**Mastering Machine Learning: What’s New in XGBoost 2.0**

[[Vivekpandian](https://medium.com/@vivekpandian?source=post_page-----84d59cdddc6d--------------------------------)](https://medium.com/@vivekpandian?source=post_page-----84d59cdddc6d--------------------------------)

[Vivekpandian](https://medium.com/@vivekpandian?source=post_page-----84d59cdddc6d--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F23481d72062b&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40vivekpandian%2Fmastering-machine-learning-whats-new-in-xgboost-2-0-84d59cdddc6d&user=Vivekpandian&userId=23481d72062b&source=post_page-23481d72062b----84d59cdddc6d---------------------post_header-----------)

3 min read

·

Oct 12, 2023

66



Image Credit: <https://abzooba.com/resources/blogs/why-xgboost-and-why-is-it-so-powerful-in-machine-learning/>

XGBoost 2.0 marks a significant milestone in the evolution of this renowned machine learning library. This release introduces groundbreaking features, including multi-target trees with vector-leaf outputs, enhanced GPU support, and improved memory management. The default “hist” tree method now ensures more efficient model training, while the introduction of learning-to-rank capabilities and quantile regression further expands XGBoost’s versatility. With a focus on performance, flexibility, and usability, XGBoost 2.0 empowers data scientists and machine learning practitioners to achieve even greater insights and accuracy in their models.

**Multi-Target Trees with Vector-Leaf Outputs**

* Version 2.0 introduces vector-leaf tree models for multi-target regression, multi-label classification, and multi-class classification.
* Unlike the previous version, XGBoost can now build one tree for all targets.
* This feature helps prevent overfitting, produces smaller models, and considers target correlations.
* It’s still a work in progress.

**New Device Parameter**

* A new “device” parameter replaces several GPU-related parameters.
* Users can specify the device and its ordinal to run XGBoost on a specific device.
* The old “gpu\_hist” behavior is deprecated.

**Hist as Default Tree Method**

* Starting from version 2.0, the “hist” tree method is the default.
* This improves model training efficiency and consistency

**GPU-Based Approx Tree Method**

* Version 2.0 introduces initial support for the “approx” tree method on GPU.
* It can be used with the “device” and “tree\_method” parameters.
* Performance is still being optimized.

**Optimizing Histogram Size on CPU**

* A new parameter, “max\_cached\_hist\_node,” limits CPU cache size for histograms.
* It prevents aggressive caching and reduces memory usage.
* There’s also a memory usage reduction for “hist” and “approx” tree methods on distributed systems.

**Improved External Memory Support**

* XGBoost now has improved external memory support with the “hist” tree method.
* It uses memory mapping, reducing CPU memory usage.
* Users are encouraged to try it in version 2.0.0.

**Learning to Rank**

* A new implementation for the learning-to-rank task is introduced in XGBoost 2.0.
* Several new parameters and features are added for ranking tasks.
* NDCG (Normalized Discounted Cumulative Gain) is now the default objective function.

**Automatically Estimated Intercept**

* The “base\_score” parameter is automatically estimated based on input labels for optimal accuracy.

**Quantile Regression**

* XGBoost now supports quantile regression, minimizing the quantile loss.
* It allows training with multiple target quantiles simultaneously

**L1 and Quantile Regression Learning Rate**

* The L1 and quantile regression objectives now support learning rate scaling for leaf values.

**Export Cut Value**

* Users can export quantile values used for the “hist” tree method.

**Column-Based Split and Federated Learning**

* Progress is made on column-based splits for federated learning.
* Version 2.0 supports various data split methods and vertical federated learning.

**PySpark**

* The PySpark interface gains new features and optimizations, including GPU-based prediction and improved data initialization.

**Other General New Features**

* Lists new features that are general to all language bindings, such as using the array interface for CSC matrices and including CUDA compute 90 in the default build.

**Other General Optimization**

* Lists general optimizations that apply to all language bindings, like improved performance for input with array\_interface on CPU and CUDA optimizations.

In conclusion, XGBoost 2.0 stands as a testament to the relentless pursuit of excellence in the field of machine learning. Its innovative features, such as multi-target trees, enhanced GPU support, and advanced memory management, redefine the boundaries of what’s achievable in model development. With the default adoption of the “hist” tree method, model training becomes more efficient and consistent. Moreover, the inclusion of learning-to-rank capabilities and quantile regression opens doors to a wider range of applications. XGBoost 2.0 solidifies its position as a cornerstone tool for data scientists, offering the means to unlock new insights and push the boundaries of machine learning further than ever before.

Reference :  
<https://github.com/dmlc/xgboost/releases>

[Machine Learning](https://medium.com/tag/machine-learning?source=post_page-----84d59cdddc6d---------------machine_learning-----------------)

[Xgboost](https://medium.com/tag/xgboost?source=post_page-----84d59cdddc6d---------------xgboost-----------------)

[Data Science](https://medium.com/tag/data-science?source=post_page-----84d59cdddc6d---------------data_science-----------------)

[Regression](https://medium.com/tag/regression?source=post_page-----84d59cdddc6d---------------regression-----------------)

[Classification](https://medium.com/tag/classification?source=post_page-----84d59cdddc6d---------------classification-----------------)