Model Validation

Peter Casey

A little about me...

- PhD in Political Science from WashU
    - Failed Ryan Moore's first exam
    - Passed the class with an A-
- Work at intersection of data science and politics / public policy
    - Data Scientist, Democratic National Committee
    - Sr. Data Scientist, The Lab @ DC
    - Director of Analytics, Catalist
    - Program Director, Data Science for Social Good
    - Director of Strategy, California Policy Lab

What’s the purpose of a predictive model?

A predictive model helps us make inferences about outcomes we can’t observe using information about outcomes we have observed

- Making inferences about future outcomes using past outcomes
- Making inferences about all of our outcomes of interest with information about a sample of outcomes

Why do we validate our models?

- Make sure our model's predictions generalize to unobserved outcomes

- Compare performance across models to select the best option

- Check that the model is performing well across subgroups within your population (checking for <u>bias</u> and <u>underperformance</u>)

Understand the problem you’re trying to solve

- What is the outcome you’re trying to predict?
- Can you observe the outcome you’re trying to predict, or is your data only a proxy for it?
- How common is that outcome?
- What population is impacted by this outcome?
- How was the data you’re using generated? Does it represent the population?
- Once you identify the outcome you’re trying to predict, what action will you take?
    - Are you trying to apply a costly intervention efficiently (i.e., minimize false positives)?
    - Or are you trying to avoid missing costly outcomes (i.e., minimize false negatives)?
- What context(s) will your model be applied?

Validation Decisions

- **Performance metric** to optimize

- How you'll construct your **validation set(s)**

- Figure out what **subsamples** you may want to validate
    - Where your model may be **biased** or **underperform**

Choosing your Performance Metric

Performance metrics

- Focus on classifiers
  - More common
  - More performance metrics to choose between
- Important to return to the problem you’re solving
  - Are you trying to…
    - Efficiently target a costly intervention? (Precision)
    - Identify as many instances of an outcome as you can? (Recall)
    - Distinguish between two different outcomes? (ROC-AUC)
    - Fit closely to actual probabilities? (Brier Score, Log Loss)

What we’re optimizing for

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

A note on accuracy

- The proportion of correct predictions
- (TP + TN) / (TP + TN + FP + FN)
- Commonly used, but *usually not the right metric*
    - Treats false positives and false negatives *equally*. Usually, we care more about one than another.
    - Usually, sets threshold for positive and negative prediction at 0.5, but this may not be the best threshold value (more on this later)

Precision: Minimizing False Positives

- The proportion of the model’s positive classifications that are actually positive
- **Positive Predictive Value:** TP / (TP + FP)
- How often the model’s positive guesses are correct
- Useful when you have limited resources and want to identify the best targets for your intervention (e.g., inspecting for rodents, reaching out to voters)

Precision @ N

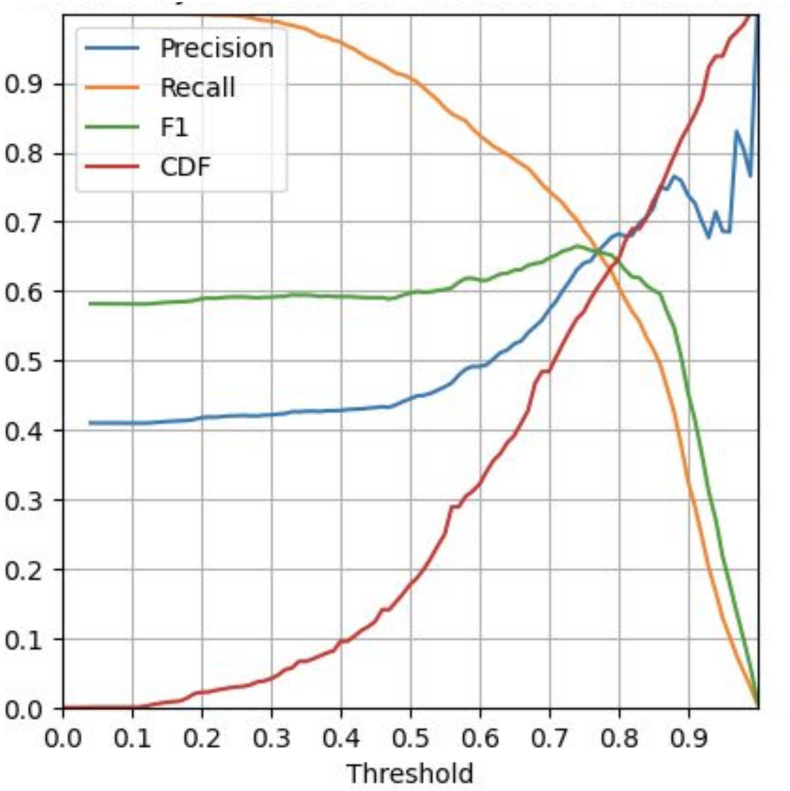
- Precision typically looks at the proportion of correct classifications with a predicted probability over 0.5
- When resources are severely constrained, and you know how many people you want to reach out to you may want to use **Precision @ N**
- First identify the number N you want to target
- Then look at precision for the N targets with the highest predicted probability

Recall: Minimizing False Negatives

- **Recall** (also known as **sensitivity**) is the the proportion of actual positives that the model correctly identifies
- **True Positive Rate**: TP / (TP + FN)
- Useful when the cost of your intervention is low but the cost of missing a positive case is high (e.g., fraud)
- **Recall @ N**: The proportion of actual positives your model accurately identifies in the top N predictions

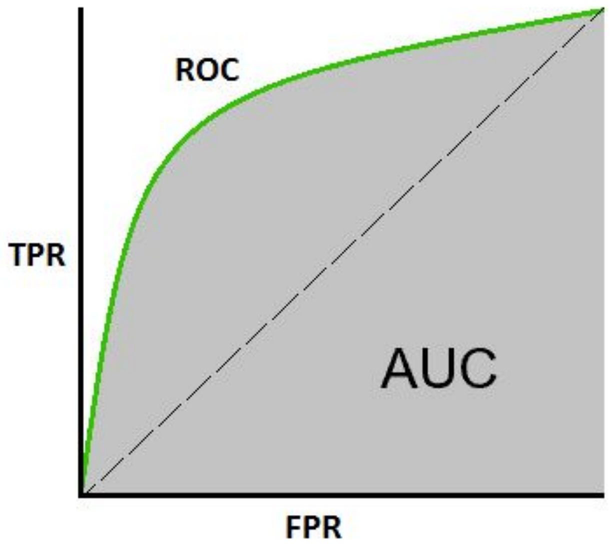
Combining Performance Metrics

- Trade-off between precision and recall
- Considen precision and recall curves on when evaluating:
  - Model performance
  - Choosing model decision threshold
- Example
  - At 0.8
    - Precision: ~0.7
    - Recall: ~0.6
    - CDF: ~0.65, or 35% of sample



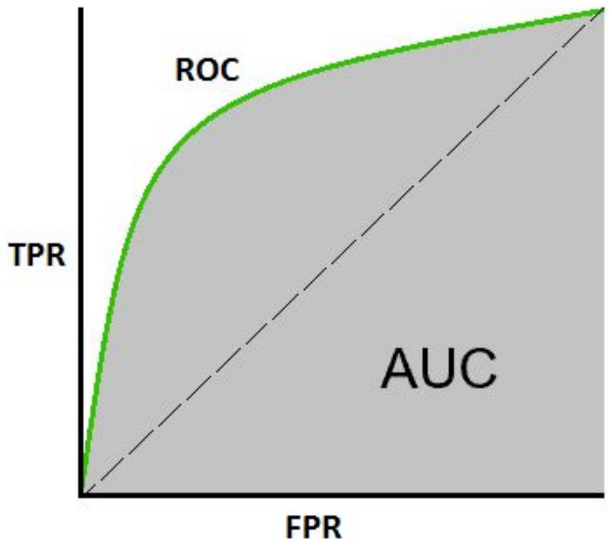
Receiver Operating Characteristic (ROC)

- Graphical plot summarizing trade-off between true-positive rate (sensitivity or recall) and true-negative rate (specificity)
- Similar to Type 1 and Type 2 error: as the TPR increases, so does the FPR
- As our model threshold accounts for more positive observations, it also must include more negative ones



ROC-AUC

- Area under the curve summarizes trade-off
- Gives the probability that the model correctly distinguishes between two classifier labels
- This metric is best if you:
  - Care about ranking
  - Care about distinguishing between positive and negative classes (e.g., distinguishing between Democrats and Republicans)
  - Sample is not heavily imbalanced
    - If sample is imbalanced, and you care about making correct positive predictions, the AUC of the **precision-recall curve** is better



Other performance metrics

- **Fbeta:** Summarize trade-off between precision and recall. Beta parameter allows you to prioritize precision (e.g., beta = 0.5) or recall (e.g., beta = 2)
  - May seem like the "best of both worlds" but in truth it's probably not answering the question you want to answer: Minimizing false positives or false negatives
- **Calibration metrics:** Summarize how well model predictions face actual probabilities
  - **Brier score:** Average difference between predicted and actual probability (similar to MSE)
  - **Log-loss:** Uses log of probabilities to penalize large errors
  - Rarely the goal of a predictive model, but can be helpful if you're trying to fit your predictions closely to the actual probability of an outcome

For Regressors

- **Root Mean-Squared Error (RMSE):** Summarizes the distance between predicted values and actual outcome, while penalizing outliers

- **Mean Absolute Error (MAE):** Summarizes the distance between predicted values and actual outcome, but does **NOT** penalize outliers

Constructing your Validation Set

Return to your problem

- What problem are you trying to solve?
  - Are you predicting what will happen in the future from what was observed in the past?
  - Are you generalizing from a sample to a population?
- What groups in your sample might your model predict differently than others? What's more costly for them: false positives or false negatives?
- How will your model be used?
  - Is the context that generates the outcomes you're trying to predict the same as the process that generated your data?
  - If not, you may want to validate your model against data generated by that process
    - Examples:
      - Data collected in the field
      - Data collected independently that yields similar outcomes to your own data-generating process

Train, Test, Validate

- When training a predictive model, it's often desirable to have **at least** three different data sets
  - Training
  - Testing
  - Validation
- The purpose of the testing set is to ensure that your model does not overfit to the training set and therefore performs well on data out of sample
- If you model iteratively, you can also risk overfitting to your **testing set**
- **<u>Always</u>** have an independent validation set

Validation Set

- Ensure distributions match population of interest
- May want to oversample smaller subgroups to avoid missing rare observations
- May be a subset of the data you have on hand for training the model
- Important to compare your model’s performance to data generated by the same process where your model will be used
    - Examples:
        - Data collected independently that yields similar outcomes to your own data-generating process
        - Data collected in the field

Cross-validation

- Simple train-test approaches are actually a special case of cross-validation

- Cross-validation usually involves splitting your training set into multiple cross-sections (often 3 to 5, sometimes more), holding out one cross-section and training the model on the rest

- This can also be difficult with a smaller data set

Cross-validation

- Cross-validation is especially powerful when you have natural cross-sections in your data
- One good example is time-series cross-validation: dividing a data set into months or years and then predicting future outcomes based on models trained on past data
- Others may include geographic or cohort cross-validation
  - Examples:
    - Voting behavior across states
    - School attendance among different classes (e.g., Class of 2024, Class of 2025, etc.)

Validating against subgroups

- Models sometimes perform differently on subgroups in your data, and may perform better on some subgroups than others
- It is important to validate your model for subgroups where the model may perform differently
- Common disparities:
    - Differences in performance (the model performs better for some subgroups than others)
    - Differences in errors (the model tends to overpredict the outcome for some subgroups and underpredict it for others)
        - Important to know **what's more costly**: false positives or false negatives
        - In **public policy**
            - **False negatives** are more costly if you're providing a public benefit
            - **False positives** are more costly if you're making a punitive decision

Validating against subgroups: Examples and risks

- Recidivism: Costly false positives
  - ProPublica COMPAS Recidivism Score
  - Recidivism is more common among Black people in part because of an unjust history of over-policing
  - Models tend to overpredict the likelihood that Black people will commit another crime and underpredict that white people will commit another crime
  - If a person used such a model to make bail decisions, they may overly penalize Black defendants compared to White defendants

Independently-Collected Validation Sets

- Field validations: Data collected in the field that simulates who your model will be used
- Survey data collected independently that should yield similar outcomes to your own data-generating process

Example: Field Validation

- Rodent inspection field validation
  - Randomly-selected 100 city blocks for inspection with a predicted probability over 0.5
  - Rodent Control inspected each block and recorded if they found rat burrows
  - Compared proportion of locations with rat burrows to predicted probabilities
- Phone quality score validation
  - Select ~100k phone numbers (10k / decile)
  - Collect phone dispositions (connected / disconnected, right / wrong person)
  - Compare phone dispositions to model predicted probabilities
- Automating pipelines to continuously validate models against incoming data

Example: Independently-Collected Survey Data

- When developing a model of support for a policy issue like reproductive choice, one may compare model predictions to the responses to a survey not used in model training
- Responses may be to similar survey questions, or to different questions that we would be expect to be correlated with the response used for training
- For example:
  - Do you think abortion should be legal?
  - To what extent do you agree: Safe, effective, and affordable methods of abortion care should be available to women in their community?
- We would expect people who agree with the former are more likely to agree with the latter, so this could be a good validation.

Questions?