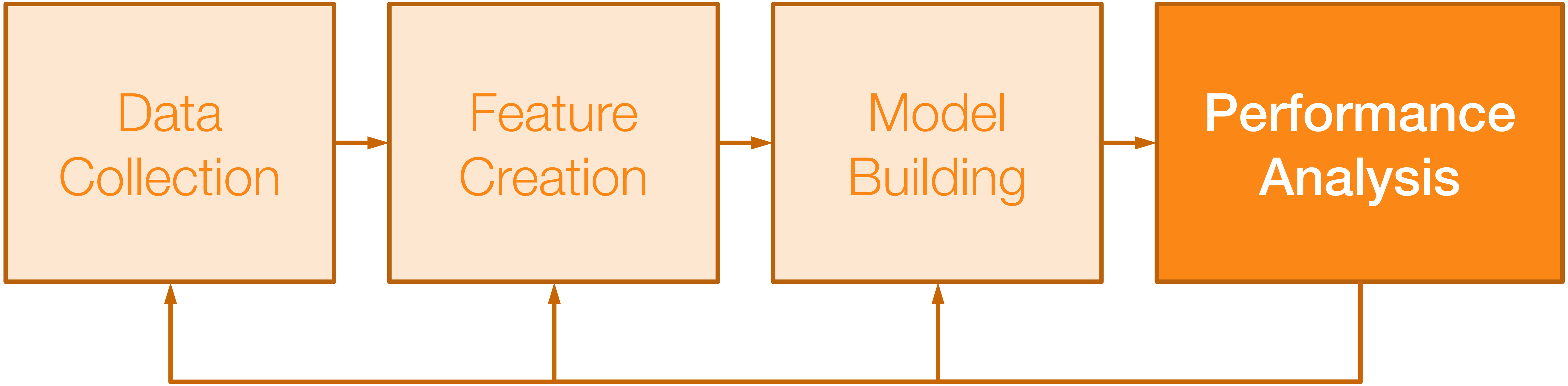
Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers

Donghao Ren[1,2], Saleema Amershi[2], Bongshin Lee[2], Jina Suh[2] and Jason D. Williams[2]

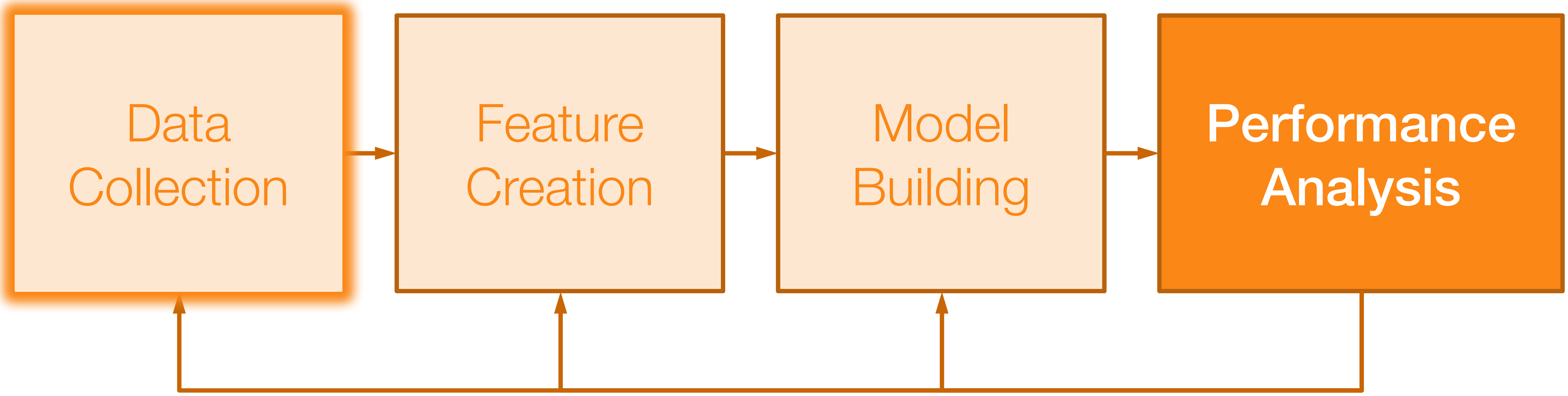
[1] University of California, Santa Barbara

[2] Microsoft Research, Redmond

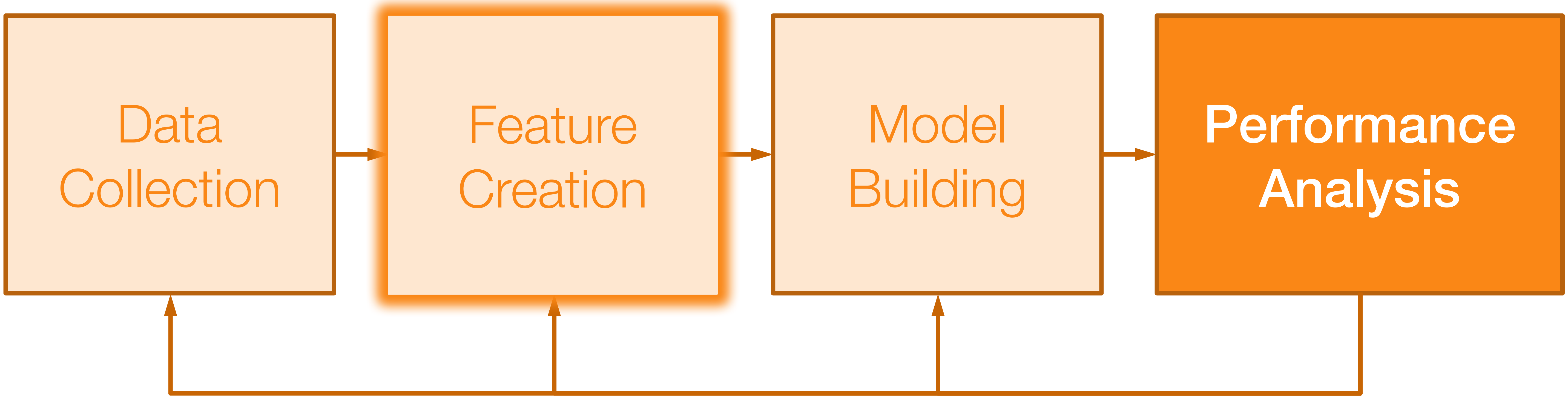
Performance analysis is critical in machine learning



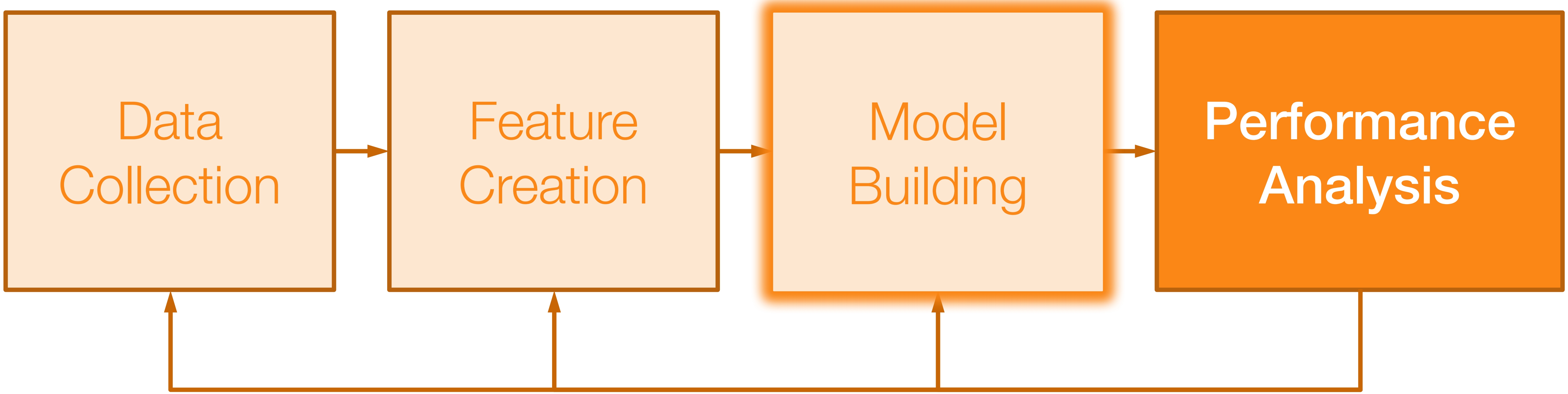
Performance analysis is critical in machine learning



Performance analysis is critical in machine learning



Performance analysis is critical in machine learning



Common ways of performance analysis

- Summary statistics
  - Accuracy
  - Precision
  - Recall
  - Log-Loss
  - …
- Confusion Matrix

Predicted Class

Actual Class

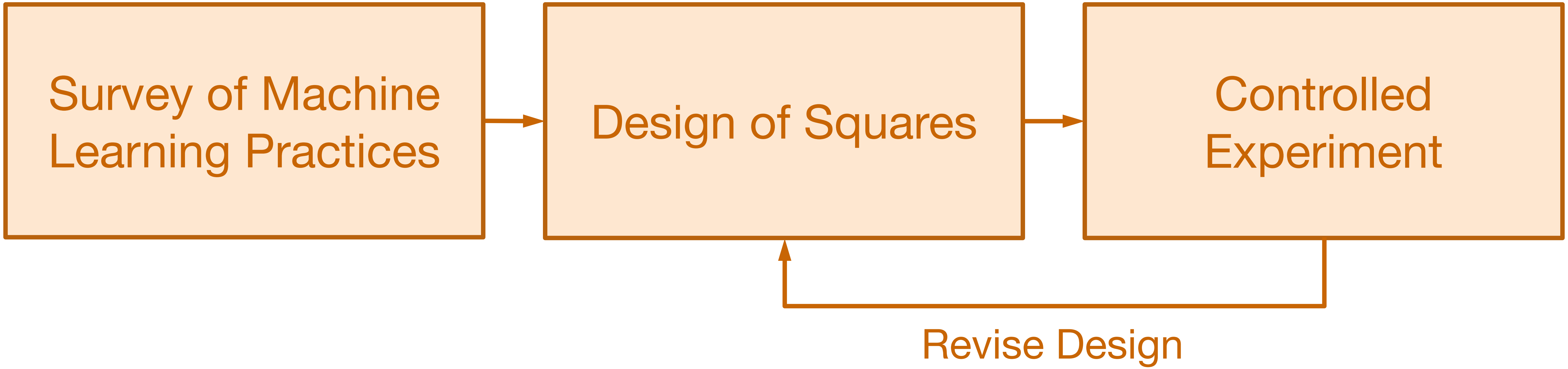
89.3%	0.6%	6.9%	3.3%
31.4%	42.0%	19.4%	7.2%
18.6%	0.4%	79.8%	1.2%
16.3%	1.1%	2.4%	80.2%

Problems

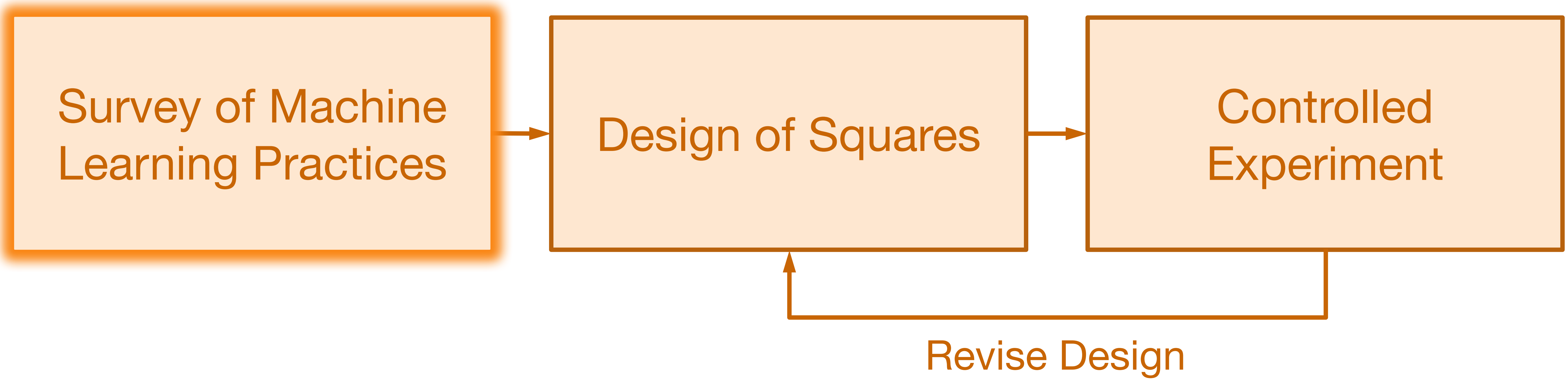
- Disconnected from the underlying data.
- Hide important information such as score distribution.
- Not trivial to support ***multiclass*** classifiers.

Squares

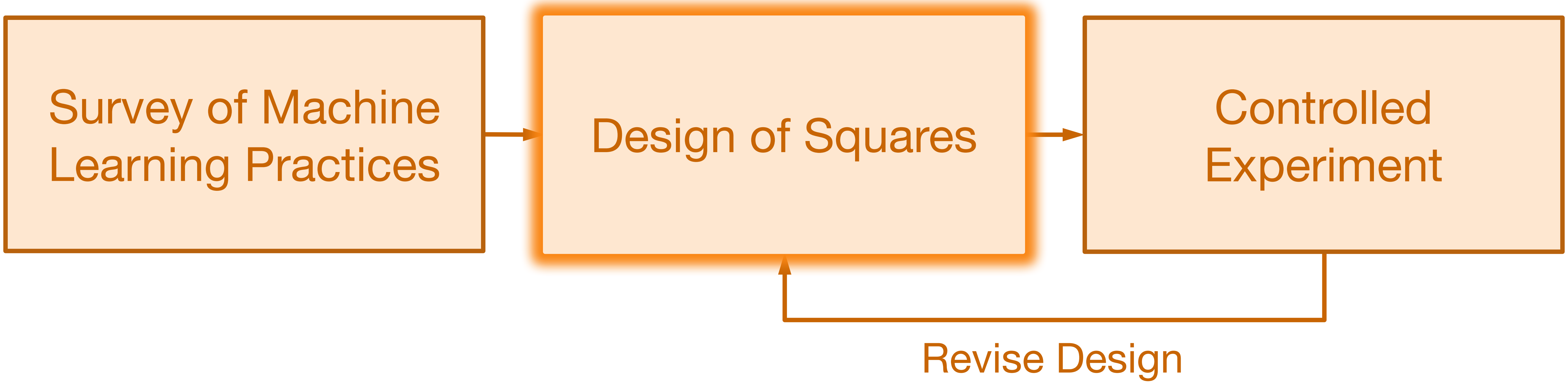
Design Process



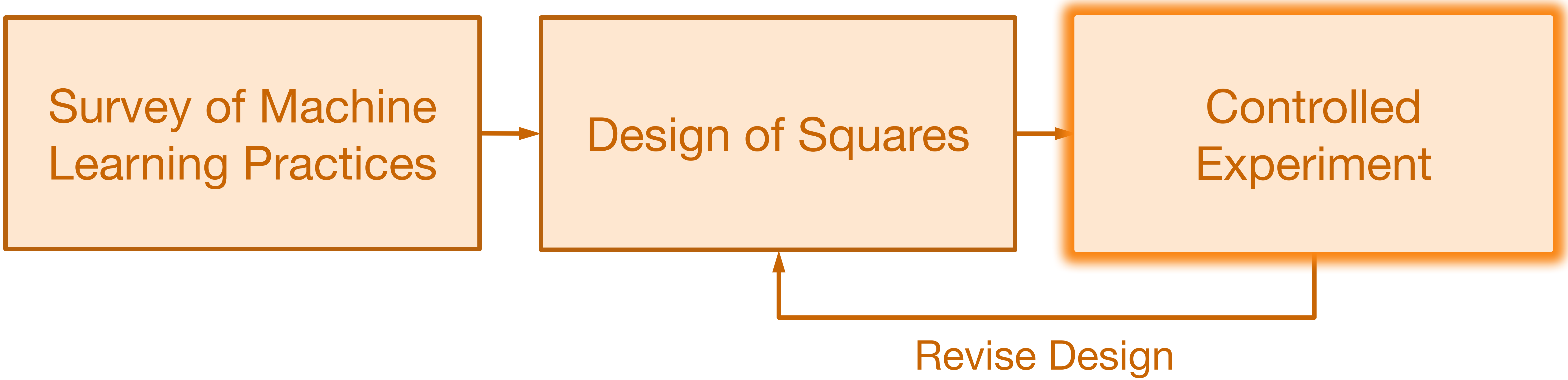
Design Process



Design Process



Design Process

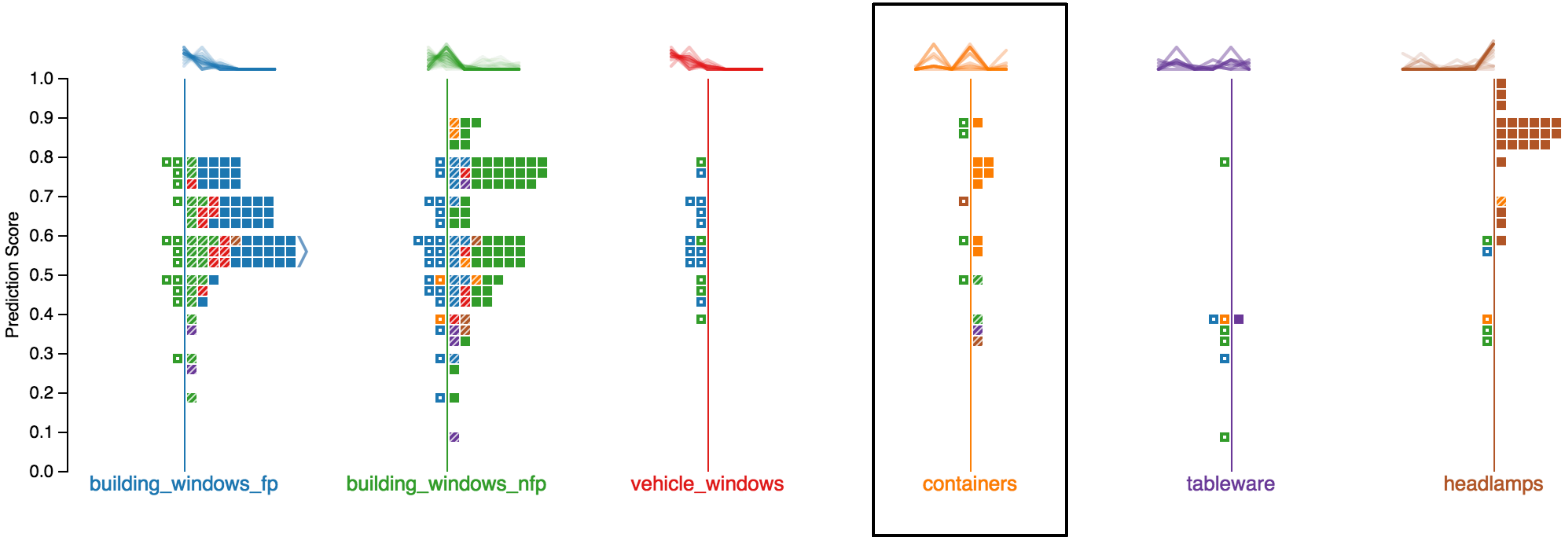


Design Goals

- G1: Show performance at multiple levels of detail to help practitioners prioritize efforts.
  - Overall / Class-level / Instance-level
  - Error severity (errors with higher score on the wrong class are more severe)
- G2: Be agnostic to common performance metrics.
  - Support a wider range of scenarios.
- G3: Connect performance to data.
  - Provide access to data. Use small visual footprint to reserve space for scenario-dependent data access views.

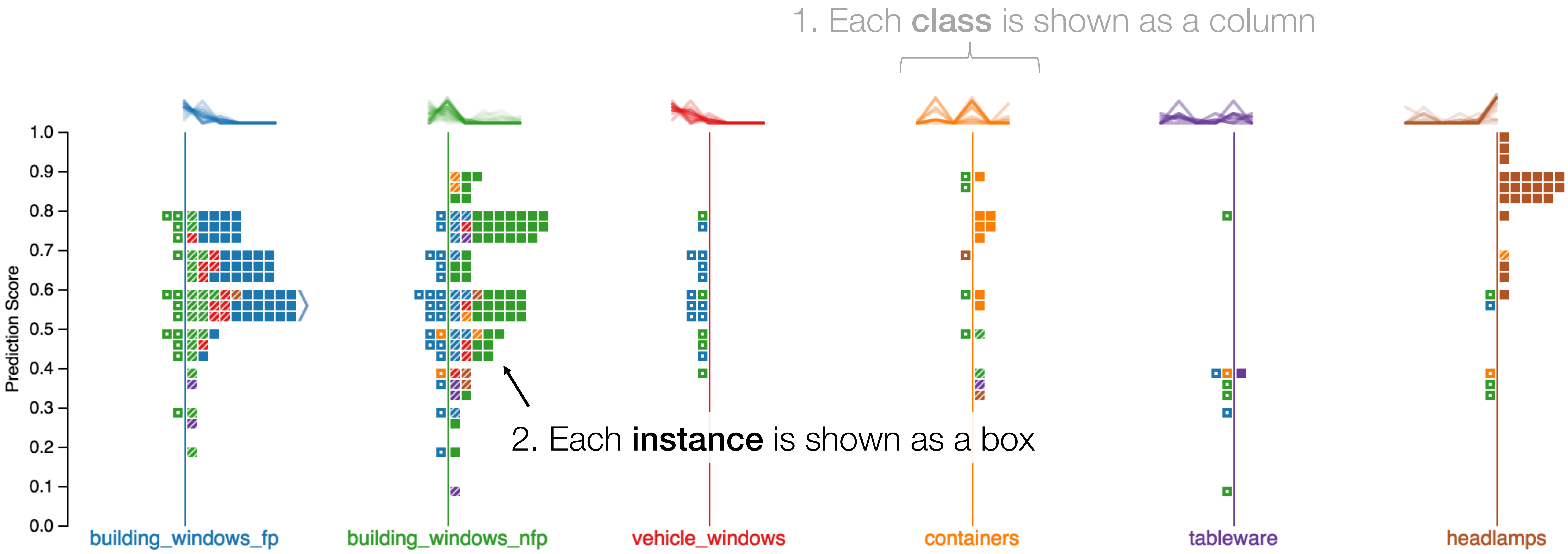
Squares Visualization Design

1. Each **class** is shown as a column



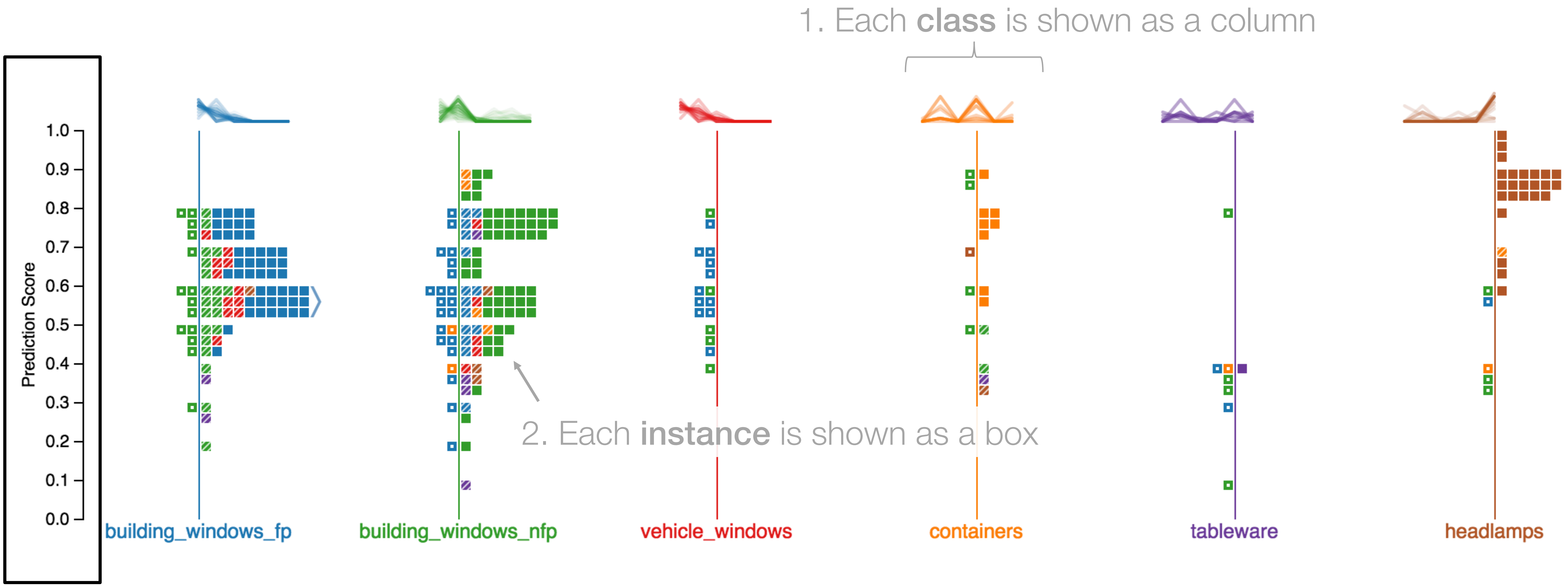
Dataset: **Glasses** from the UCI Machine Learning Repository

Visualization Design



Dataset: **Glasses** from the UCI Machine Learning Repository

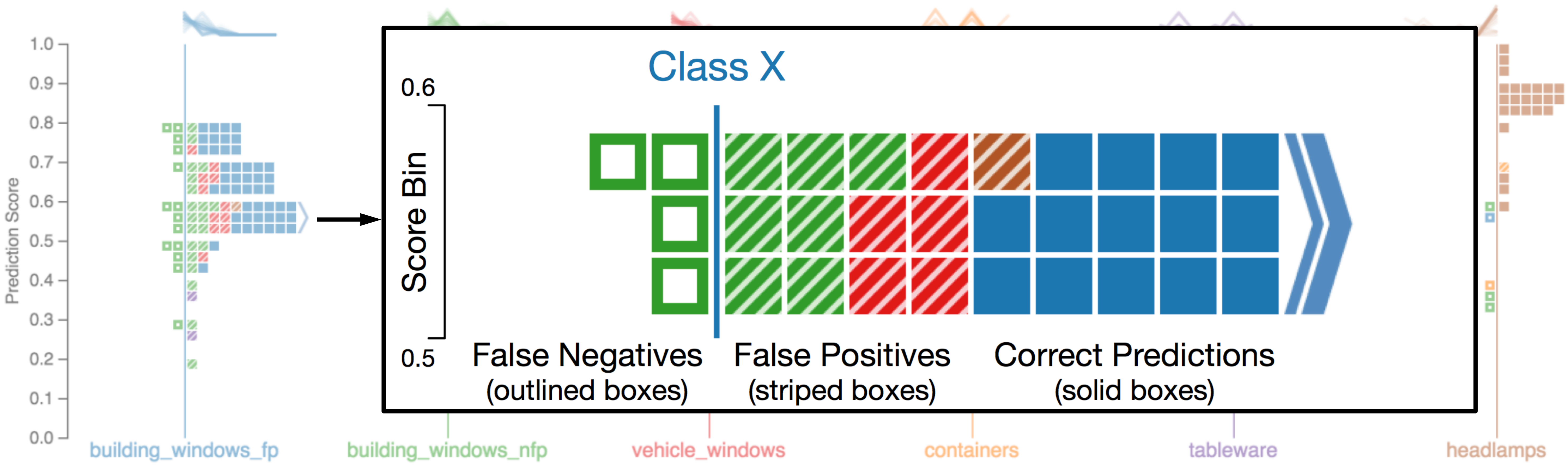
Visualization Design



3. Instances are binned according to **prediction scores**

Dataset: **Glasses** from the UCI Machine Learning Repository

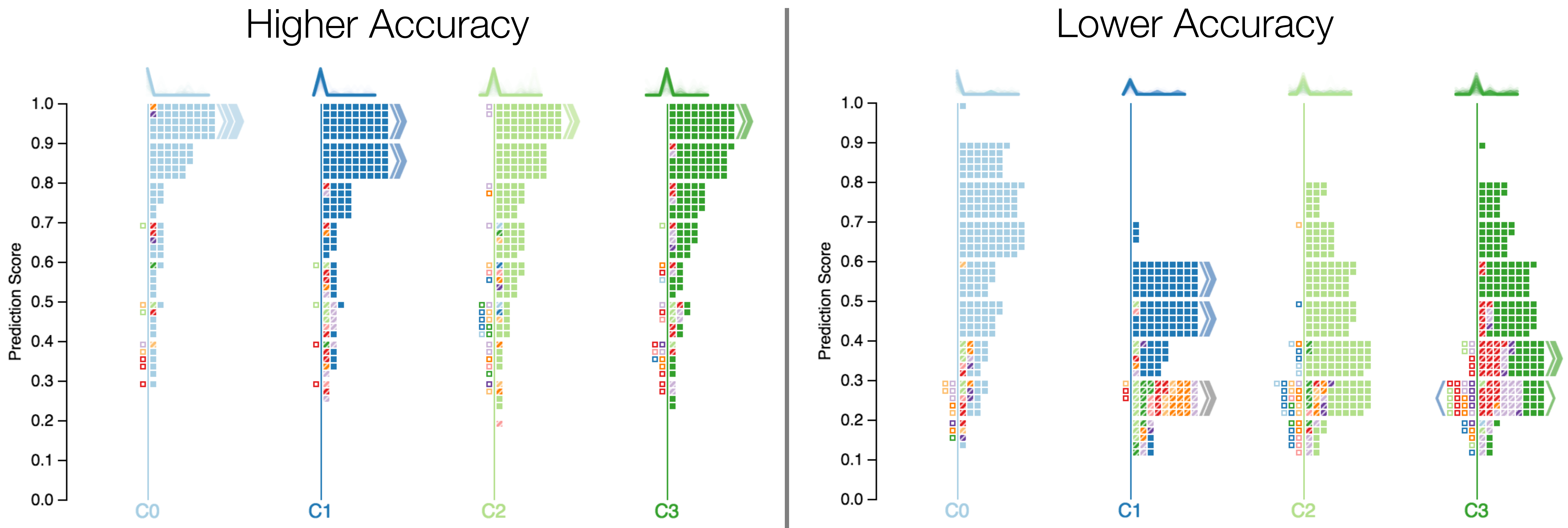
Visualization Design



Dataset: **Glasses** from the UCI Machine Learning Repository

Visualizing Count-Based Metrics: Overall Accuracy

- Accuracy: $\dfrac{\text{Correct Predictions}}{\text{Total \# of Instances}} = \dfrac{\blacksquare}{\blacksquare + ▨}$



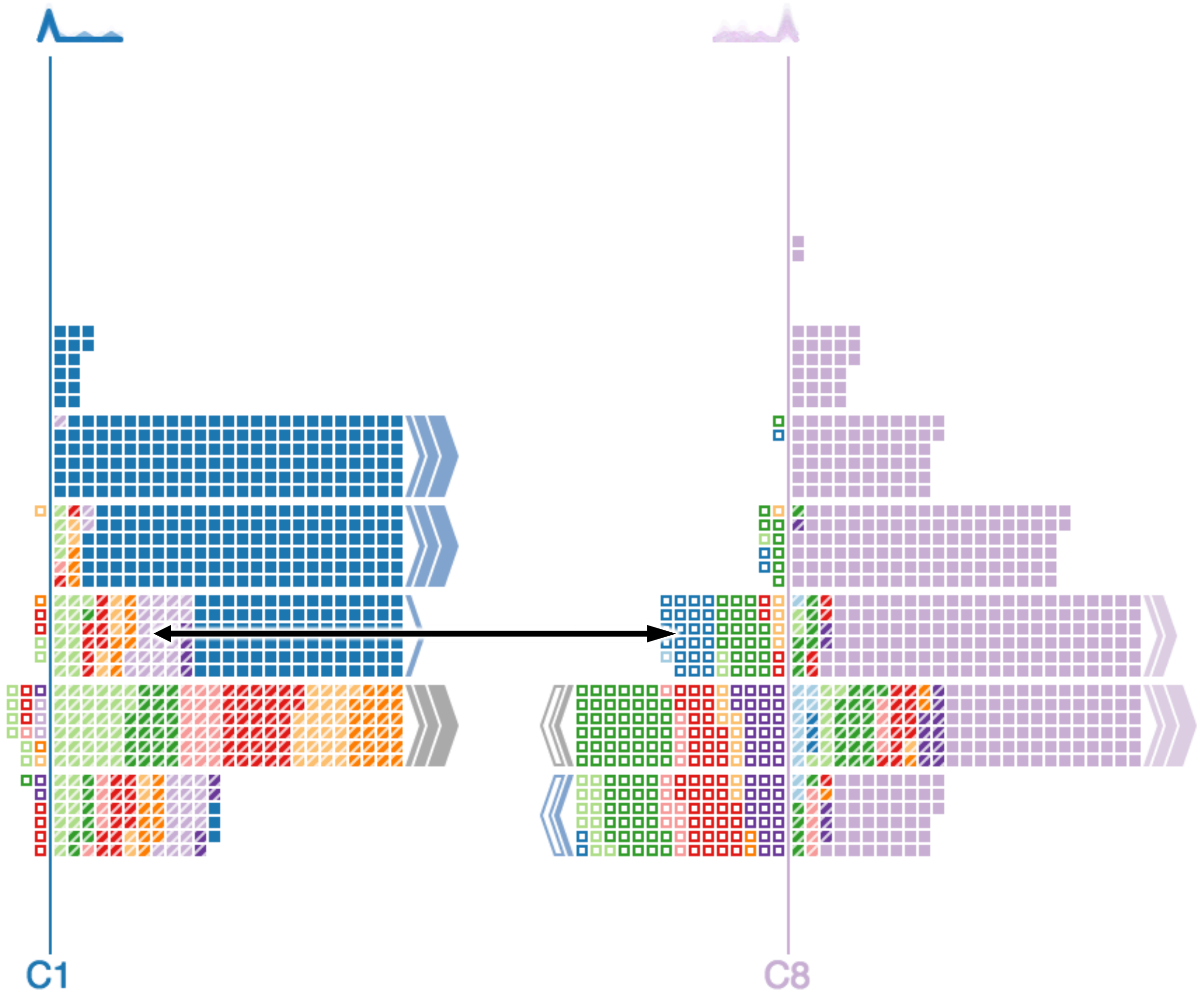
Visualizing Count-Based Metrics: Class-Level

- Class-level precision and recall:

Precision: Recall:

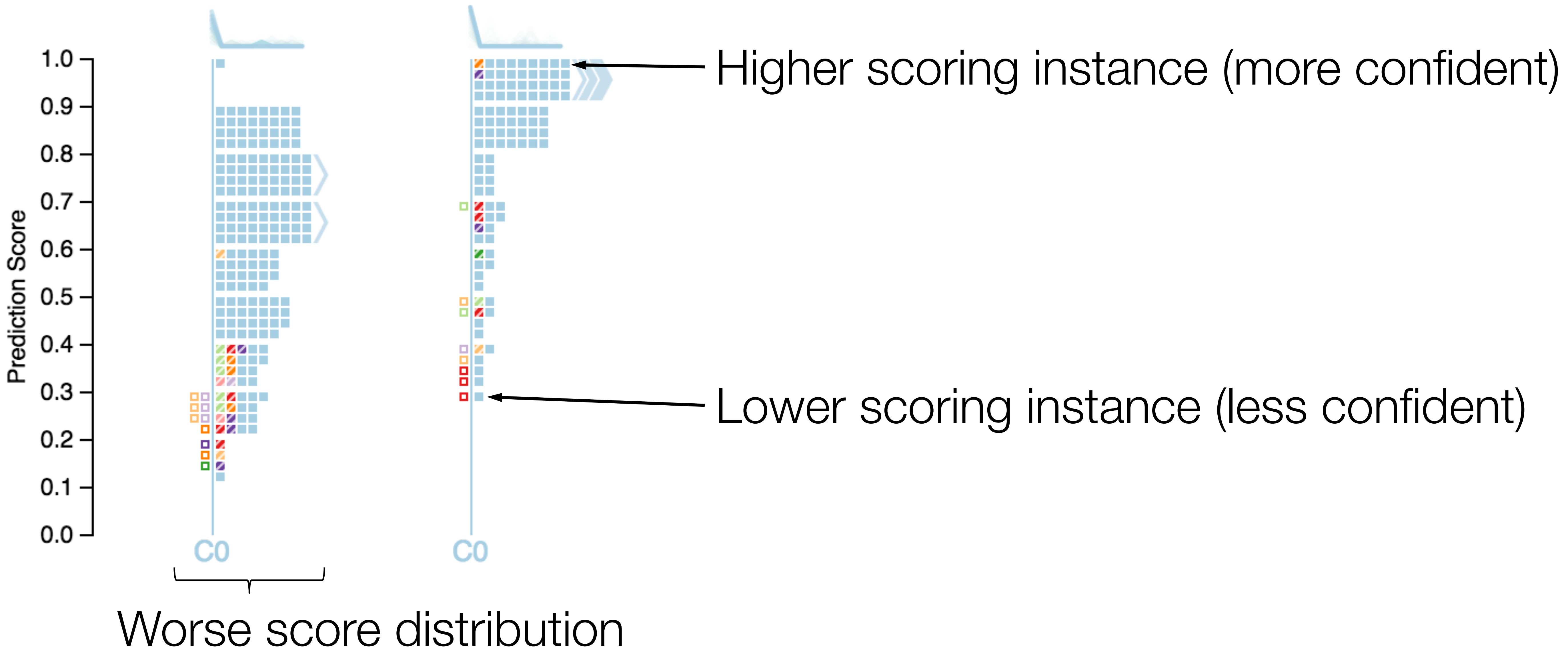
$$\frac{TP}{TP + FP} = \frac{\blacksquare}{\blacksquare + \boxtimes} \qquad \frac{TP}{TP + FN} = \frac{\blacksquare}{\blacksquare + \square}$$

FPs and FNs are comparably salient:
One-to-one correspondence between outlined boxes and striped boxes

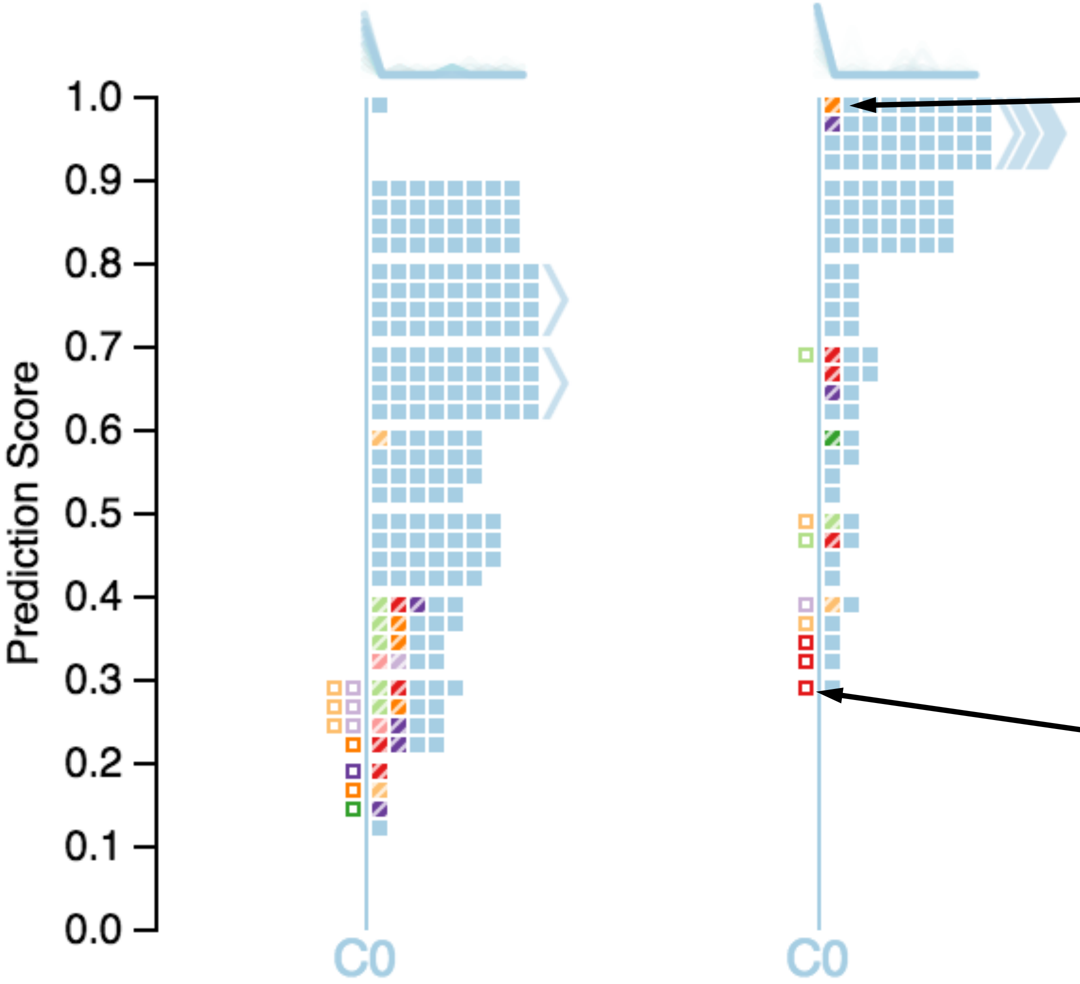


Lower Precision Lower Recall

Visualizing Score-Based Metrics



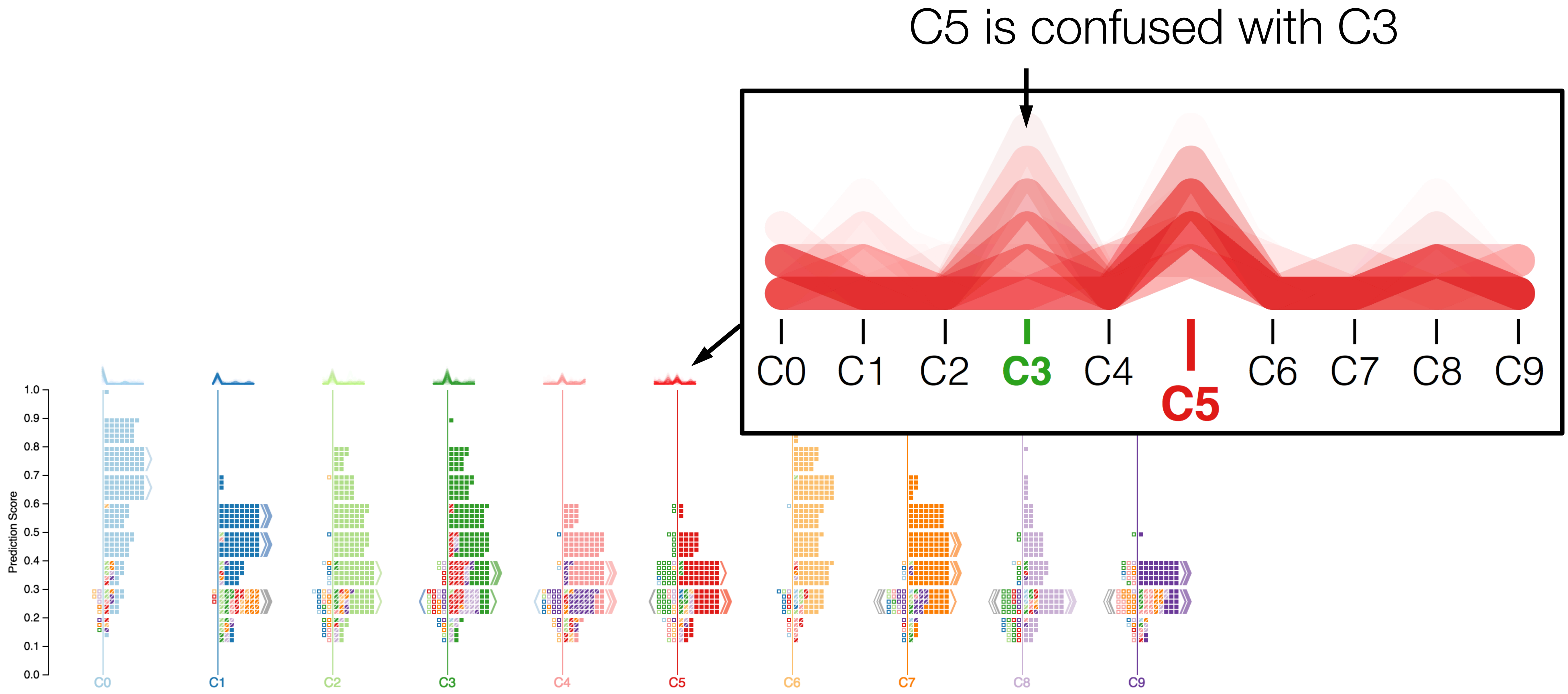
Help Prioritizing Debugging Efforts



More severe error (confidently wrong)

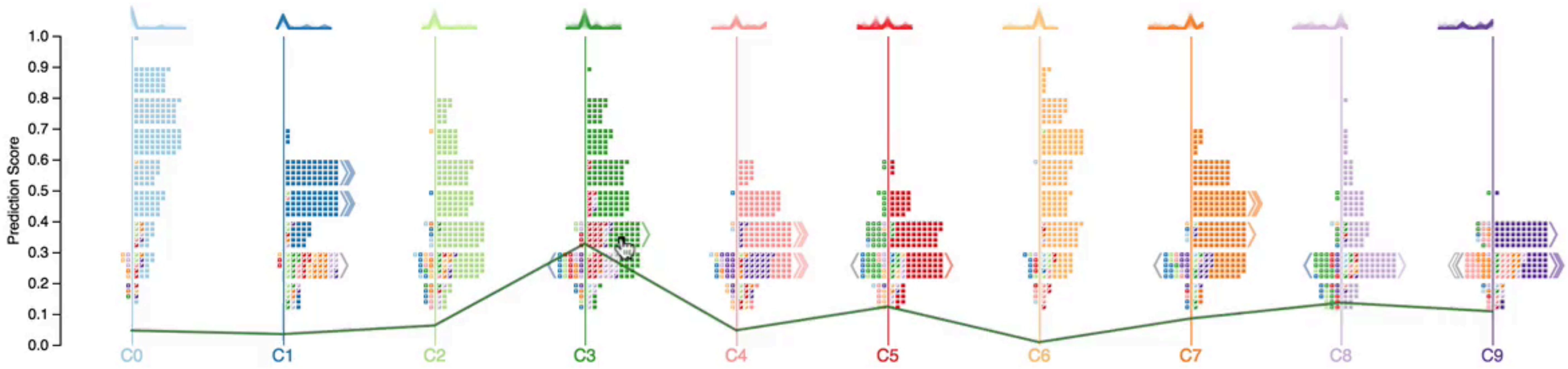
Less severe error (prediction can flip if scores change slightly)

Visualizing Confusion Between Classes



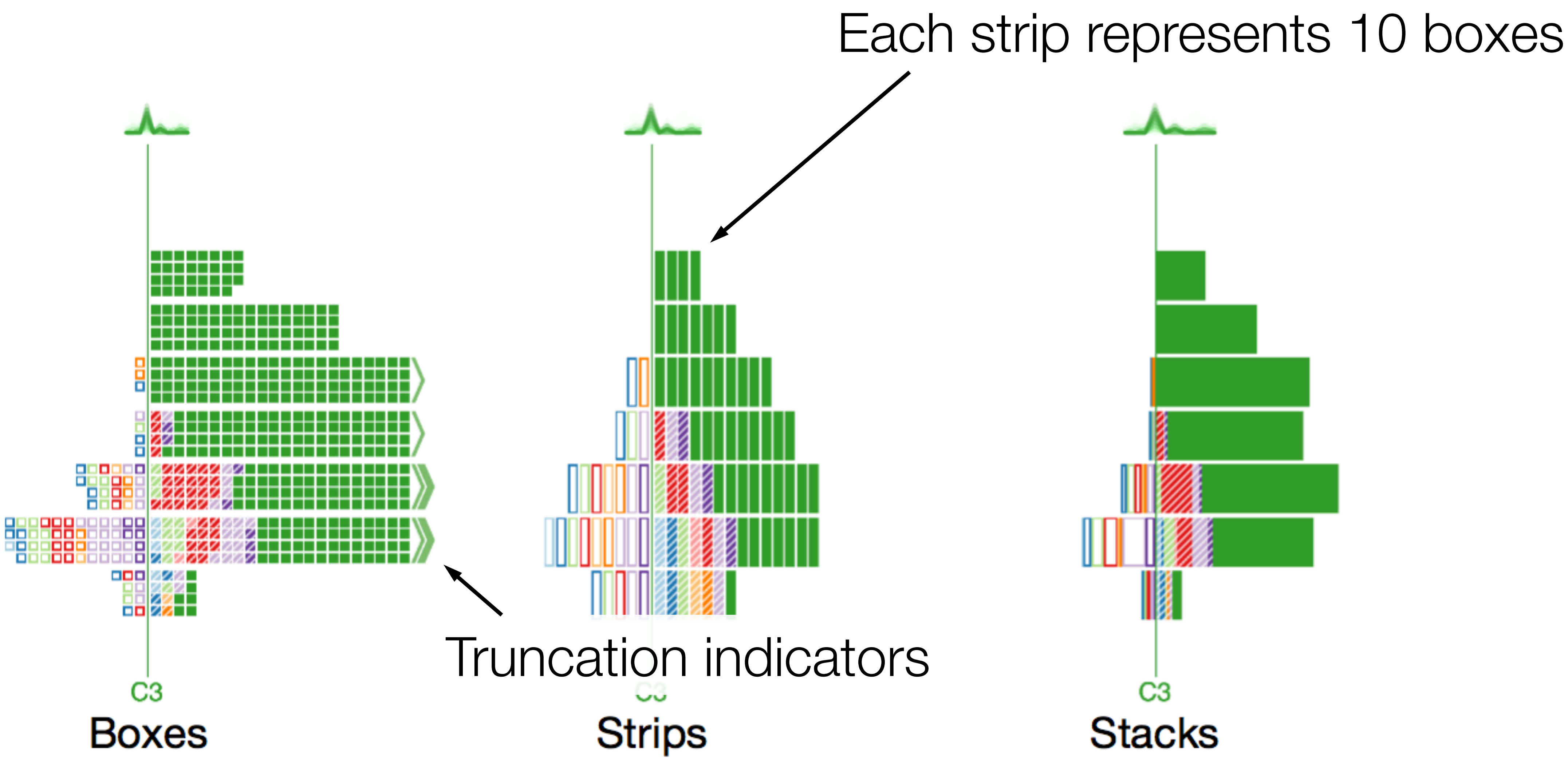
Dataset: **MNIST Handwritten Digits**

Instance-Level Details

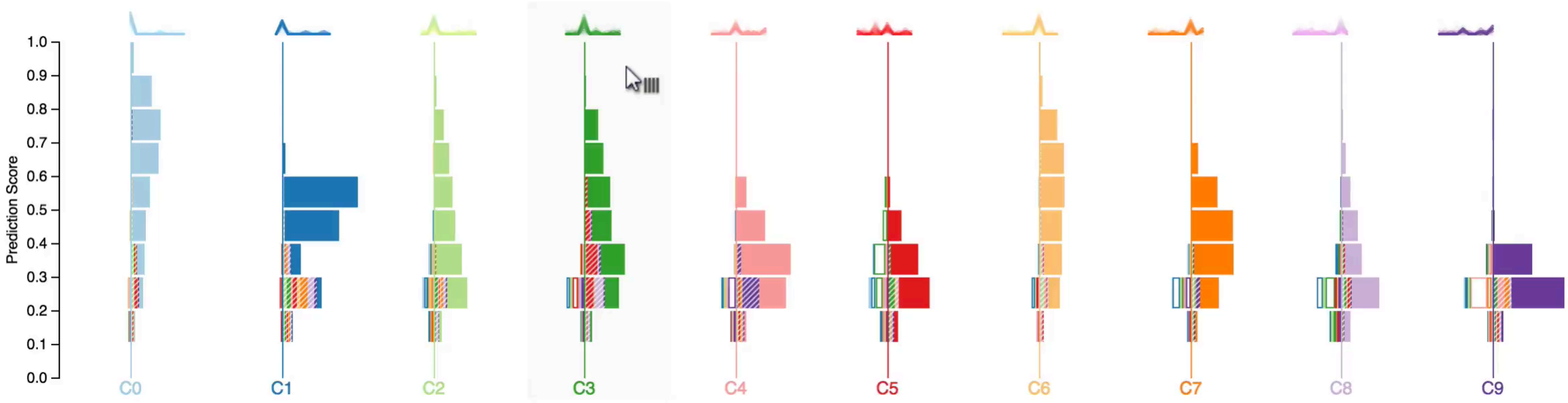


On-hover parallel coordinates for detailed scores

Scalability



Scalability



Toggle between 3-levels of aggregation

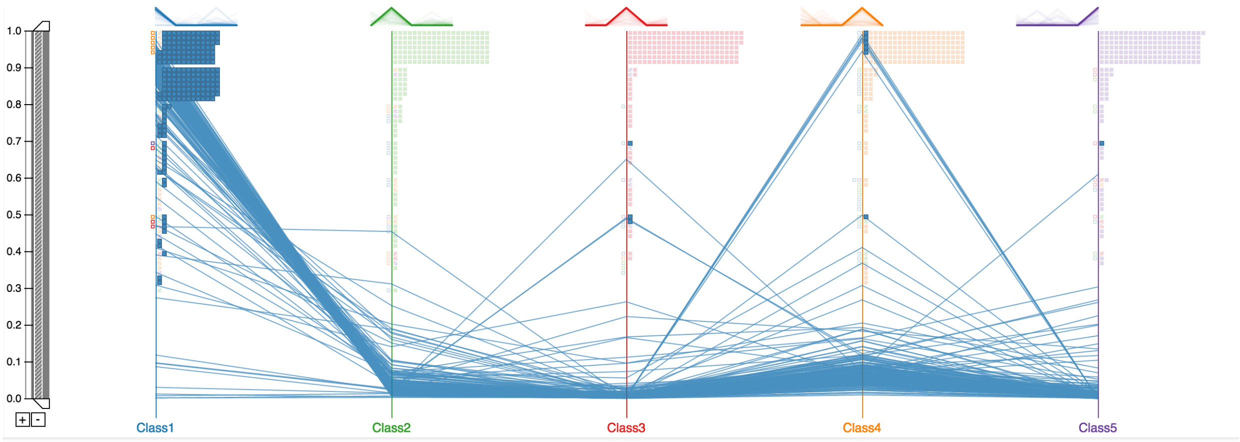
Evaluation

Controlled Experiment

- 24 participants

- Part 1: Comparison
  - Compare Squares against a commonly used ConfusionMatrix
  - Within-subject design
- Part 2: (Squares Only) Score Distribution
  - Evaluate Squares’ ability to convey score distribution

Part 1: Squares vs. Confusion Matrix



ID	Label [↓]	Prediction	Class1	Class2	Class3	Class4	Class5
00516	Class1	Class1	0.866	0.018	0.021	0.066	0.029
00000	Class1	Class4	0.031	0.016	0.006	0.945	0.002
00002	Class1	Class1	0.935	0.021	0.003	0.039	0.003
00022	Class1	Class1	0.878	0.027	0.001	0.092	0.001
00025	Class1	Class1	0.810	0.025	0.036	0.091	0.038
00026	Class1	Class1	0.863	0.027	0.013	0.081	0.016
00028	Class1	Class1	0.906	0.038	0.002	0.045	0.007
00032	Class1	Class1	0.865	0.062	0.001	0.067	0.004
00036	Class1	Class1	0.891	0.026	0.002	0.080	0.002
00044	Class1	Class1	0.468	0.455	0.002	0.067	0.007
00052	Class1	Class1	0.892	0.012	0.003	0.091	0.002
00053	Class1	Class1	0.886	0.034	0.002	0.076	0.001
00054	Class1	Class1	0.918	0.007	0.002	0.072	0.001
00069	Class1	Class1	0.909	0.019	0.004	0.065	0.002
00076	Class1	Class1	0.852	0.044	0.004	0.086	0.014
00083	Class1	Class1	0.924	0.009	0.001	0.065	0.001
00084	Class1	Class1	0.771	0.058	0.010	0.157	0.004
00090	Class1	Class1	0.800	0.106	0.008	0.065	0.021
00097	Class1	Class1	0.960	0.018	0.001	0.018	0.003
00109	Class1	Class1	0.932	0.014	0.004	0.049	0.001
00114	Class1	Class1	0.912	0.025	0.002	0.059	0.002

Predicted

Labeled	Class1	Class2	Class3	Class4	Class5
Class1	186	2	2	1	3
Class2	6	169	1	4	2
Class3	3	0	211	0	3
Class4	3	10	2	199	0
Class5	4	4	5	0	180

Select/Deselect individual cells.
Select cells of a given row/column.

ID	Label	Prediction	Class1	Class2	Class3	Class4	Class5
00014	Class4	Class2	0.098	0.431	0.021	0.423	0.027
00049	Class1	Class2	0.323	0.546	0.005	0.053	0.074
00174	Class4	Class2	0.003	0.861	0.000	0.126	0.009
00193	Class4	Class2	0.037	0.565	0.009	0.081	0.308
00198	Class4	Class2	0.008	0.484	0.016	0.344	0.148
00225	Class4	Class2	0.026	0.675	0.007	0.213	0.080
00230	Class4	Class2	0.018	0.449	0.079	0.255	0.199
00272	Class5	Class2	0.209	0.589	0.012	0.094	0.096
00381	Class5	Class2	0.296	0.332	0.006	0.049	0.317
00526	Class5	Class2	0.024	0.769	0.033	0.095	0.078
00537	Class4	Class2	0.013	0.801	0.002	0.042	0.141
00542	Class4	Class2	0.003	0.915	0.001	0.061	0.019
00700	Class5	Class2	0.220	0.622	0.008	0.074	0.076
00768	Class4	Class2	0.007	0.527	0.001	0.459	0.006
00934	Class4	Class2	0.009	0.585	0.000	0.401	0.004
00998	Class1	Class2	0.220	0.579	0.009	0.175	0.016

Squares with a Sortable Table

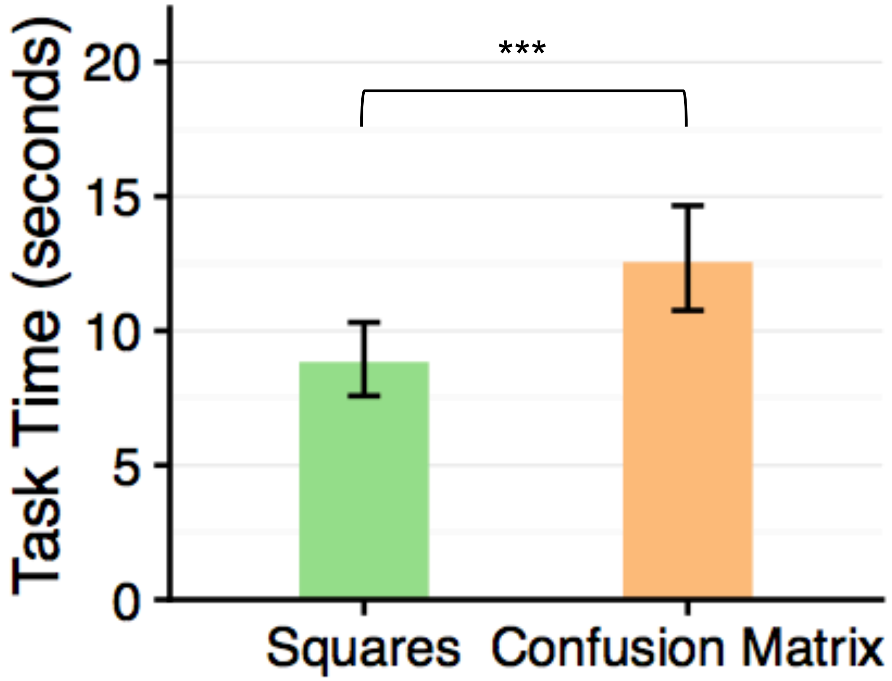
Confusion Matrix with a Sortable Table

Part 1: Tasks

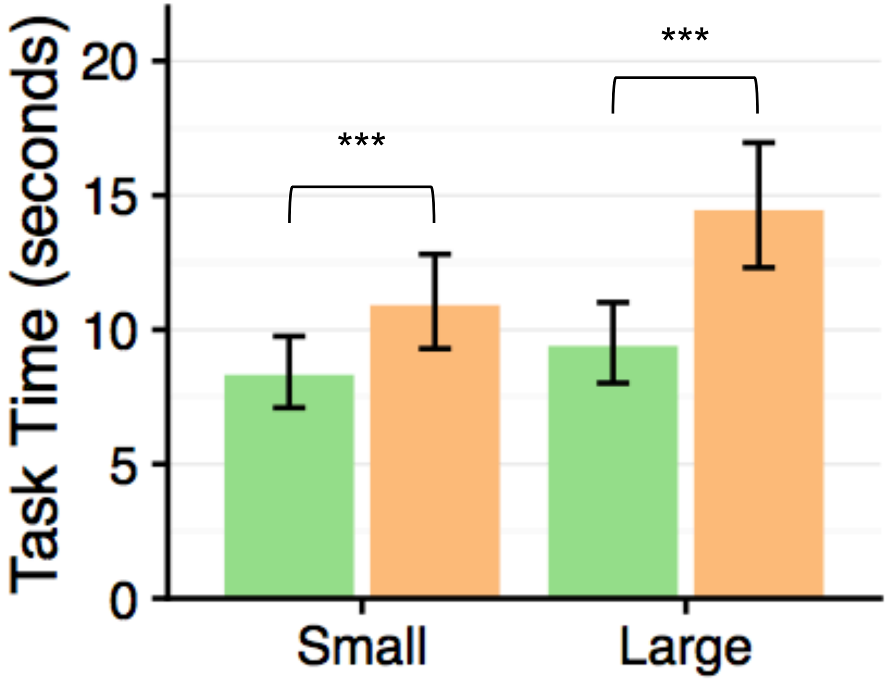
- T1 – Overall
  - Select the classifier with the larger number of errors
- T2 – Class-level
  - Select one of the two classes with the most errors
- T3 – Instance-level
  - Select an error with a score of .9 or above in the wrong class

Part 1: Squares Performed Better

- Task Time



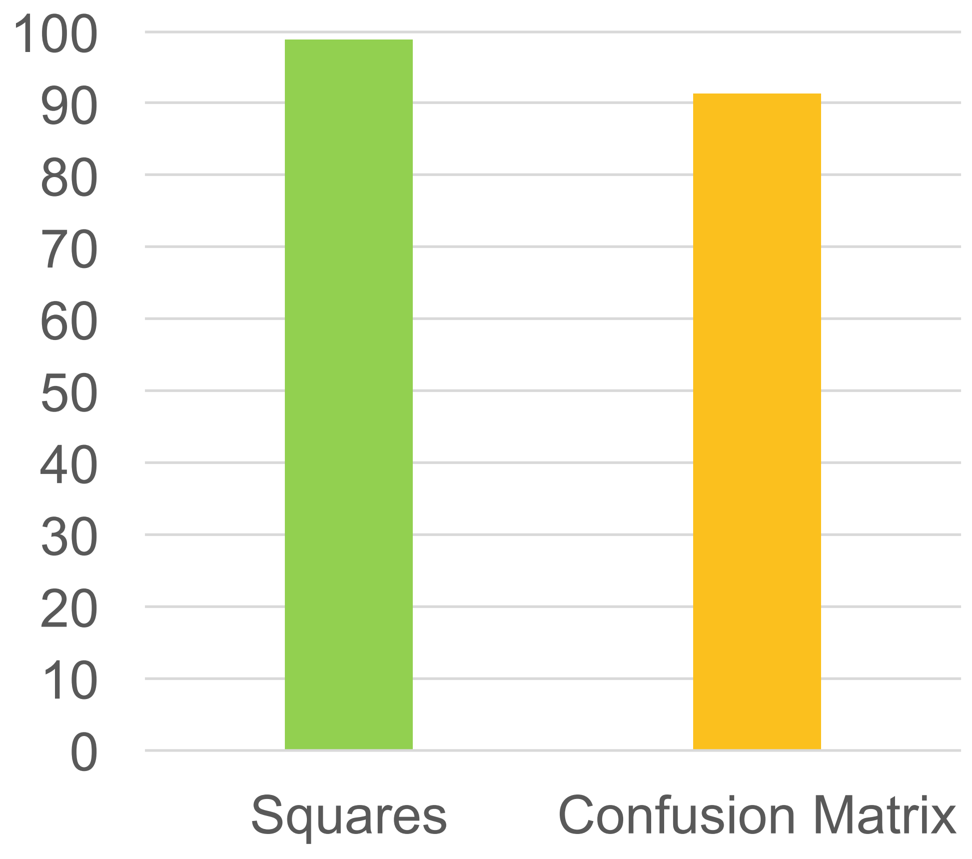
Squares lead to faster task time
(Main Effect: $p < 0.001$)



Squares scale better in terms of the number of classes
(Interaction Effect: $p = 0.012$)

Part 1: Squares Performed Better

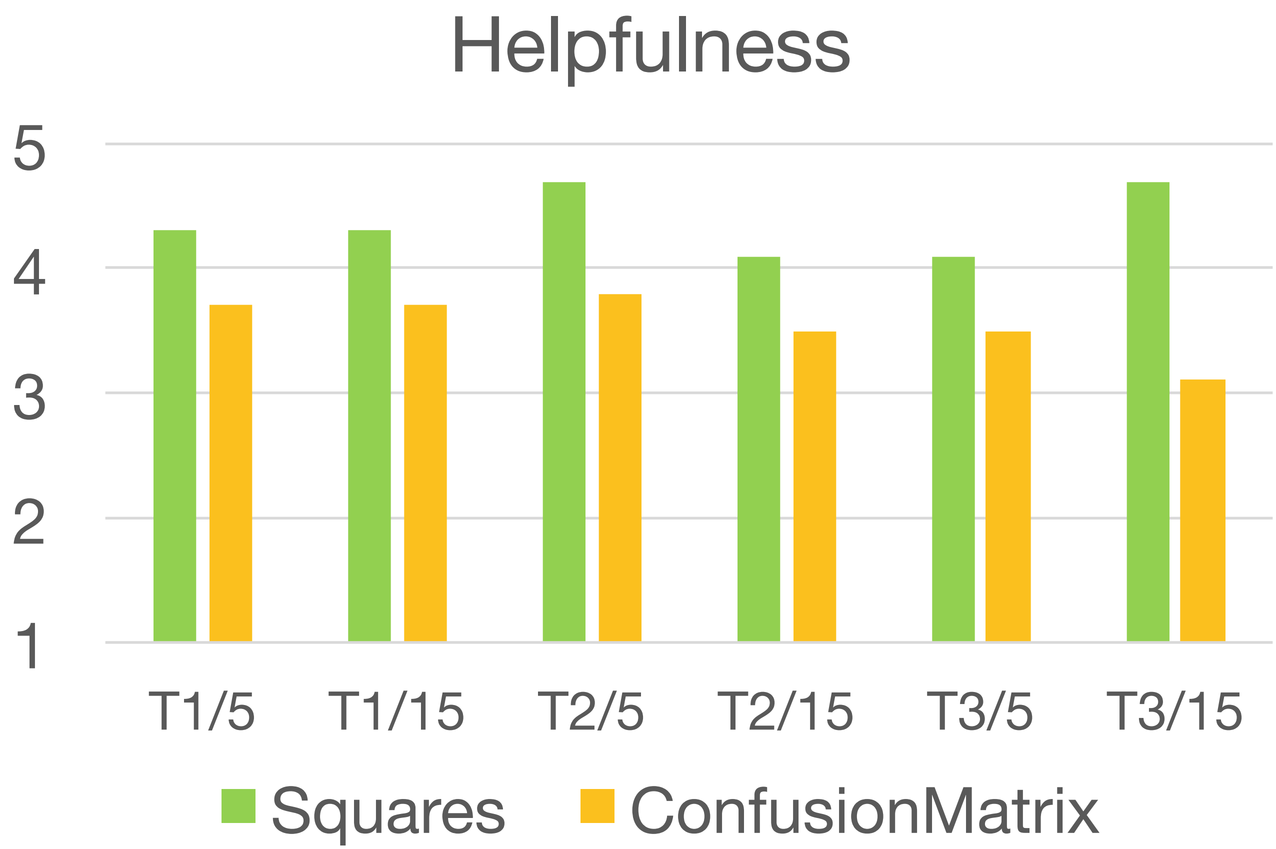
- Accuracy



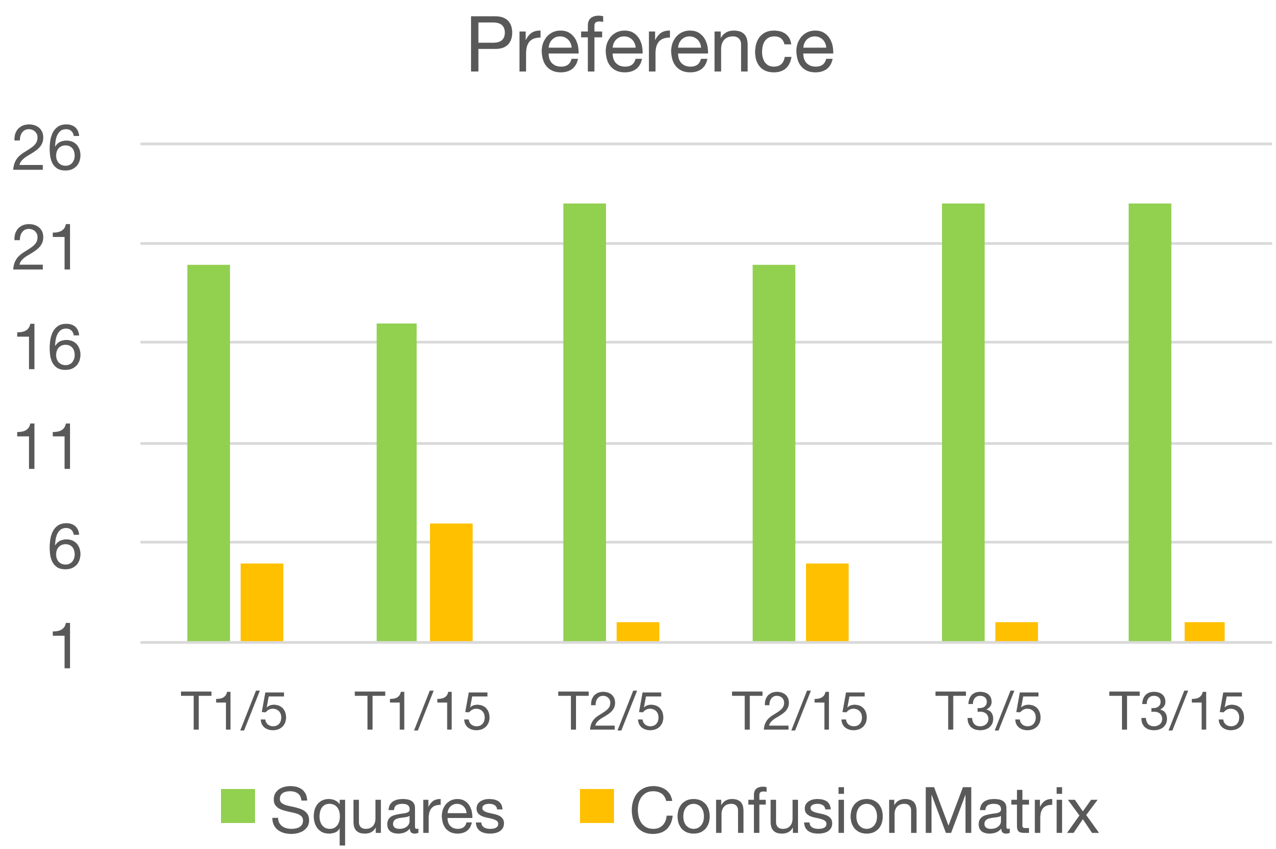
($p < 0.001$)

- Squares lead to more accurate results

Part 1: People Preferred Squares



Squares was more helpful

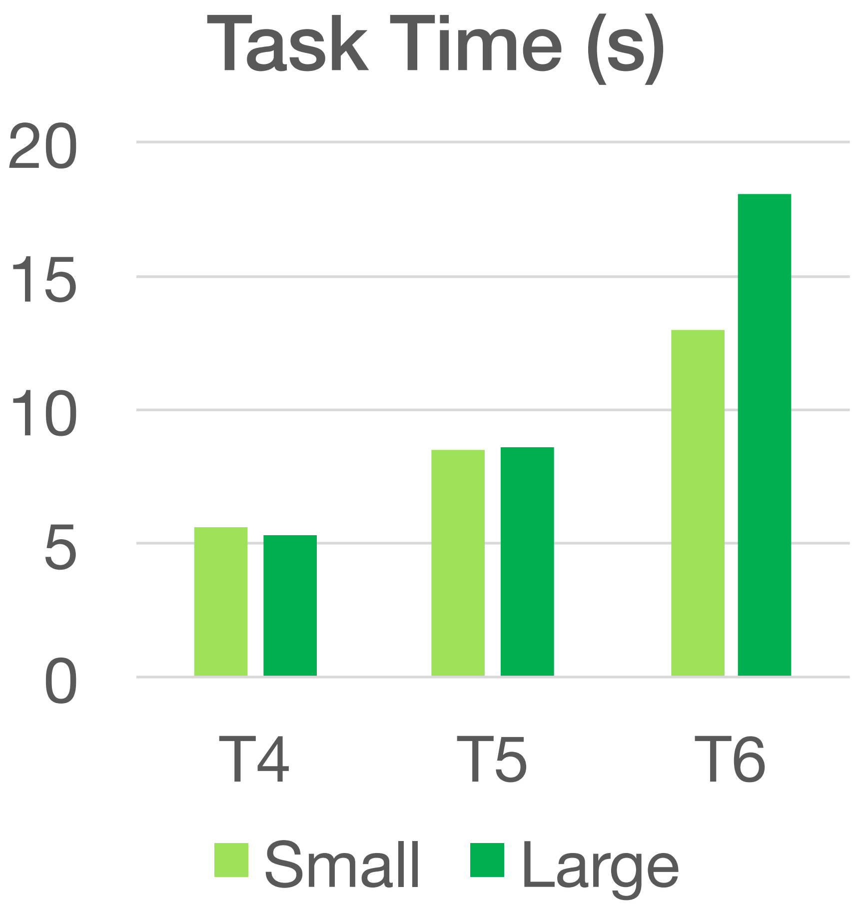


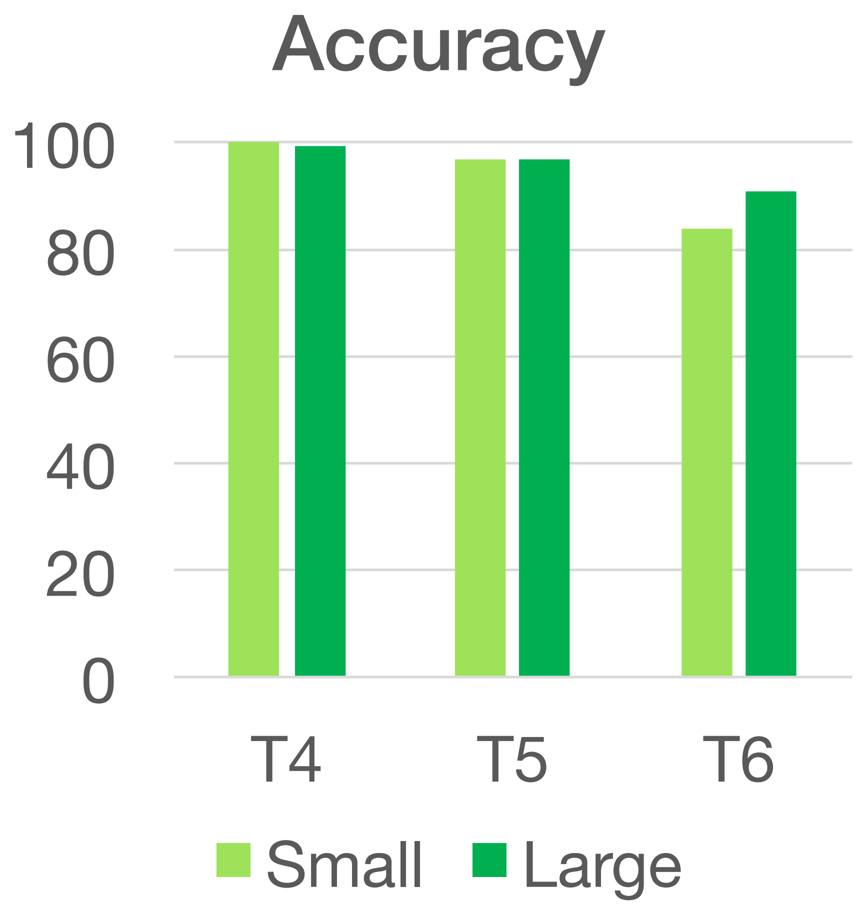
Squares was preferred

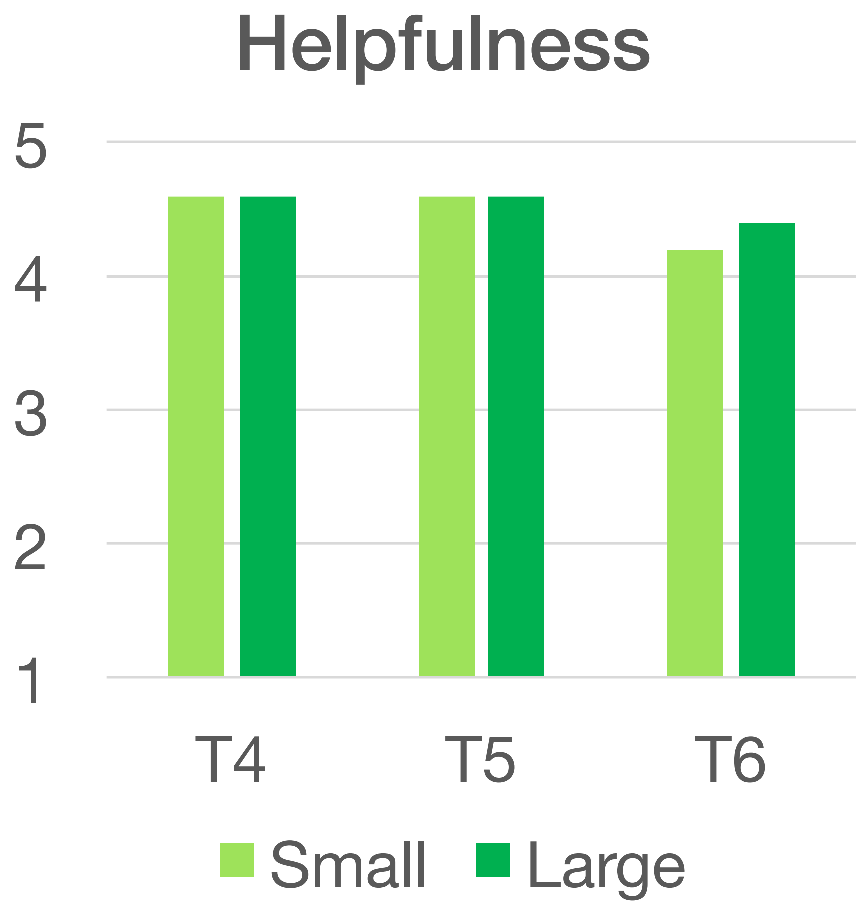
Part 2: (Squares Only) Distribution Tasks

- T4 – Overall
  - Select the classifier with the worst distribution
- T5 – Class-level
  - Select one of the two classes with the worst distribution
- T6 – Confusion
  - Select the two classes most confused with each other

Part 2: Squares was helpful in distribution tasks





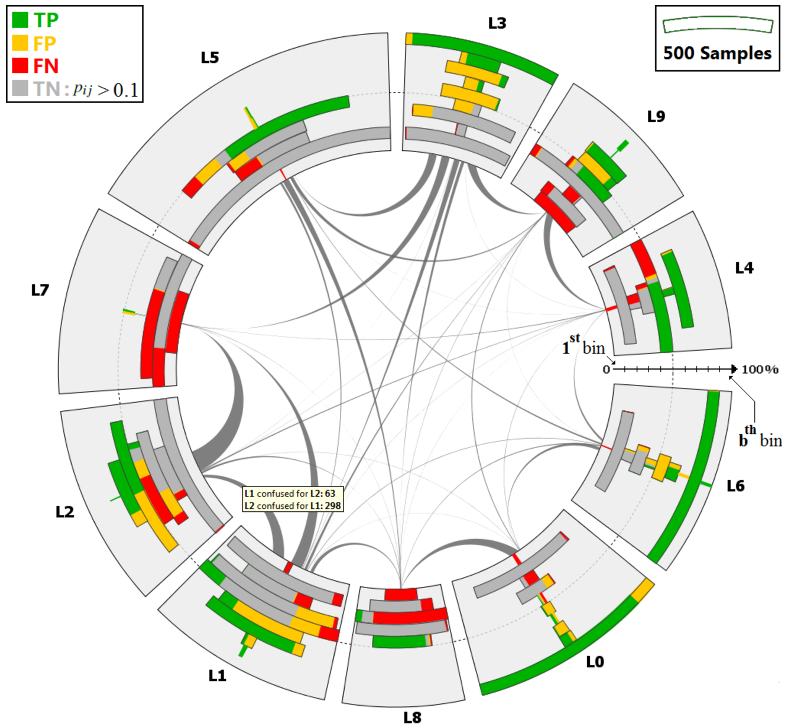


Freeform Feedback

- Positive:
  - *"Granular and at the same time general overview of the classifiers is great."*
  - *"Seeing the distribution of scores is very helpful."*
  - *"Had fun for the first time while classifying!"*
- Negative:
  - *"I prefer having numbers than pure display."*
  - *"[Confusion Matrix is] more straightforward, lower learning curve."*

Future Work

- Further Evaluation
  - Compare to alternative designs of Confusion Matrix, as well as other visualization designs in the literature
- Scalability
  - Supporting more than 20 classes
  - Optimizing color assignments

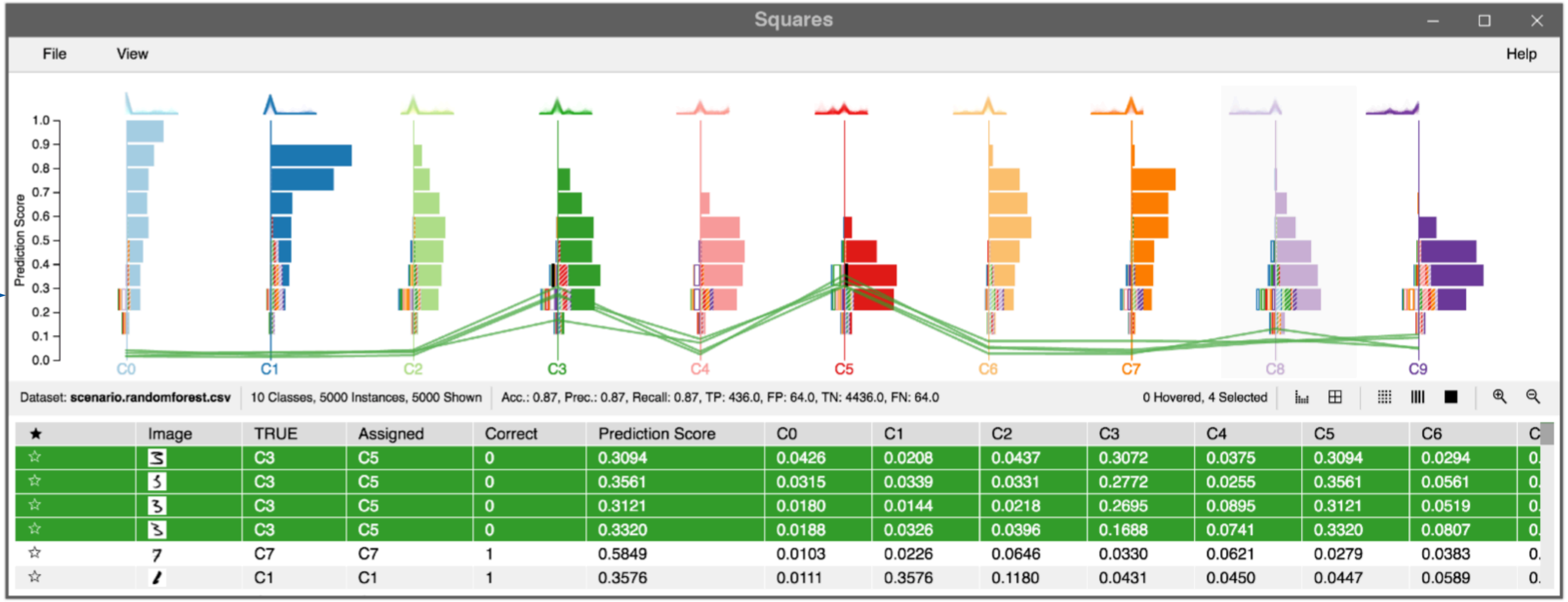


Confusion Wheel [B. Alsallakh, VAST '14]

Squares as a Tool

- Deployed along with a machine learning toolkit within Microsoft

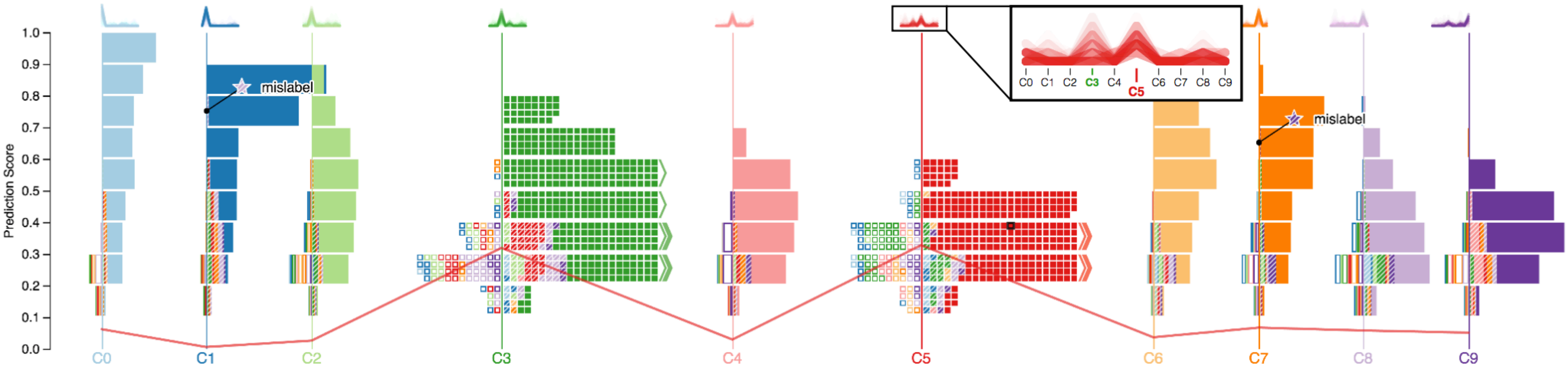
Model Building Interface



Acknowledgements

- We thank the support and feedback from the Machine Teaching Group at Microsoft Research.

- We thank the anonymous reviewers for their constructive comments.



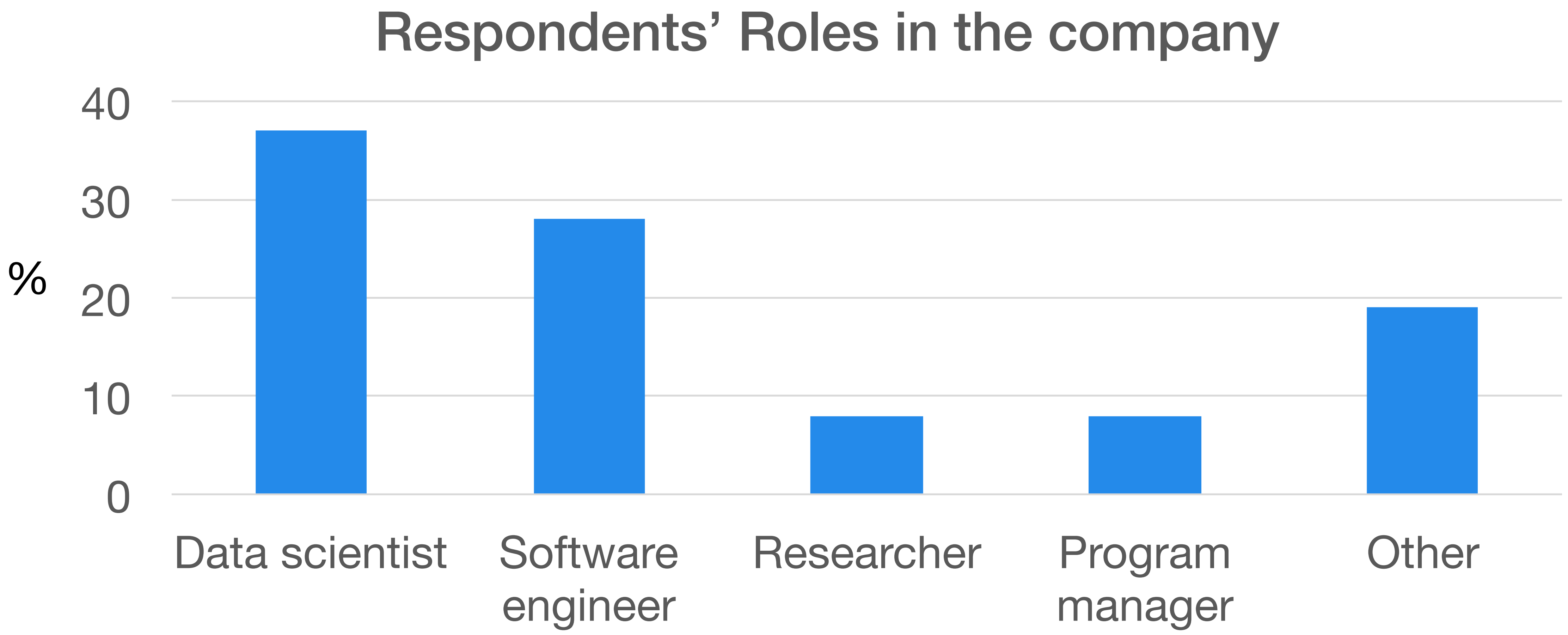
Thanks! Questions?

Donghao Ren (donghao.ren@gmail.com)

University of California, Santa Barbara

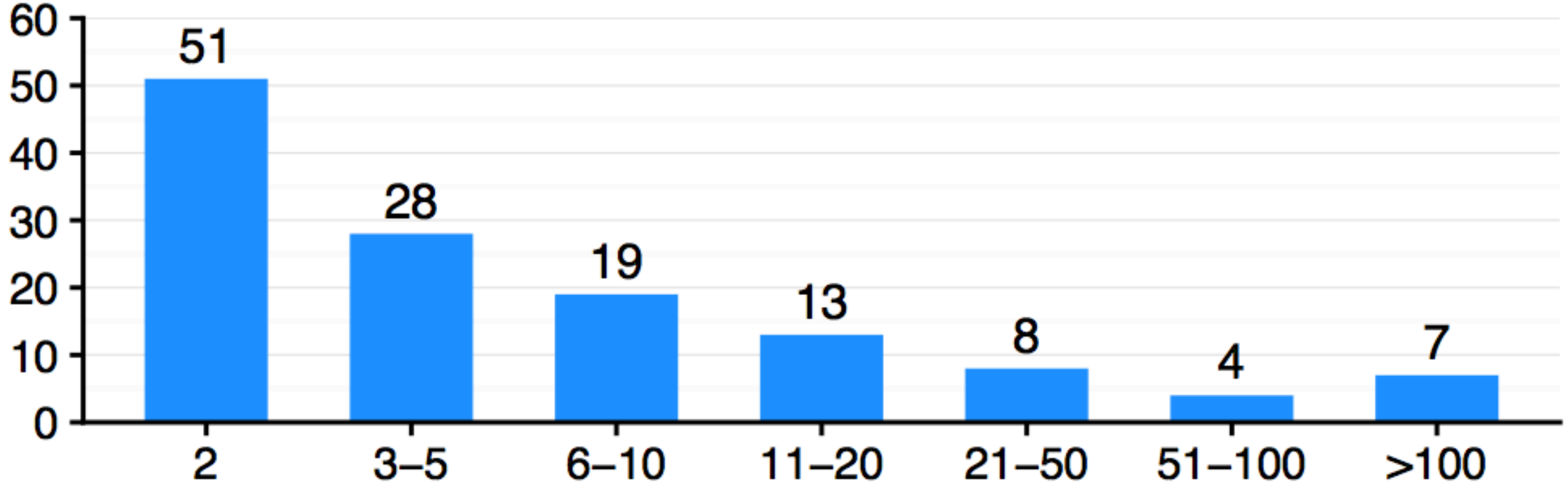
Survey of Machine Learning Practices

- Survey within a large software company in July. 2015.
- 102 respondents:



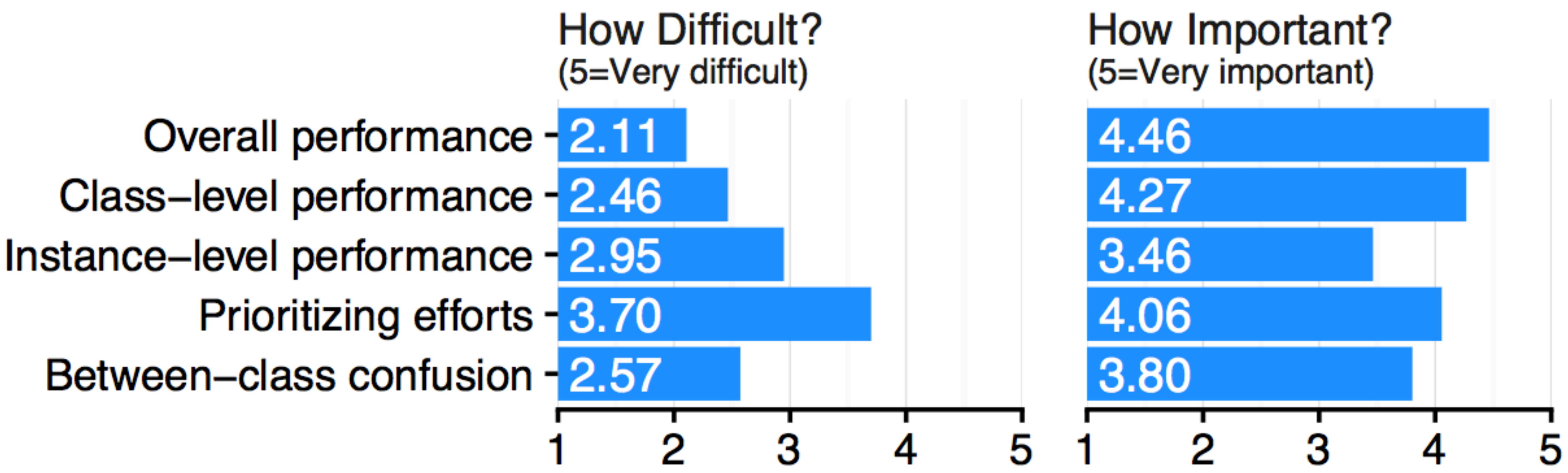
Number of Classes

- How many classes do your classifiers typically deal with (check all that apply)?
  - Most respondents typically deal with less than 20 classes.



Important Tasks

- "How difficult" and "how important" ratings of tasks:
  - Prioritizing efforts is difficult even for expert users.
  - Understanding instance-level performance is relatively more difficult in common tools.



Integrating into LUIS (Language Understanding Intelligent Service)