Aalto-yliopisto
Perustieteiden korkeakoulu

Impact of model size on tree ensemble prediction accuracy and optimization time

Eetu Reijonen

1.11.2023

Advisor: Nikita Belyak

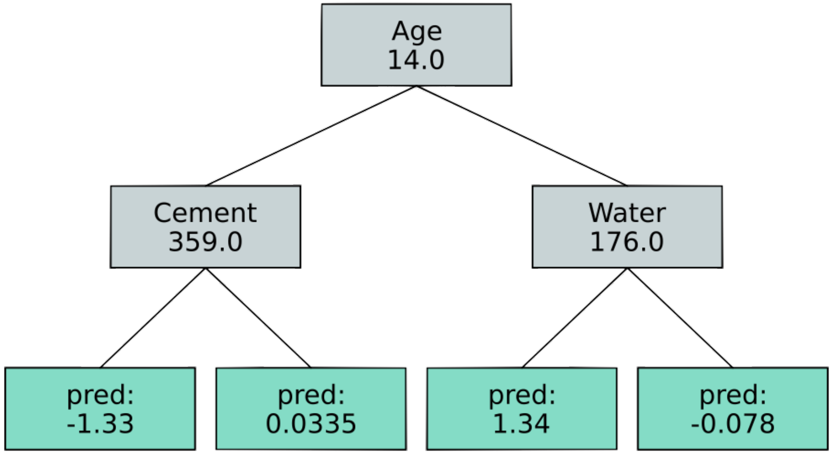
Supervisor: Fabricio Oliveira

Systeemianalyysin laboratorio

Background – decision trees and tree ensembles

- A decision tree maps an input to an output leaf that gives the tree prediction
  - Interpretable machine-learning model
- Tree ensemble = collection (forest) of decision trees
  - Random forests, gradient-boosted trees (XGBoost)

$DT : (Age, Cement, Water) \rightarrow Concrete\ strength$



Background – tree ensemble optimization

- How to find an input that maximizes/minimizes the tree ensemble output? (for regression trees)
- Formulated as a mixed-integer optimization (MIO) problem
- ”Optimizing a tree ensemble” – solving the corresponding MIO problem

$$\underset{\mathbf{x},\mathbf{y}}{\text{maximize}} \quad \sum_{t=1}^{T} \sum_{\ell \in \mathbf{leaves}(t)} \lambda_t \cdot p_{t,\ell} \cdot y_{t,\ell} \tag{2a}$$

$$\text{subject to} \quad \sum_{\ell \in \mathbf{leaves}(t)} y_{t,\ell} = 1, \quad \forall\, t \in \{1,\dots,T\}, \tag{2b}$$

$$\sum_{\ell \in \mathbf{left}(s)} y_{t,\ell} \le \sum_{j \in \mathbf{C}(s)} x_{\mathbf{V}(s),j}, \quad \forall\, t \in \{1,\dots,T\},\ s \in \mathbf{splits}(t), \tag{2c}$$

$$\sum_{\ell \in \mathbf{right}(s)} y_{t,\ell} \le 1 - \sum_{j \in \mathbf{C}(s)} x_{\mathbf{V}(s),j}, \quad \forall\, t \in \{1,\dots,T\},\ s \in \mathbf{splits}(t), \tag{2d}$$

$$\sum_{j=1}^{K_i} x_{i,j} = 1, \quad \forall\, i \in \mathscr{C}, \tag{2e}$$

$$x_{i,j} \le x_{i,j+1}, \quad \forall\, i \in \mathscr{N},\ j \in \{1,\dots,K_i - 1\}, \tag{2f}$$

$$x_{i,j} \in \{0,1\}, \quad \forall\, i \in \{1,\dots,n\},\ j \in \{1,\dots,K_i\} \tag{2g}$$

$$y_{t,\ell} \ge 0, \quad \forall\, t \in \{1,\dots,T\},\ \ell \in \mathbf{leaves}(t). \tag{2h}$$

Mišić, V.V., 2020. Optimization of tree ensembles. *Operations Research*, *68*(5), pp.1605-1624.

Objective

- Evaluate the tradeoff between tree ensemble prediction accuracy and optimization time
  - Tree ensemble size: number of trees and maximum depth of trees
  - Increasing the size improves prediction accuracy but also increases the optimization time

Methods

- Programming language: Julia, tree ensemble model: EvoTrees.jl (gradient-boosted trees), MIO formulation: JuMP, solver: Gurobi
- Hardware: 2016 HP laptop with i7 and 16 GB of RAM
- Three datasets: concrete strength, drug design - OX2 and 3A4

Table 1: Summary of the datasets used

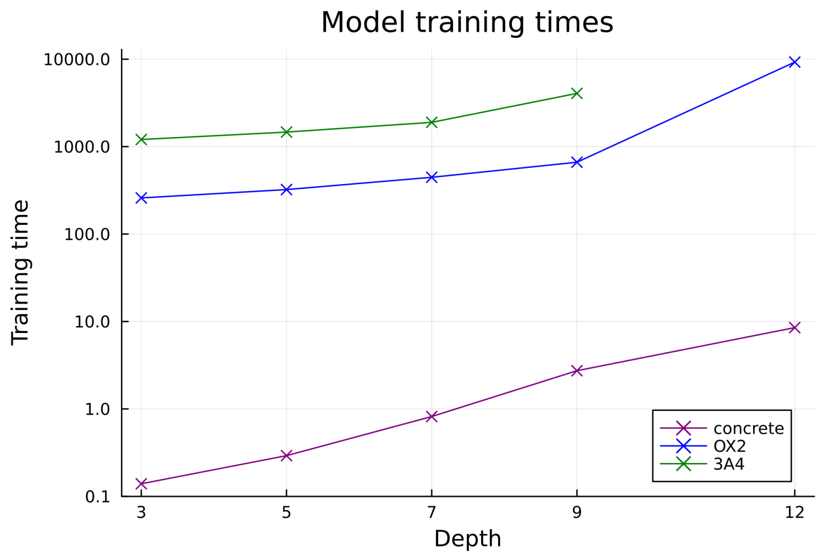
Dataset	No. variables	No. observations (train)	No. observations (test)
Concrete	9	772	258
OX2	5790	11151	3704
3A4	9491	37241	12338

Experiments

1. Training EvoTrees models for each of the datasets
   - Forest sizes: 50, 100, 200, 350, 500, 750, 1000
   - Maximum depths: 3, 5, 7, 9, 12
   - 3 datasets x 7 forest sizes x 5 depths = 105 models
   - Training time and testing prediction accuracy measured for each of the EvoTrees models
2. Formulating the MIO problems and solving them for each of the EvoTrees models
   - Optimization time measured for each
   - Time limit of 2 hours imposed

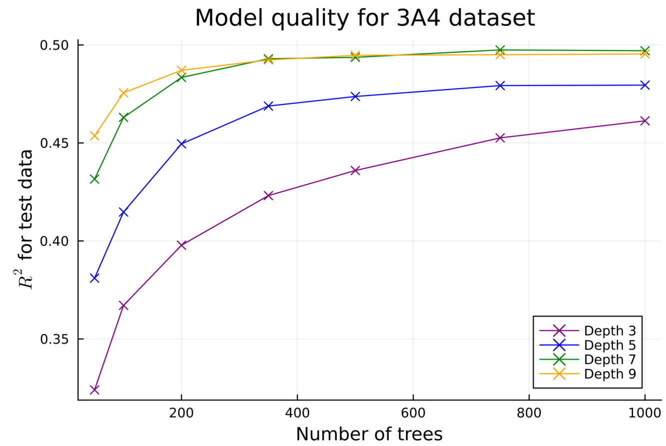
Results – EvoTrees training time

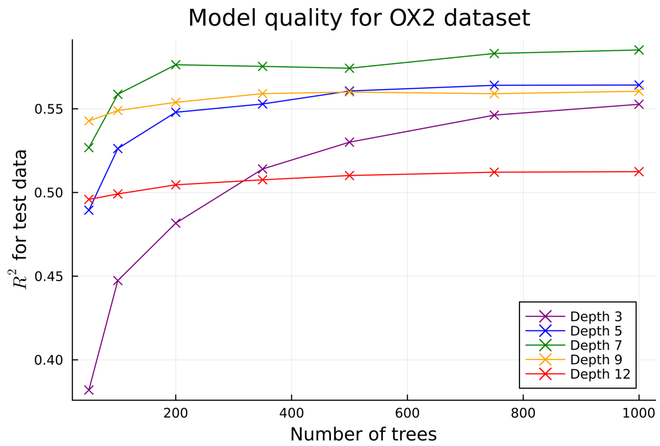
- Every EvoTrees model trained has 1000 trees
  - Predictions of EvoTrees models with fewer trees generated by limiting the number of trees
- Exponential increase in training time with the increase in maximum depth
- Small dataset fast (concrete), large datasets slow (drug design)

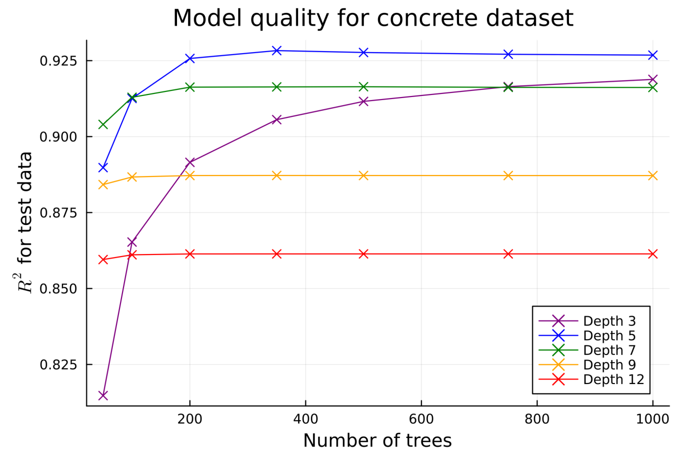


Results – EvoTrees prediction accuracy

- Coefficient of determination (R^2) – “goodness of the model” (from 0 to 1)
- Concrete: R^2 of 0.93 (200 trees, depth 5)
- OX2: R^2 of 0.57 (200 trees, depth 7)
- 3A4: R^2 of 0.48 (200 trees, depth 7)
- R^2 scores do not significantly improve with larger models

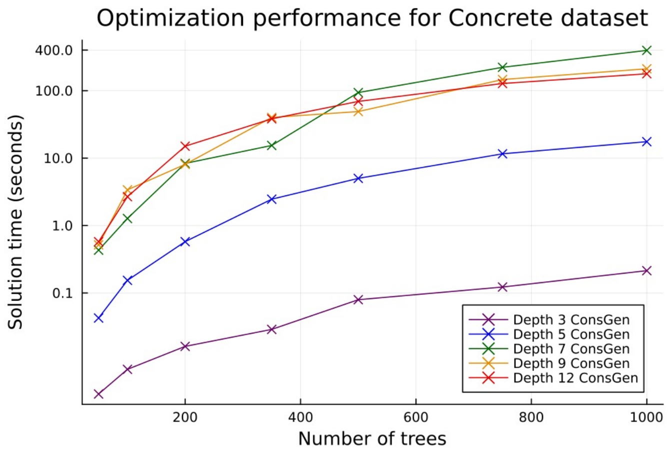


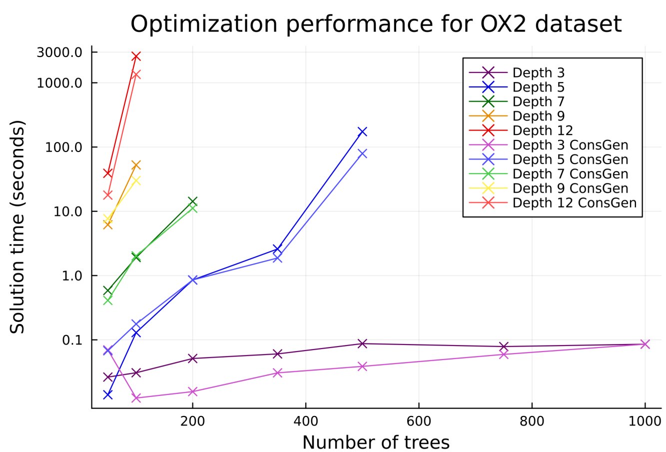


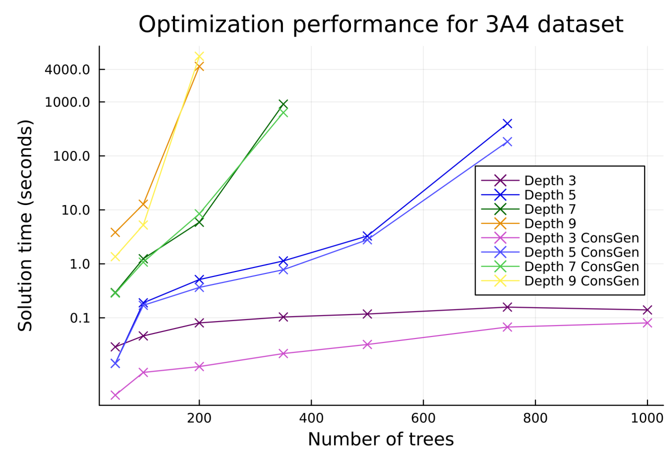


Results – optimization time (MIO solve time)

- For the EvoTrees model sizes mentioned in the last slide:
  - Concrete (200 trees, depth 5) ~1s
  - 3A4 (200 trees, depth 7) ~10s
  - OX2 (200 trees, depth 7) ~10s
- Explosion in optimization time







Conclusions

- To maximize prediction accuracy, larger EvoTrees models are required for larger datasets
  - Size of dataset = number of observations and variables
  - Size of EvoTrees model = number of trees and maximum depth
- Increasing EvoTrees model size doesn't improve prediction accuracy after a certain point
- Increasing EvoTrees model size increases training time and optimization time exponentially
- For our datasets, good (and maximal) prediction accuracy could be reached with EvoTrees models that can be optimized in seconds
  - 0.93 for concrete, 0.48 for 3A4, 0.57 for OX2 (Kaggle competition winner 0.49)

Limitations

- Lack of variation in the datasets
  - Number of variables and type of data
- Experiments only conducted once
  - Taking the average of multiple runs would add more reliability to the results
- Only gradient-boosted trees used
  - Random forest models could have been tested as well
- Non-powerful hardware
  - Could even larger models be optimized in reasonable time with more powerful hardware?

References

- Mišić, V.V., 2020. Optimization of tree ensembles. *Operations Research*, *68*(5), pp.1605-1624.
- Merck Molecular Activity Challenge Leaderboard. Kaggle. https://www.kaggle.com/competitions/MerckActivity/leaderboard. Visited 24.10.2023