Model Selection (and Validation) Part II

Rayid Ghani and Kit Rodolfa

Carnegie Mellon University

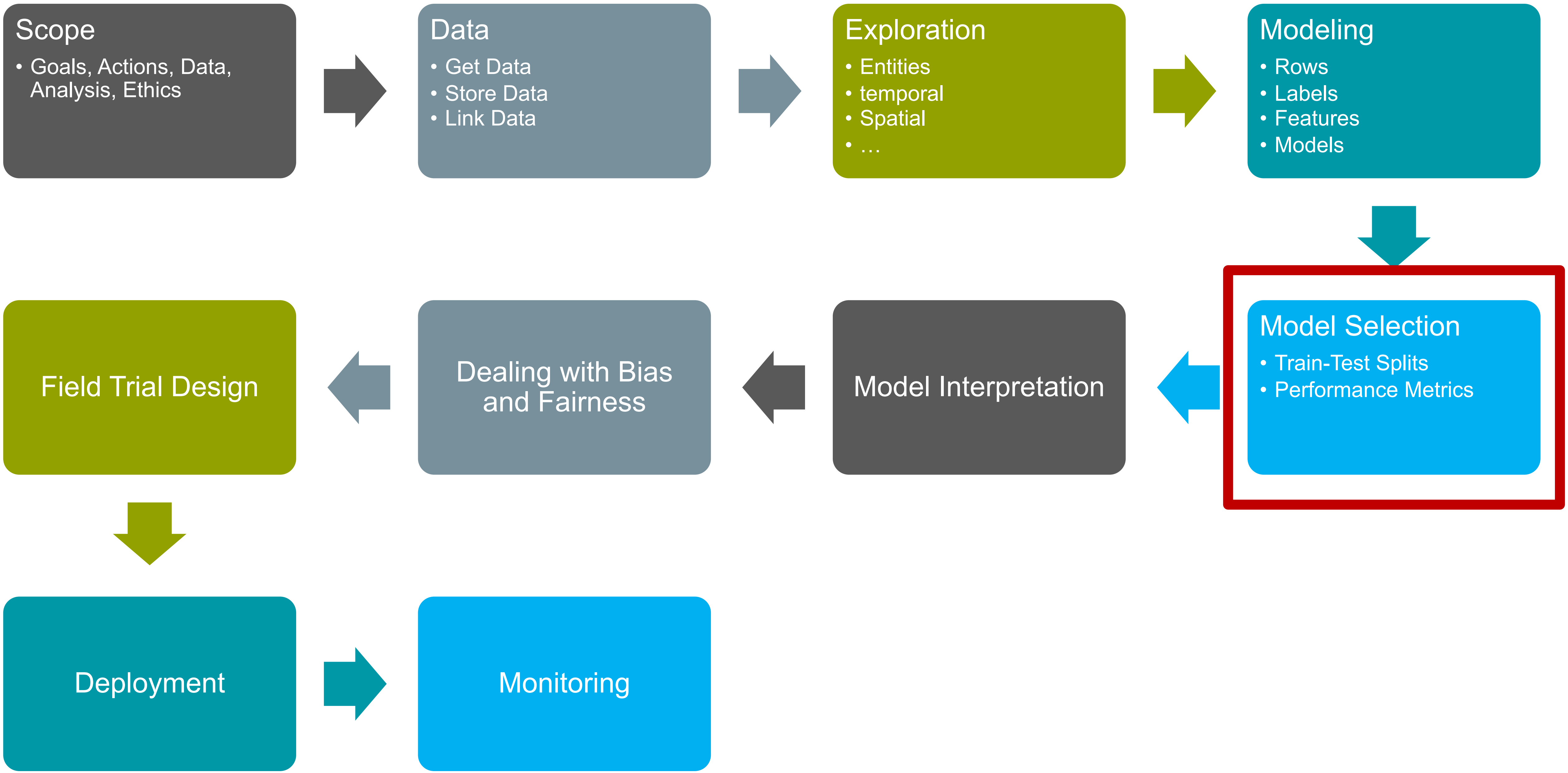
MACHINE LEARNING DEPARTMENT

Heinz College
INFORMATION SYSTEMS • PUBLIC POLICY • MANAGEMENT

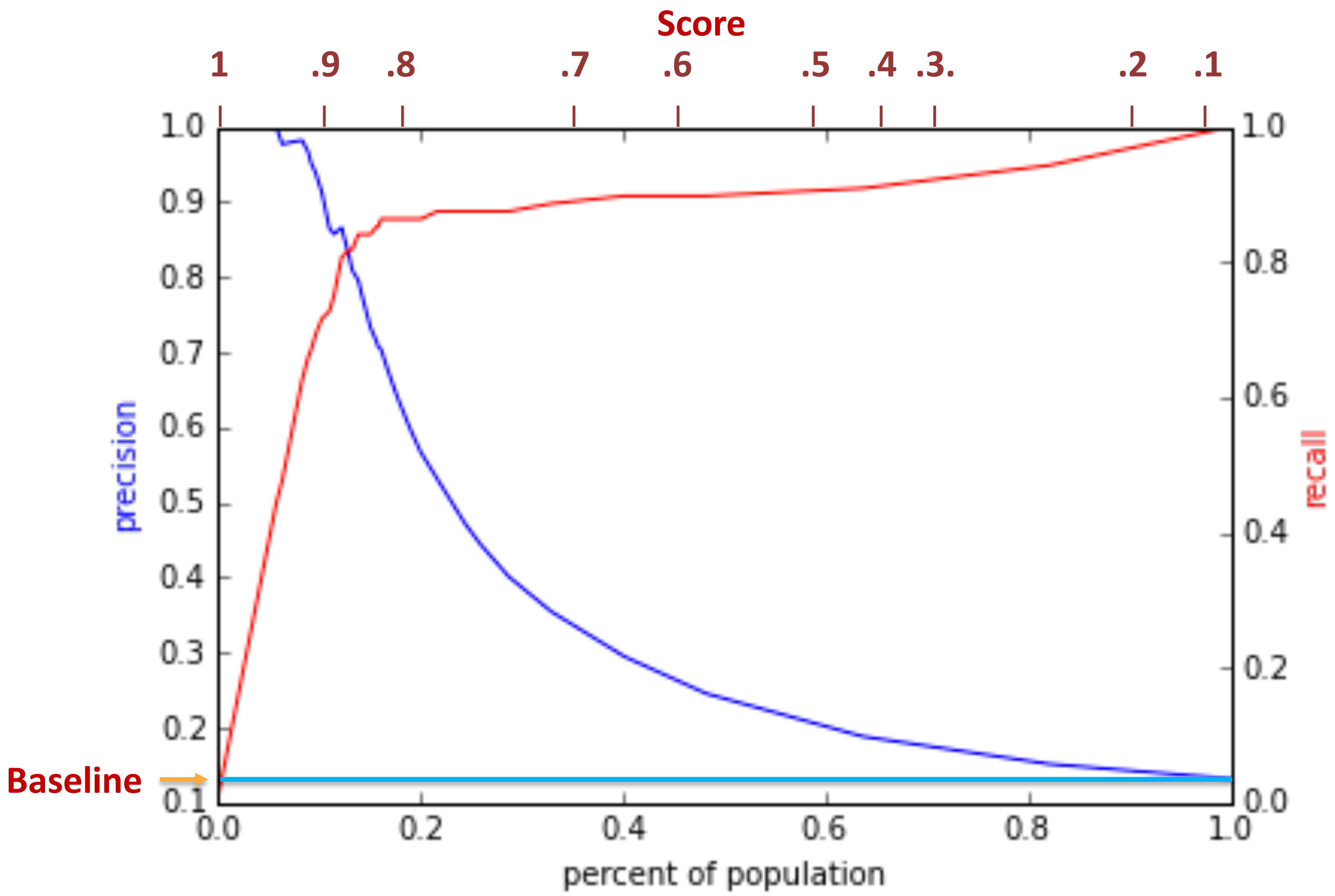
Things to remember

Coming Up Next Week:

- Monday: Modeling Results Update Assignment (posted on canvas)
- Tuesday: Weekly Feedback Form and Readings
- Midterm:
  - Take Home
  - No class on Thursday
  - Due Friday (Canvas)


Scope
• Goals, Actions, Data, Analysis, Ethics
Data
• Get Data
• Store Data
• Link Data
Exploration
• Entities
• temporal
• Spatial
• …
Modeling
• Rows
• Labels
• Features
• Models
Model Selection
• Train-Test Splits
• Performance Metrics
Model Interpretation
Dealing with Bias and Fairness
Field Trial Design
Deployment
Monitoring

Reminder: The PR-k graph



Reminder: How to solve a prediction problem

- Define and Create Rows (unit of prediction)
- Define and Create Label (outcome variable)
- Define and Create Features (predictors)
- Create Training and Validation/Test Sets
- Train model(s) on Training Set
- Validate model(s) on Validation/Test Set
- Select “best” model

Why do we need to do model selection?

- You’ve run a large number of different types of models varying …
- You need to understand what types of models are effective under what circumstances, **and**
- You need to decide which one(s) to use in the **future**

The wonderful for loop

- For train-test splits (CV or temporal)
  - For subsets of Feature Sets (Demographic only, Behavior only, Temporal only, etc.)
    - For Classifiers (RFC, SVM, DT, NN, Logit, GB, Boosting)
      - For parameters (cross products of different parameters)
        - Fit
        - Predict (predict_proba for the sklearners and no argmax for the NNers)
        - Evaluate (remember the thresholds)

Results

Classifier	Param 1	Param 2	Validation Split	Feature Subsets used	Accuracy	Prec @ 1%	AUC

Analyzing the results

K Fold CV

?

Temporal

?

Analyzing the results

K Fold CV

- Average metric value?
- Equal weight to all validation examples
- Maybe look at variance?

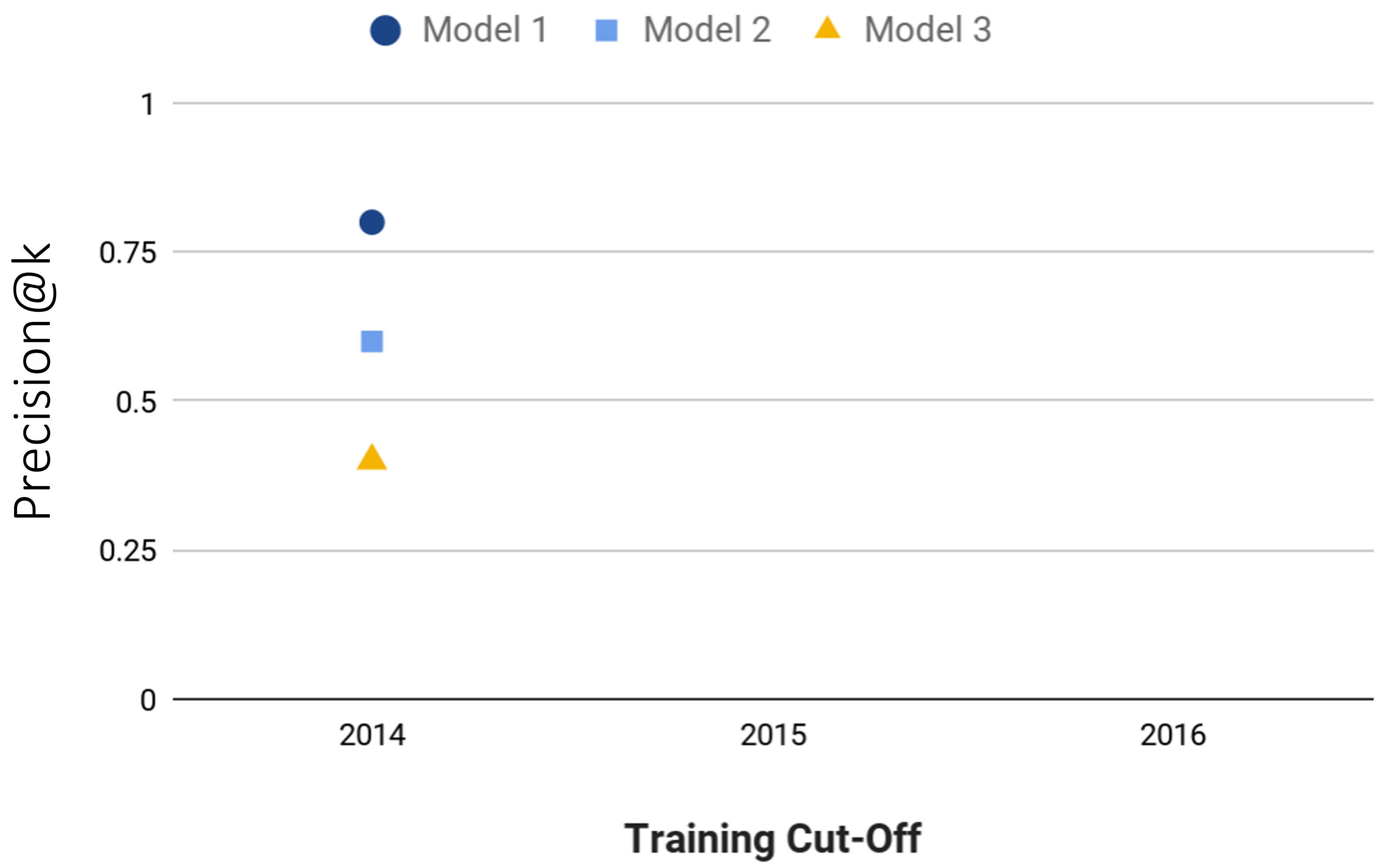
Temporal

- Average metric value?
- Robustness to distribution shifts?
- Care about recency more? How much more?
- May not weigh every validation example equally

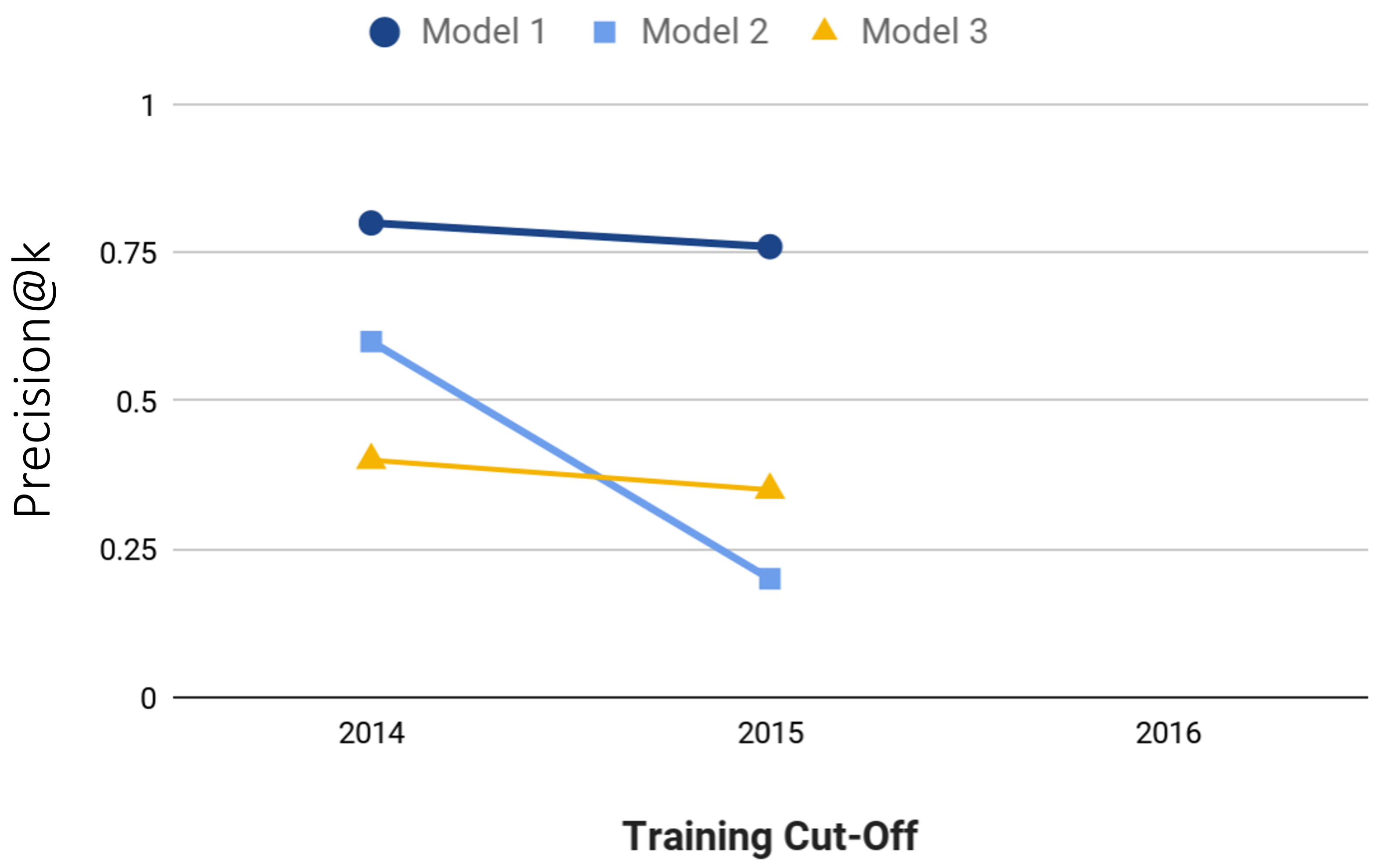
Analyzing the results

- Which approaches work best?
    - Which classifiers?
    - Which parameters?
    - Over which metrics?
- Value of different features/feature sets?
- Variance in performance over time?
    - Highest average?
    - Lowest variance?
    - Getting better over time?

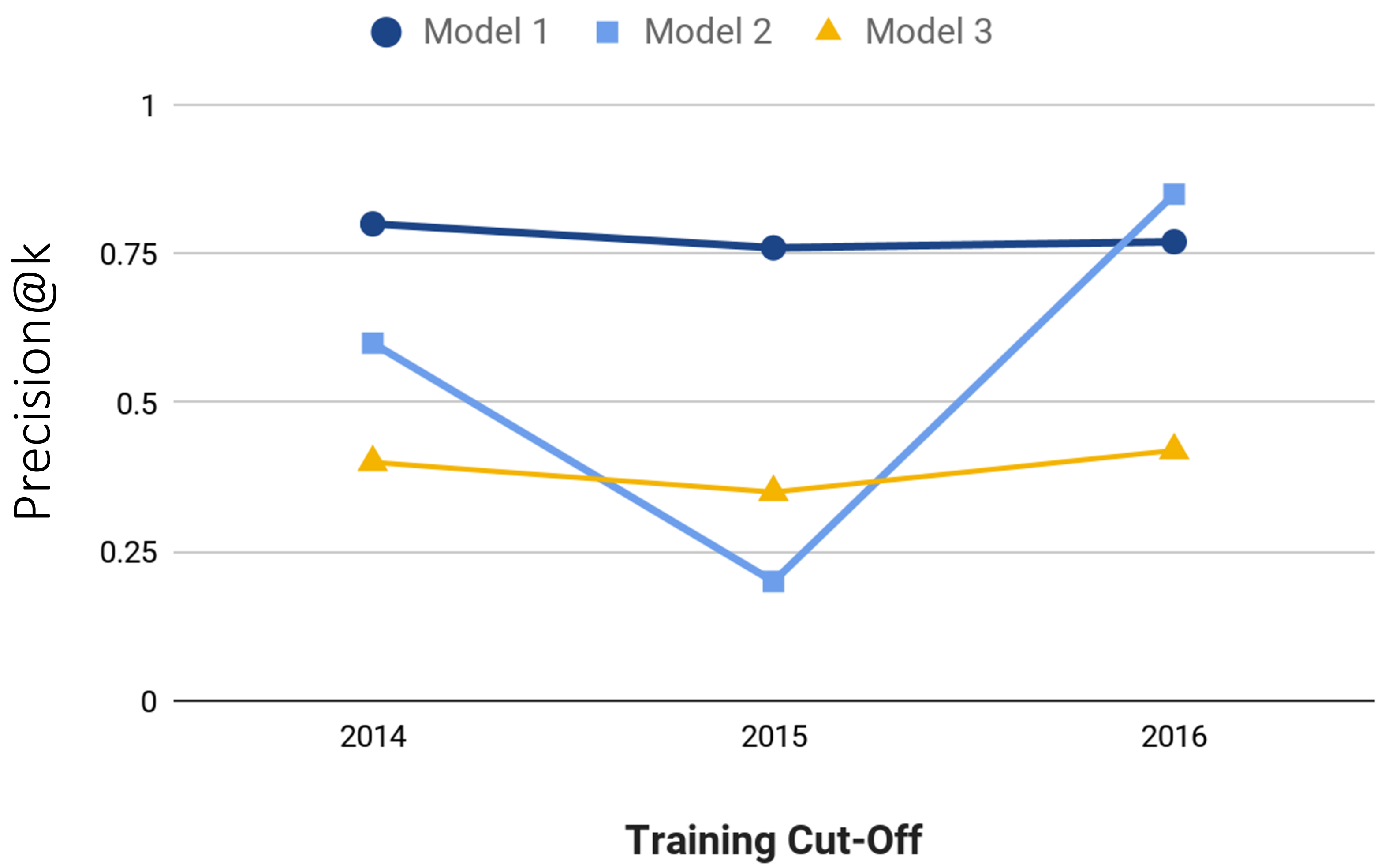
Model Selection



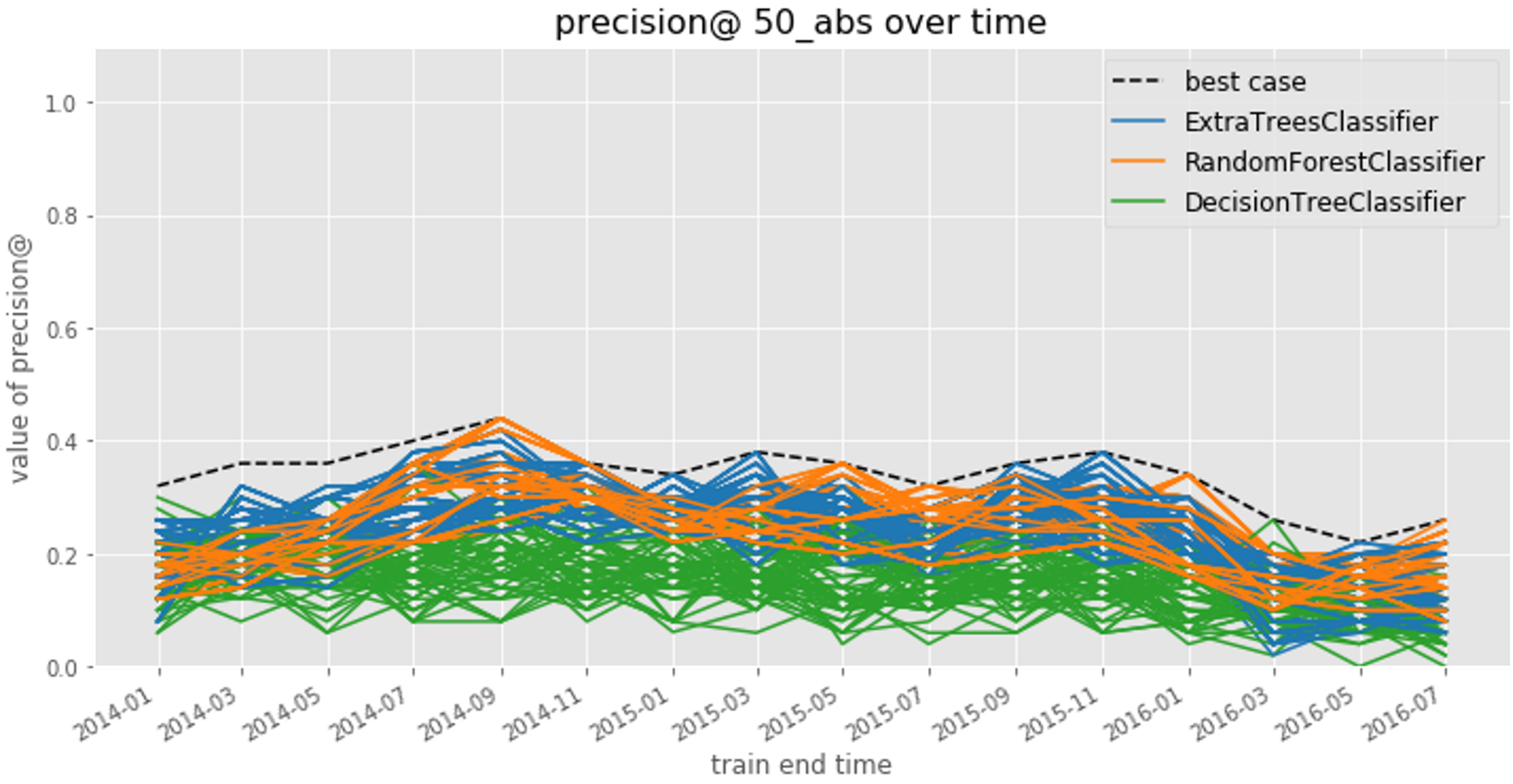
Model Selection



Model Selection



Model Selection



Model Selection

- How can you narrow hundreds or thousands of model specifications down to a handful of the best-performing ones?

- How do you balance performance and stability?
  - mean performance?
  - balancing mean and variance?
  - recency-weighted mean?

- What is the “regret” in subsequent time periods from using different strategies for choosing a model to deploy?

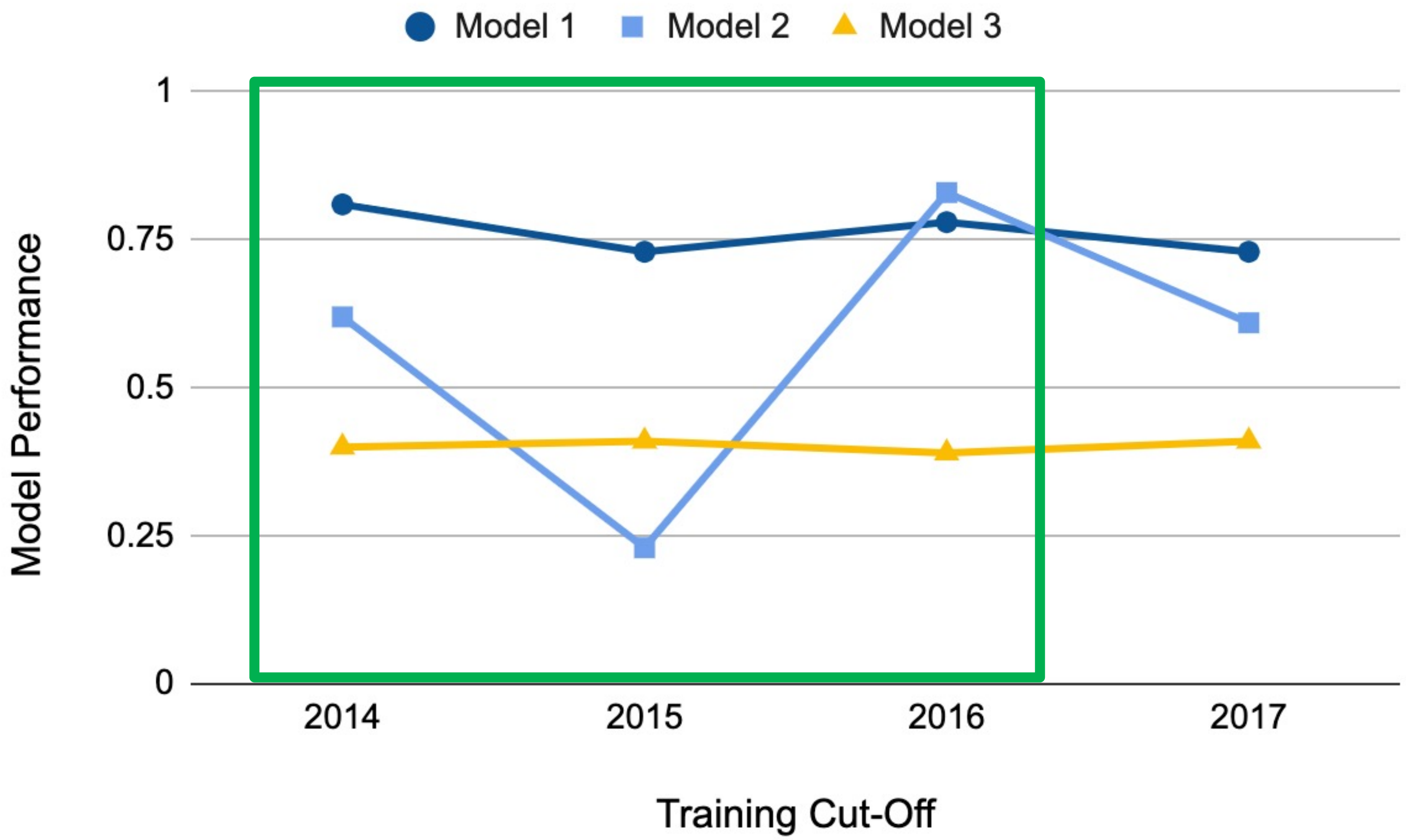
Model Selection

Comparing Model Selection Strategies:

- Use test set performances up to a given point in time to do model selection using each strategy

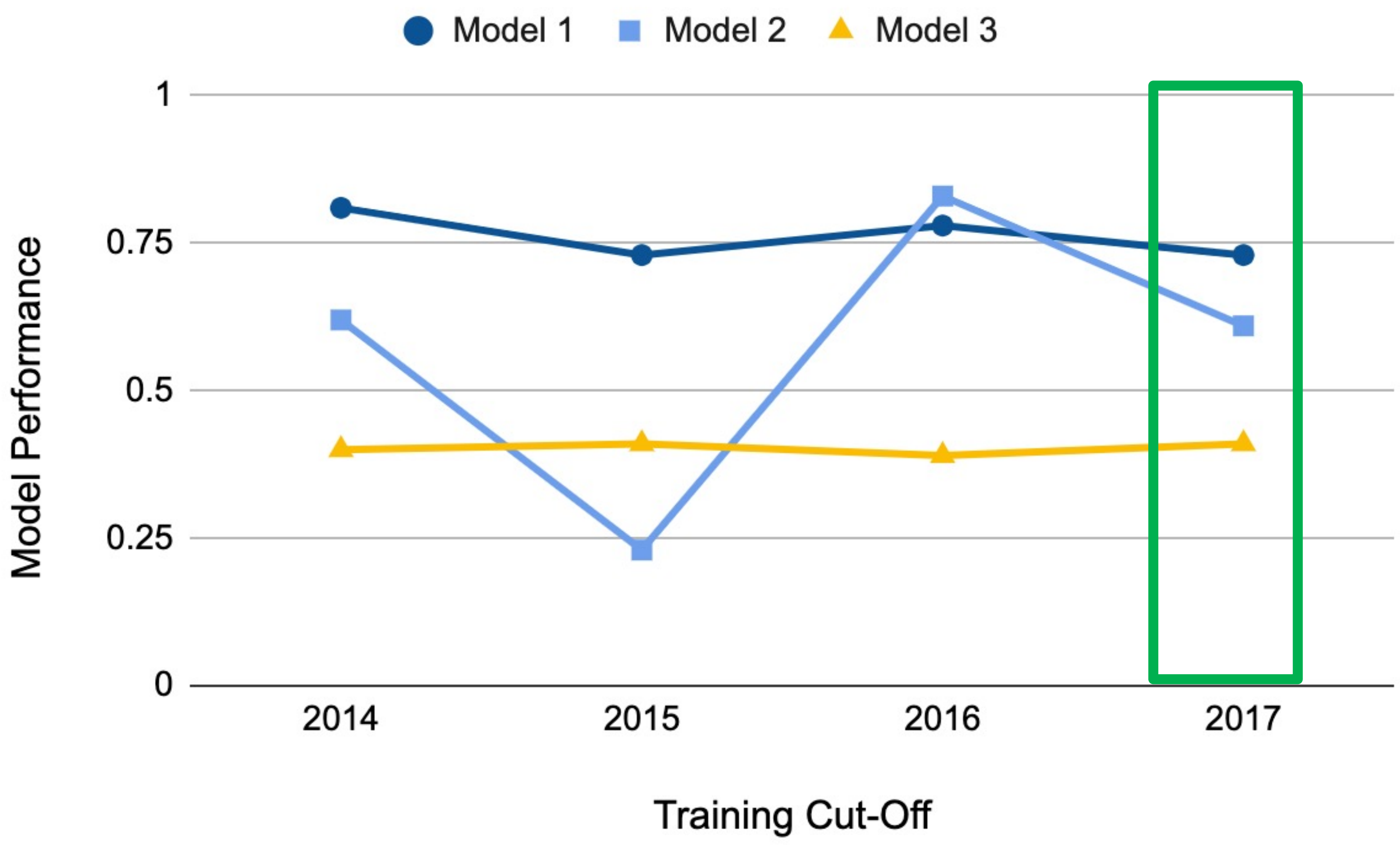
- On the subsequent validation set, calculate a **regret** for the model selection strategy as the difference between the performance of the model specification that strategy chooses and the best-performing model specification on this new validation set

Model Selection



Apply model selection strategies to validation set performance through 2016

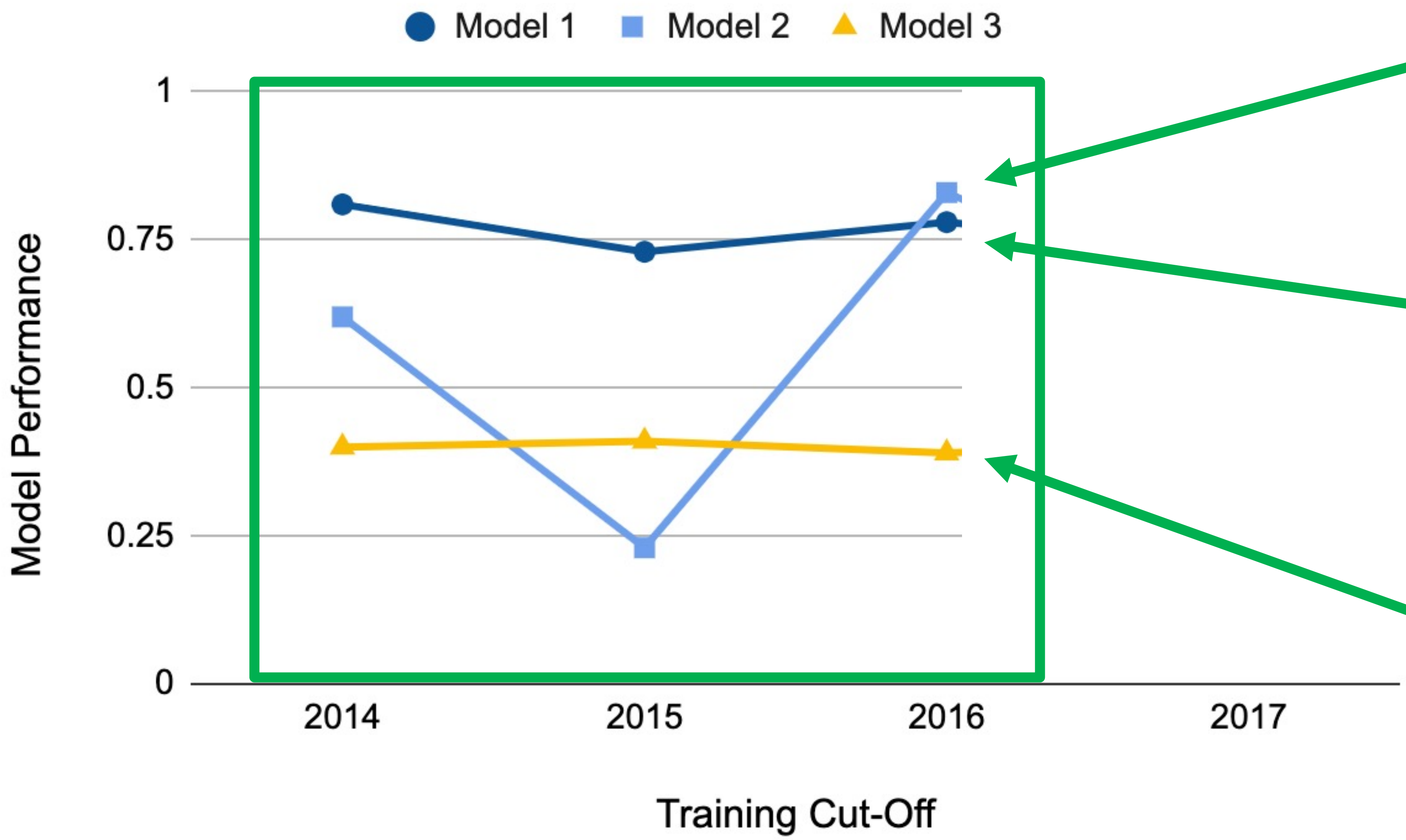
Model Selection



Apply model selection strategies to test set performance through 2016

Calculate regret based on 2017 validation set performance

Model Selection

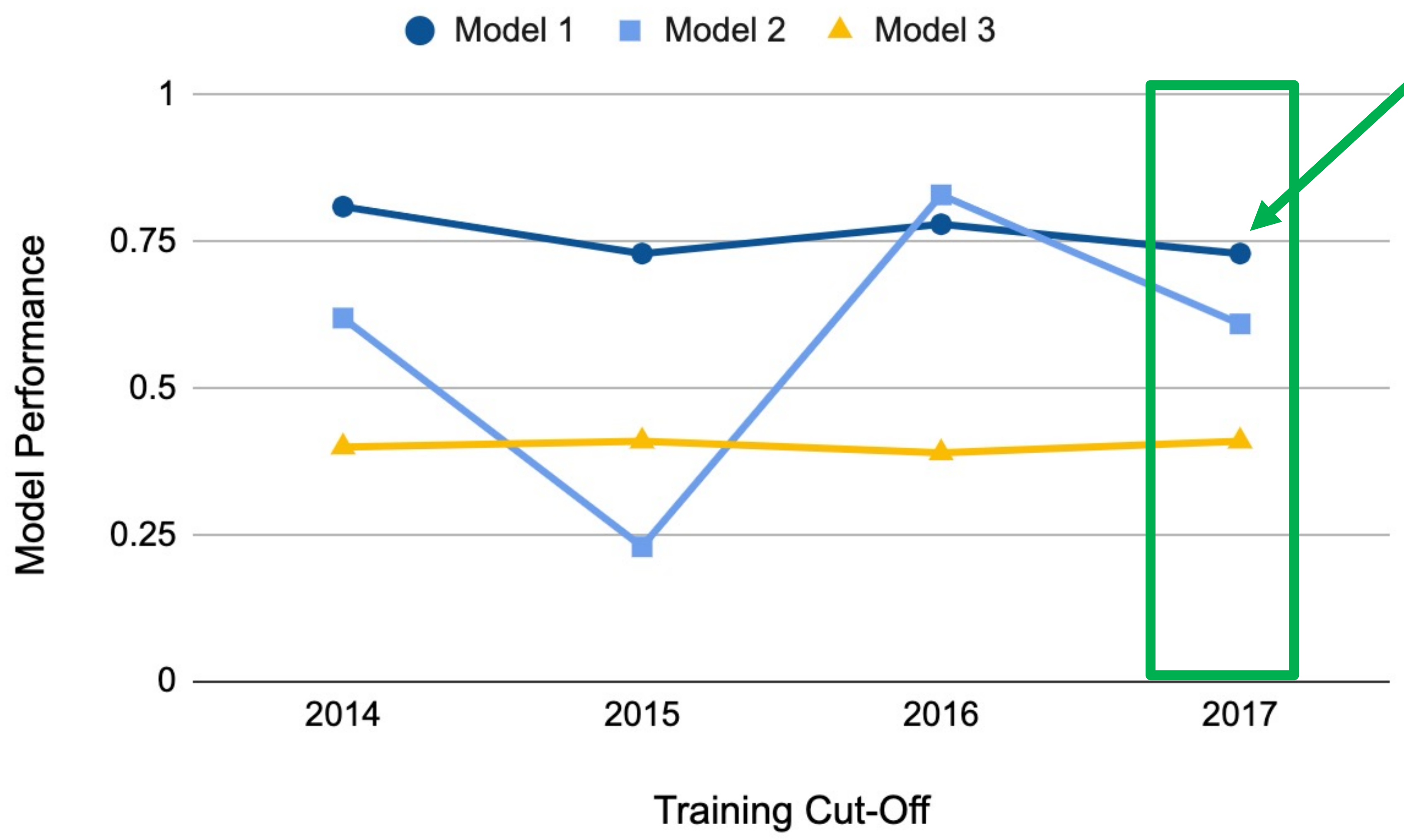


Best Recent Perf:
Model 2

Best Average Perf:
Model 1

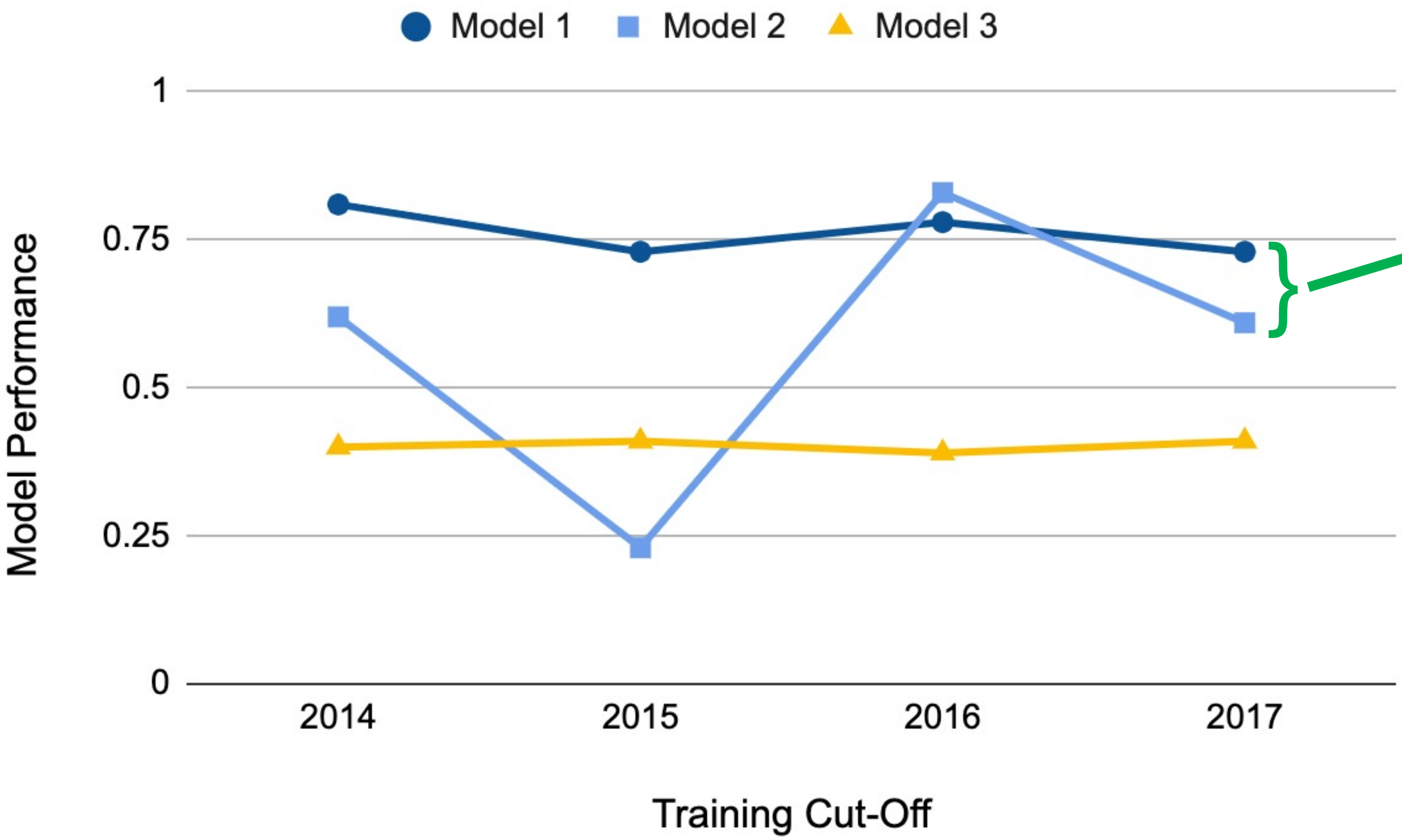
Lowest Variance:
Model 3

Model Selection



Best Model in 2017:
Model 1
Perf = 0.73

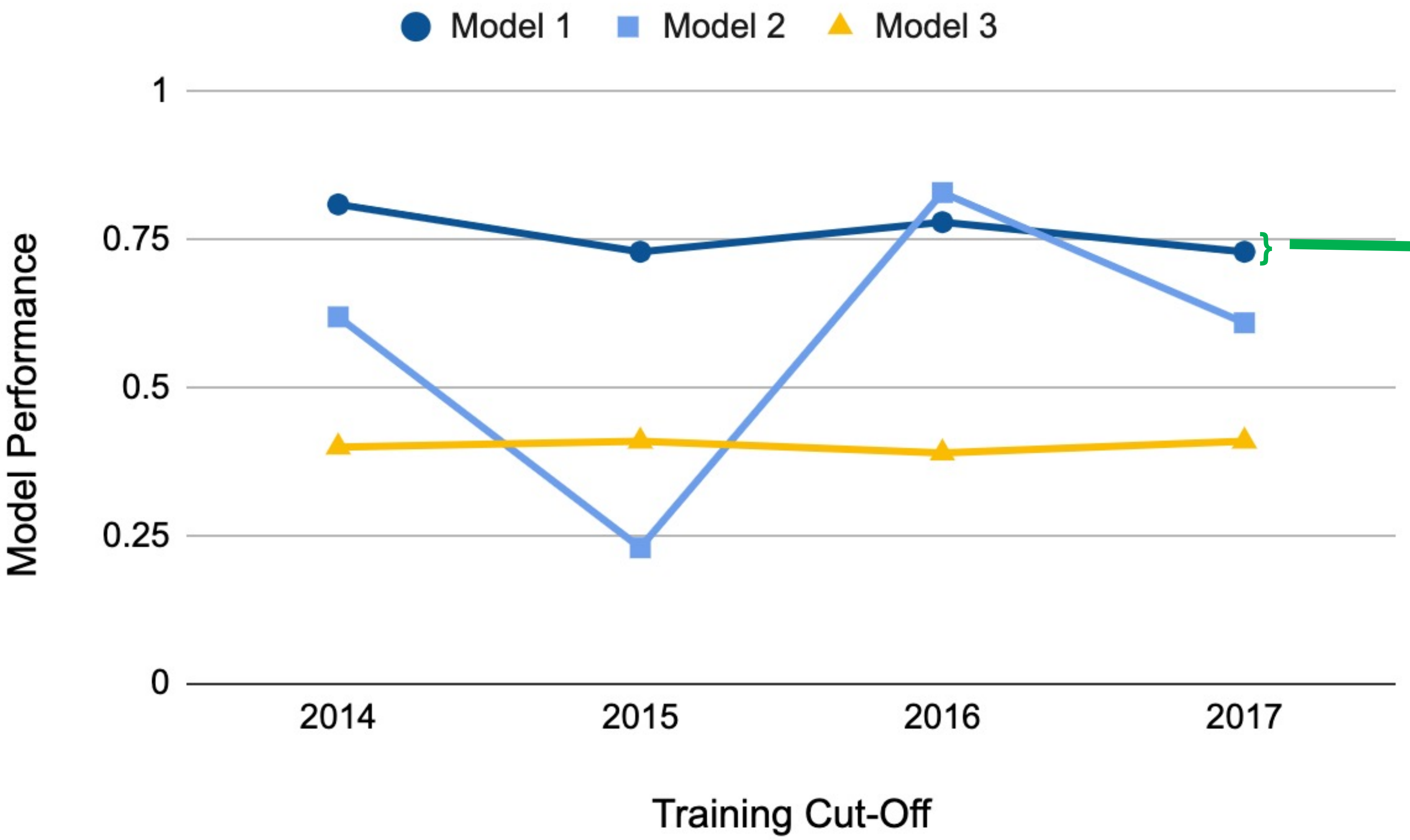
Model Selection



Best Recent Perf:
Model 2

Regret:
0.73 – 0.61 = 0.12

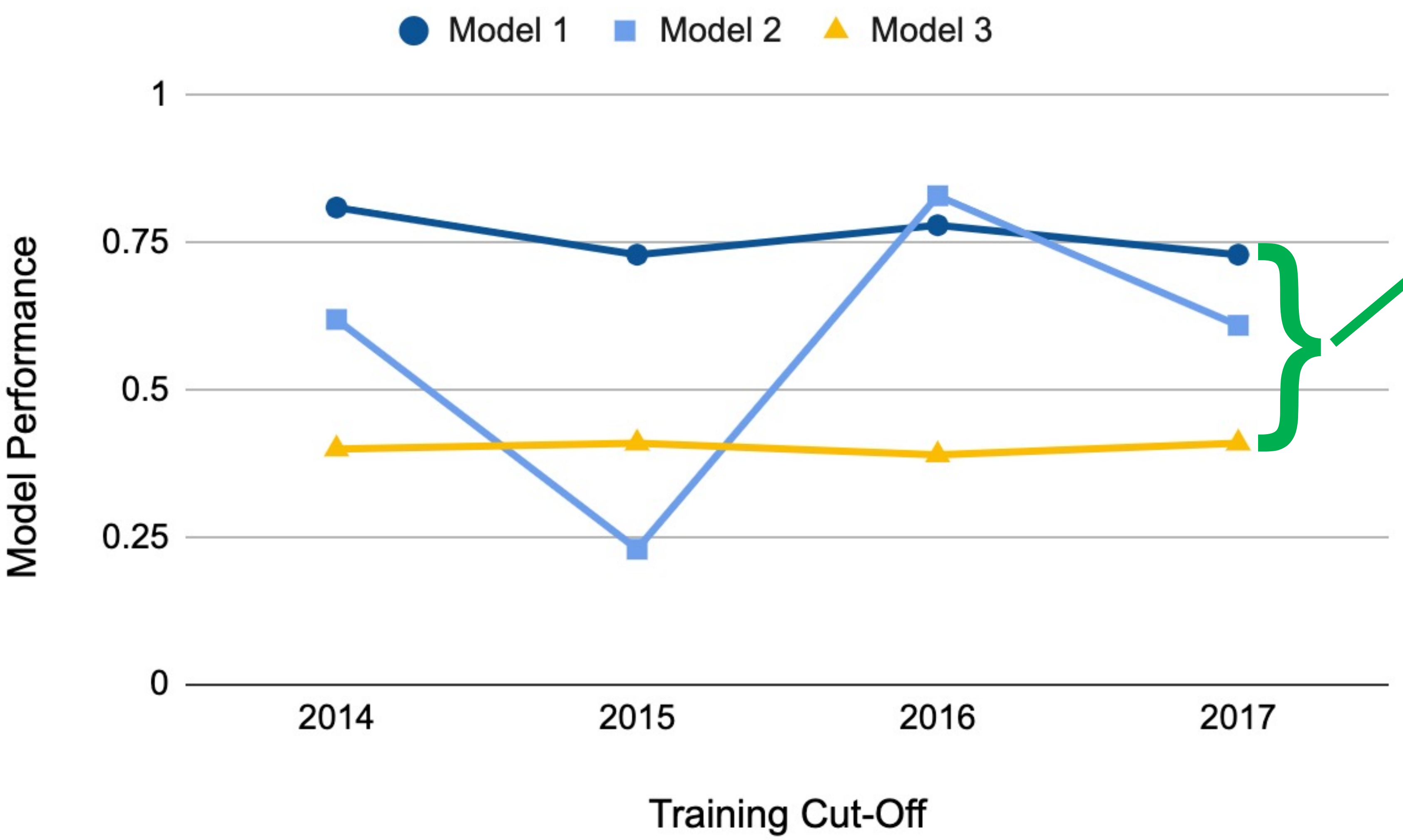
Model Selection



Best Average Perf:
Model 1

Regret:
0.73 – 0.73 = 0.00

Model Selection



Lowest Variance:
Model 3

Regret:
0.73 – 0.41 = 0.32

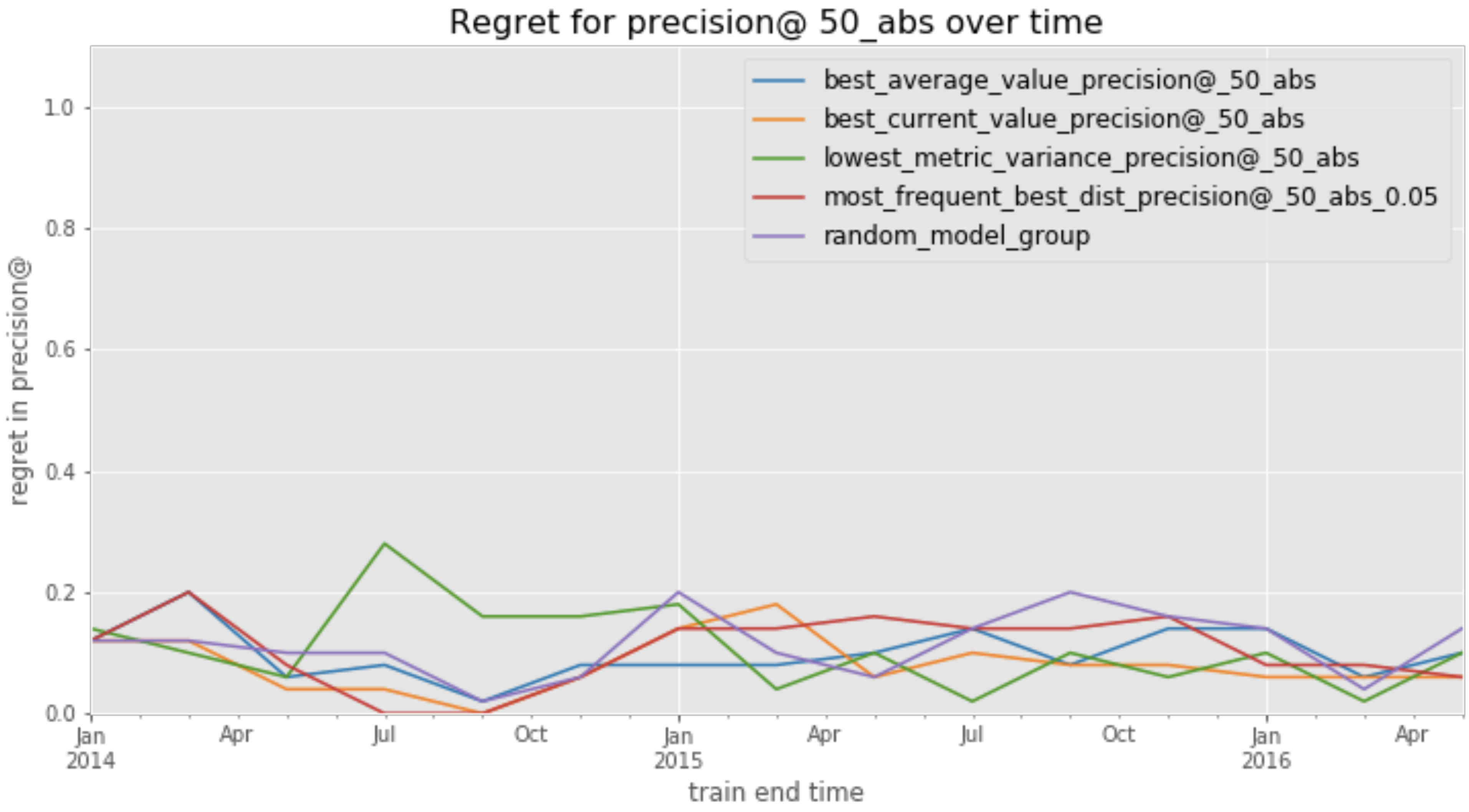
Model Selection

Strategy	Best Model Through 2016	Regret in 2017
Best Recent Perf.	Model 2	0.12
Best Average Perf.	Model 1	0.00
Lowest Variance	Model 3	0.32

Model Selection

Strategy	Best Model Through 2016	Regret in 2017
Best Recent Perf.	Model 2	0.12
Best Average Perf.	Model 1	0.00
Lowest Variance	Model 3	0.32

Model Selection



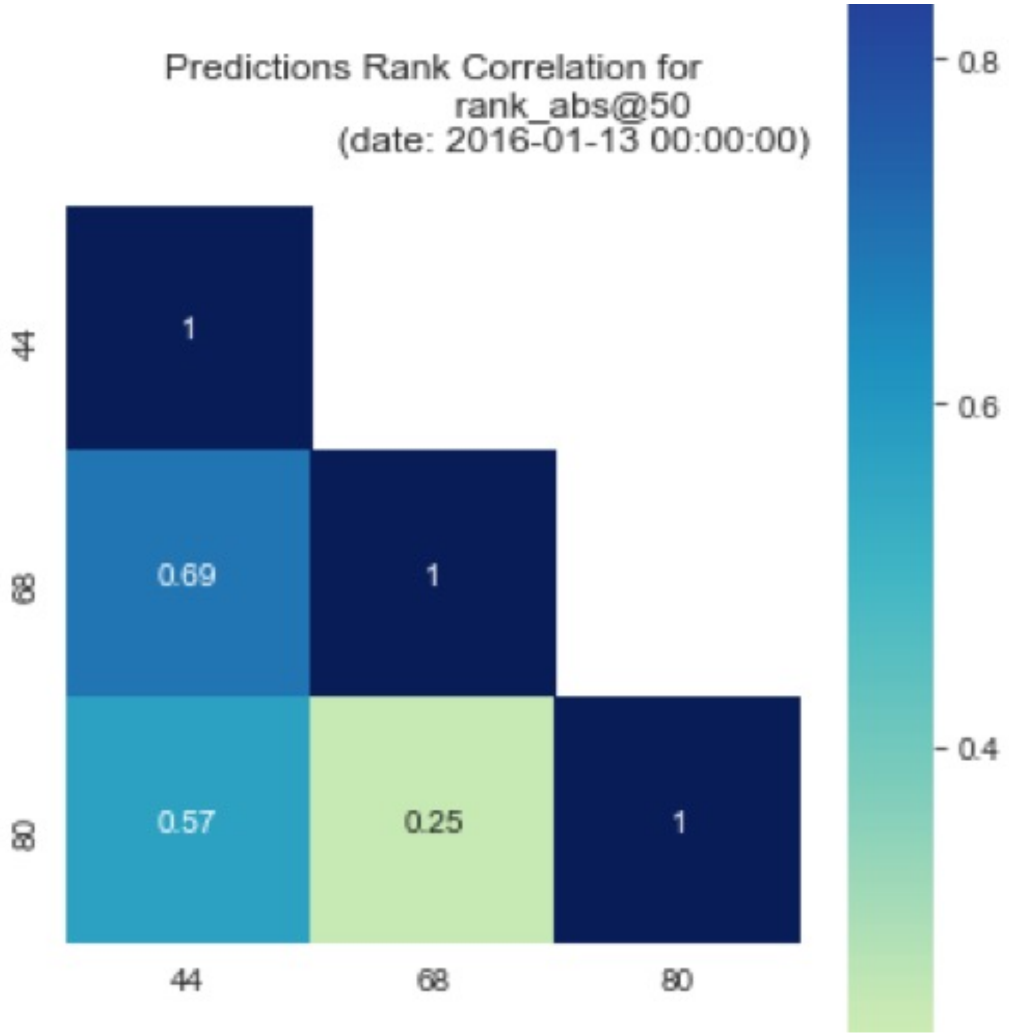
Model Selection

Model Selection

- May not be obvious which strategy / model specification is "best"

- Among good candidates, may be instructive to ask how similar or different the lists each strategy would produce are

- May ultimately want to deploy (or at least test) a strategy that combines across several specifications



Some Open Research Questions

- What are the conditions under which temporal validation out-performs traditional cross-validation? By how much?
- Likewise, what can we learn about how well certain strategies perform in terms of regret under different real-world conditions?
- Many problems in policy settings involve resource constraints that require optimization at the top of the list, but few methods optimize for this directly.
    - e.g., Transductive Top k

Some Open Research Questions

Transductive Optimization of Top k Precision

Li-Ping Liu **Thomas G. Dietterich**
EECS, Oregon State University
Corvallis, OR 97330, USA
{liuli@eecs.oregonstate.edu, tgd@oregonstate.edu }

Nan Li **Zhi-Hua Zhou**
Department of Computer Science & Technology, Nanjing University
Nanjing 210023, China
{lin, zhouzh}@lamda.nju.edu.cn

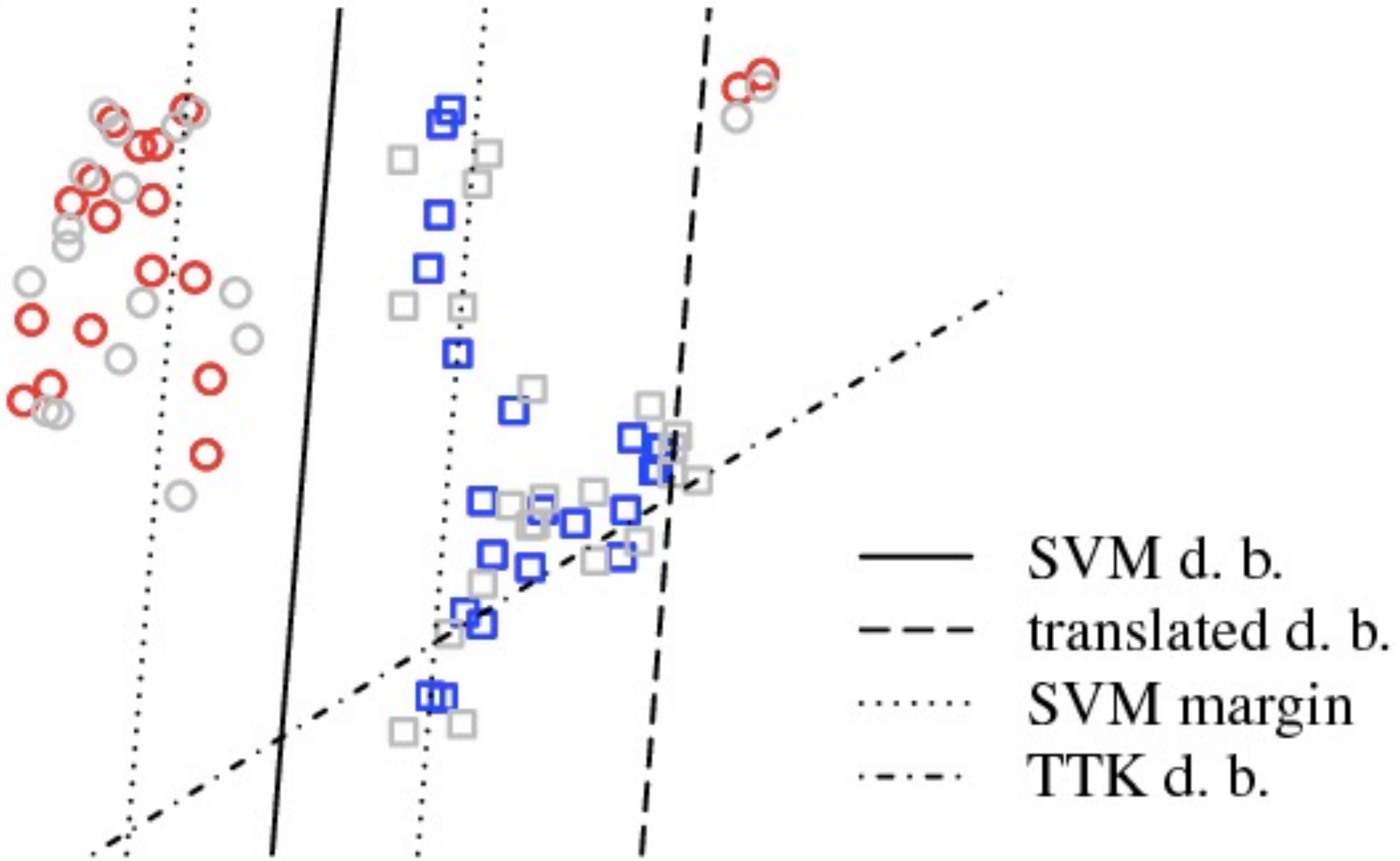
Some Open Research Questions

- The SVM loss function will find the “best” separating hyperplane overall, but perhaps we could draw a better hyperplane to separate just *k* positive examples?

- *Transductive* method: needs to be aware of the test set **without labels** to select just *k* test examples.

- Modified gradient descent procedure to project gradient direction for L2-regularized SVM loss onto a “feasible solution cone” such that <u>no more than</u> *k* test examples will be predicted positive after the step.

Some Open Research Questions



Some Open Research Questions

Paper shows improvements on synthetic examples and some "standard" datasets, but still more to investigate:

- Can be slow to converge on larger datasets
- "At most" *k* examples can yield many fewer than the desired *k*, particularly for rare events (why doesn't the algorithm target *exactly k*?)
- Although creating a "top k" boundary, still penalizes false positives and false negatives equally during optimization
- Can we do better at the top, even if we don't have access to the test list?

Things to remember

Coming Up Next Week:

- Monday: Modeling Results Update Assignment (posted on canvas)
- Tuesday: Weekly Feedback Form and Readings
- Midterm:
  - Take Home
  - No class on Thursday
  - Due Friday (Canvas)