Comparison chart: Machine learning frameworks

Feature/ aspect	TensorFlow	PyTorch	Scikit-learn	Keras	Apache MXNet	Caffe
Ease of use	Moderate—high learning curve	High—intuitive and Pythonic interface	High—simple and consistent API	Very high—user-friendly and modular	Moderate—steep learning curve	Moderate—configuration-driven
Primary strengths	Scalability, production-ready,	Flexibility, dynamic computation	Classical ML algorithms, data preprocessing	High-level API, simplicity, integration with	Hybrid programming model, distributed	Speed and efficiency, optimized for
Primary weaknesses	Complexity, verbose syntax, challenging	Less production-ready, smaller ecosystem	Not suitable for deep learning, limited	Limited flexibility, less control, performance	Smaller community, steeper learning	Limited flexibility, less active
Community & support	Very large, extensive	Large, growing rapidly	Large, well established	Large, benefits from	Smaller, but active in specific	Smaller, slower development
Deployment	Excellent—supports cloud, mobile, and	Good—emerging tools for production	Limited—mainly for data analysis and small-scale	Good—integrated with TensorFlow for	Excellent—optimized for large-scale	Moderate—mainly research and
Supported models	Deep learning (CNNs, RNNs,	Deep learning (CNNs, RNNs,	Classical ML (SVM, Decision	Deep learning (CNNs, RNNs)	Deep learning (CNNs, RNNs),	Convolutional Neural
Scalability	Very high—supports large-scale distributed	High—supports distributed training	Moderate—limited to single-machine	High—scales with TensorFlow	Very High—designed for distributed	Moderate—optimized for single-machine
Flexibility	High—can handle a wide range of	Very High—dynamic graph allows for on-the-	Moderate—best for standard ML tasks	Moderate—high-level abstraction limits	High—supports both symbolic and imperative	Low—best for specific tasks like image
GPU support	Extensive—supports multiple GPUs	Extensive—strong GPU acceleration	Limited—mainly CPU-based	Good—via TensorFlow backend	Extensive—optimized for GPU and distributed	High—optimized for GPU use
Use cases	Enterprise-scale AI, deep learning,	Research, prototyping, deep learning	Data analysis, classical machine	Quick prototyping, small to medium-	Large-scale deep learning, cloud-based AI	Image recognition, real-time
Key libraries/ extensions	TensorFlow Lite, TensorFlow.js, TensorFlow	TorchVision, PyTorch Lightning	None specific, integrates with Pandas, NumPy	Part of TensorFlow, supports TFRS	Gluon, ONNX (interoperability with other	Caffe Model Zoo
Best for	Production-ready systems, end-to-end ML	Research-focused projects, rapid prototyping	Classical ML tasks, educational use	Beginners in deep learning, rapid model	Large-scale, high-performance applications	Specialized image processing