MetaRank: Intelligent Algorithm Selection

"One of the holy grails of machine learning is to automate more and more of the feature engineering process" - Pedro Domingos

Balaji, Kenjiro, Surbhi
AutoML_Gods_BKS

Modality 1/3

Background

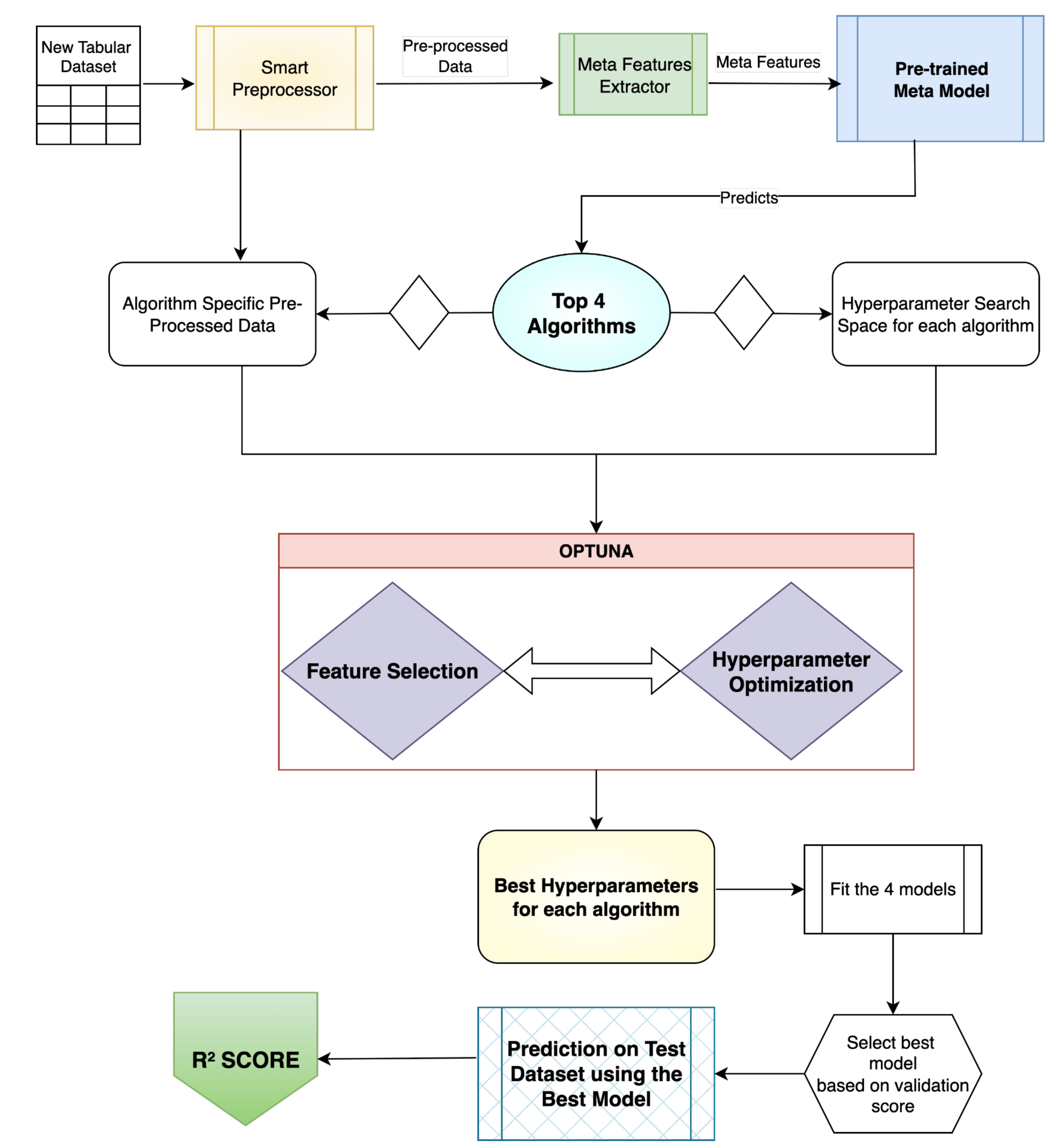
- AutoML automates model tuning and evaluation but often treats all algorithms equally.
- Traditional pipelines typically lack mechanisms to **leverage knowledge from previous datasets**.
- Choosing the right model early is crucial - it reduces search time and improves outcomes.
- Dataset **meta-features** (e.g., size, skewness, correlations) can capture important characteristics for guiding model selection.

Motivation

- In tabular regression, model performance can vary widely based on dataset **characteristics**.
- Traditional ML pipelines require expert-driven feature engineering and model selection. Hyperparameter tuning alone is not enough - real performance gains often come from effective feature transformation and selection.
- So we propose a meta-model designed to learn from historical dataset performance and predict which algorithm will work best on new, unseen datasets.

- [x] Week 1
- [x] Week 2
- [x] Week 3
- [ ] Week 4
- [ ] Week 5
- [ ] Week 6
- [ ] Week 7
- [x] Week 8
- [ ] Week 9
- [ ] Week 10
- [ ] Bonus
- [ ] Literature

Proposed Architecture



Final Inference Pipeline:

1. Extract meta-features from new dataset
   - feature size, skewness, correlations
   - 1% dataset evaluation
1. Feed into trained meta-model → get ranked list
2. Select top 4 algorithms
3. For each algorithm:
   - Algorithm specific dataset pre-processing
   - Define the hyperparameter search space
1. Run Optuna to jointly optimize Hyperparameters and Feature Selection on each algorithm
2. Train each model with the best hyperparameters
3. Pick the best model based on the validation score
4. Perform final model training for the best model

Resources Used

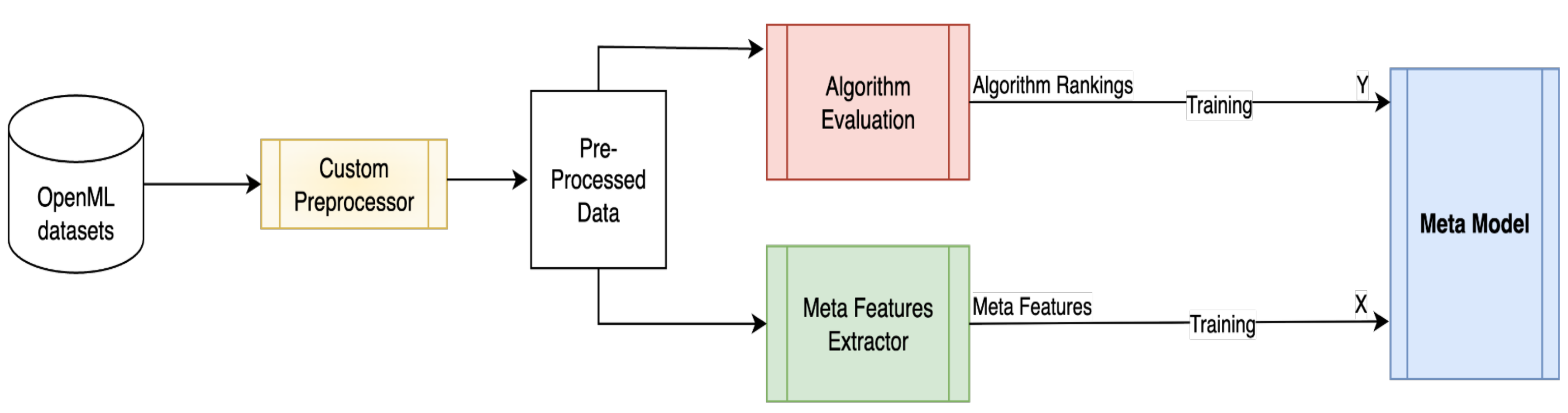
For development:
- 1 NVIDIA GeForce GTX 1050
- 1 Apple M2
- 1 Apple M3

For AutoML:
- Apple M3 pro, 18GB
- 4h

Workforce:
- 1 full week

Meta-Model Training

- Extracted 30 custom meta-features for ~200 OpenML datasets
- Algorithms Pool: XGBoost, LightGBM, TabPFN, SVR, Linear Regression, BayesianRidge, Decision Tree, Random Forest, Gradient Boosting, MLPRegressor
- Algorithm ranked according to R^2 scores for each dataset
- Meta-model trained using a custom Multi-Head Ranking Neural network:
  - Multi-head output (one head per algorithm)
  - Ranking is more robust to dataset-specific relative performance



Results & Takeaways

- Takeaways:
  - Meta-learning effectively reduces search space and improves tuning efficiency.
  - Combining algorithm selection with per-model HPO is more efficient than full AutoML search.
  - Architecture generalizes well across unseen tabular datasets.
- Limitations:
  - Meta-model trained only on OpenML datasets - may not generalize to out-of-distribution datasets
  - Only used R^2 as optimization metric - lacks support for custom metrics (e.g., inference time, fairness)
- Future Developments:
  - Diverse and larger dataset pool for the meta model
  - Extend to classification tasks and multimodal datasets
  - Add multi-objective optimization (e.g., R^2 + latency + model size)

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

AutoML.org

ufr