actually **correspond to confidences**
- Log-loss encourages calibration (*proper scoring rule*)

Calibration Metrics: Reliability Diagrams



Plot binned predicted probabilities against fraction of positives within each bin

Calibration Metrics: Techniques for Calibrated Models

- Histogram binning
- Platt scaling
- Isotonic regression
- …
- See On Calibration of Modern Neural Networks for a nice overview!

Class Imbalance: Problems

- Symptom: prevalence < 5%
- Metrics lose meaning
- Inhibits learning
    - E.g. logistic regression can be overwhelmed by majority class

Class Imbalance: Metrics

- Accuracy: high score just by **predicting majority class**
  - This should be the low-bar!
- Log-loss: majority class can dominate
- AUROC: Can attain high AUROC by *scoring negatives low*
  - Artificially increased by true negatives
  - 10% prevalence. top-10% are all negatives, next are all the positives, followed by the rest of the negatives. AUROC = 0.9.
- AUPRC: Somewhat more robust, but other challenges
  - How do you interpolate?
- For class imbalance in general: **Accuracy << AUROC << AUPRC**

Multi-class

- Confusion matrix will be NxN (still want heavy diagonals, light off-diagonals)
- Most metrics (except accuracy) generally analyzed as several 1-vs-many comparisons
- Class imbalance is common (both in absolute, and relative sense)
- Cost sensitive learning techniques (also helps in binary imbalance)
    - Assign weighted value for each block in the confusion matrix, and incorporate those into the loss function

Summary

- Score-based binary classification models
- Point metrics vs. summary metrics
- Discrimination vs. calibration
- Evaluation with class imbalance and multiple classes