AutoML Pipeline for Dabular Datasets

Moein Ghaeini, Alexis Niermann
FrankHutterFanclub-1

Motivation

1. create a general purpose pipeline that can handle any tabular dataset without overfitting to given data
2. try to incorporate ideas from the lecture, but only keep them if they improve general performance
3. Supports multiple preset configs for quick, balanced or extensive training
4. trains quickly

Contributions

pipeline design, implementing meta-learning, troubleshooting, user experience:
Moein

adapting course material, feature engineering, poster, data analysis:
Alexis

🟩	Week 1
🟧	Week 2
□	Week 3
🟪	Week 4
□	Week 5
🟥	Week 6
□	Week 7
🟪	Week 8
□	Week 9
□	Week 10
□	Bonus
□	Literature

Our Approach

Testing / Phase 1

Tried:

TabPFN: could reliably get close to baseline but very inefficient at > 1000 samples, thus doesn't leverage larger datasets well

NAS NN: tended to overfit

"Simpler" ML models usually performed better -> also greater interpretability

Metalearning + Feature Engineering: generally improved performance

Improved HPO using Optuna

Optional Multi-Criteria Optimization (efficiency + accuracy)

Final Pipeline / Phase 2

1. **Pipeline reads input config to set options like meta-learning, seed, time budget etc**
2. **Feature Engineering and Preprocessing: fills missing columns**
3. **Generates meta-features and suggests initial config based on that**
4. **Trains multiple basic ML models for 5 folds and keeps n_best for ensemble (stacking). HP searchspace predicted by Optuna**

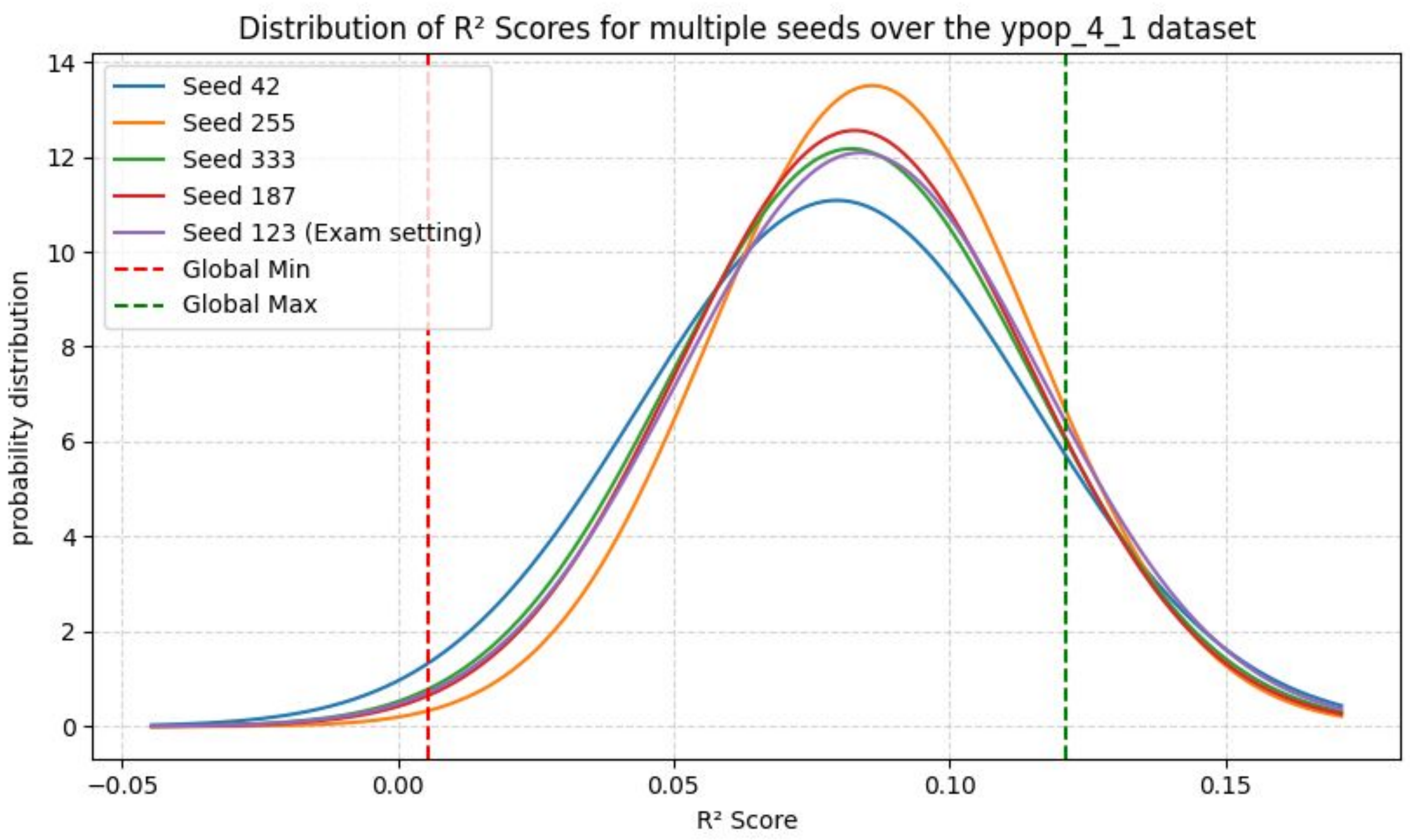
Resources Used

For development:
- 7-Core GPU
- 8-Core CPU
- Total compute estimate: 1000 CPU-h

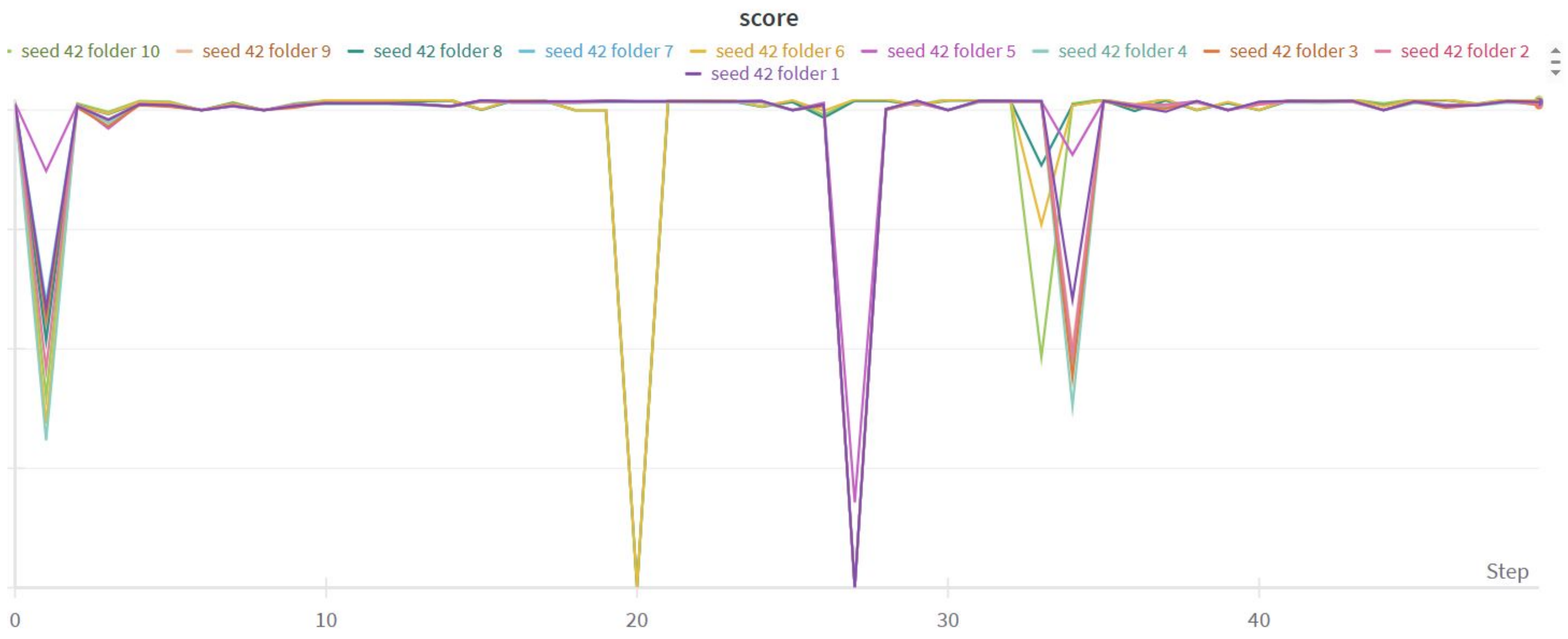
For AutoML:
- VSCode

Workforce:
- 1.5 full week on average

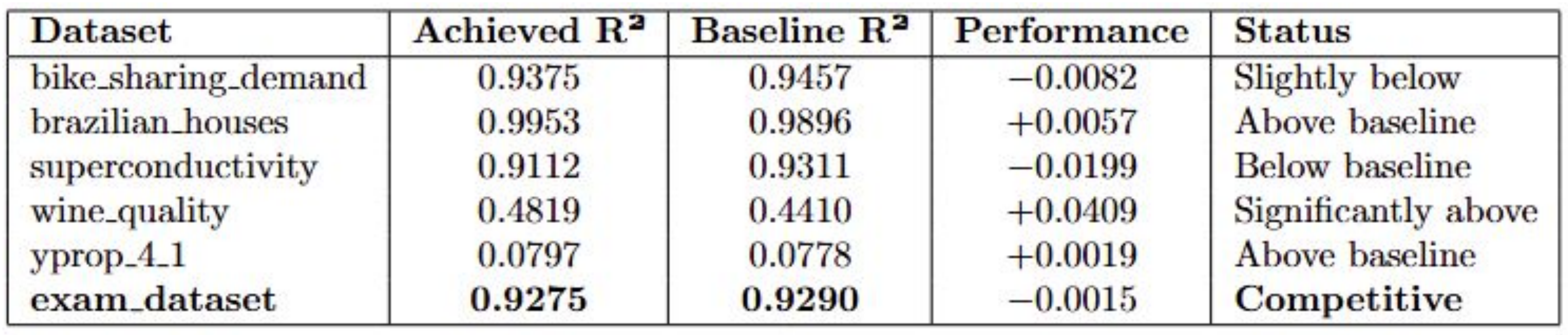
Empirical Results



different seeds on one dataset over all 10 folders using mostly the balanced setting



example changes in score across the folders during the 50 steps of one complete run



Dataset	Achieved R^2	Baseline R^2	Performance	Status
bike_sharing_demand	0.9375	0.9457	−0.0082	Slightly below
brazilian_houses	0.9953	0.9896	+0.0057	Above baseline
superconductivity	0.9112	0.9311	−0.0199	Below baseline
wine_quality	0.4819	0.4410	+0.0409	Significantly above
yprop_4_1	0.0797	0.0778	+0.0019	Above baseline
exam_dataset	**0.9275**	**0.9290**	−0.0015	**Competitive**

Number of queries for test score generation: **1**