Author: **Sasa Kolovou (Athena RC)**. License: CC-BY (Creative Commons: Attribution)
Published at https://github.com/clarin-eric/standards on the 21st of March, 2025.

ML Activity	Domain	File types
Data collection, cleaning, tokenization	**Data Preparation**	**Raw data formats:** ● **CSV**, **TSV**, **XLSX**: Tabular data. ● **JSON**, **XML**: Semi-structured data formats. ● **AVRO**, **PARQUET**, **ORC**: Optimized for big data frameworks like Hadoop and Spark (more specialized for big data systems and analytics platforms).
		Image/audio/video data: ● **JPEG**, **PNG**, **BMP**: Images. ● **MP4**, **AVI**, **MKV**: Videos. ● **WAV**, **MP3**, **FLAC**: Audio files
		Text data: ● **TXT**, **DOCX**, **PDF**: Plain and formatted text. ● **JSONL**: Line-delimited JSON, used for NLP datasets.
		Data annotations/labels: ● **COCO JSON**, **PASCAL VOC XML**, **YOLO TXT**: Annotation formats for computer vision. ● **BIO**, **CONLL-U**: Annotation formats for NLP tasks.
Pretraining/ fine-tuning, validation	**Model Training**	**Configuration files:** ● **YAML**, **JSON**, **INI**: Hyperparameters, training configurations. ● **TOML**: A human-readable configuration format (.toml).
		Checkpoints and logs: ● **HDF5**: Model weights (e.g., TensorFlow/Keras models). ● **PT**, **PTH**: PyTorch model checkpoints.

		● **PB**: Protocol Buffers for TensorFlow models. ● **ONNX**: Interchangeable format for trained models (.onnx file extension). ● **LOG**, **TXT**: Training logs and performance metrics. ● **TFRecord**: TensorFlow's format for serialized training examples (.tfrecord file extension). ● **Safetensors:** They are a file format for efficiently serialising and loading models with billions of parameters without the vulnerabilities found in pickle.
		Preprocessed datasets: • **NPY**, **NPZ**: NumPy array formats. • **TFRecord**, **LMDB**: Serialized and database-like formats for efficiency. • **Pickle (PKL)**: Serialized Python objects for data pipelines or model weights.
Deployment, inference pipelines	**Model Exchange**	**Model formats:** ● **ONNX**: Open Neural Network Exchange format for interoperability. ● **HDF5**, **PB**, **PT**, **PTH**, **JOBLIB**: Framework-specific formats for sharing models. ● **PMML**: Predictive Model Markup Language for statistical and machine learning models. ● **CoreML**: For Apple's ecosystem. ● **TFLite**: TensorFlow Lite for mobile and embedded devices. ● **TensorRT**: NVIDIA's format for optimized deployment.
		Containerized models: ● **Docker images**: To encapsulate model-serving environments. ● **ZIP**, **TAR.GZ**: Bundles of models, configurations, and dependencies.
		Deployment configurations: • **YAML, JSON**: For APIs or container apps (e.g., Kubernetes, Docker Compose). • **BentoML Bundles**, **MLflow models**: Framework-specific model packaging formats. • **Serialized pipeline formats**: SKLearn pipelines saved as Pickle or Joblib files.

---*******--

LLM Activity	**Domain**	**File types**
Data collection, cleaning, tokenization	**Data Preparation**	**Raw data formats:** ● **CSV**, **TSV**, **XLSX**: Tabular data. ● **JSON**, **XML**: Semi-structured data formats.
		Preprocessed Data: ● **Tokenized Data**: NPY, NPZ (NumPy arrays of tokenized sequences). ● **TFRecord**: Efficient format for large datasets. ● **Pickle (PKL)**: Serialized tokenized datasets (with caution for portability).
		Custom Formats: ● Hugging Face datasets library supports Arrow (**.arrow**) and Parquet files (**.parquet**).
		Annotation Formats: ● **JSONL** for NLP fine-tuning (e.g., prompt-response pairs for GPT-style models). ● **CONLL-U** for sequence labeling (NER, POS tagging).
Pretraining/ fine-tuning, validation	**Model Training**	**Configuration files:** ● **YAML**, **JSON**: Hyperparameters, tokenizer settings, training configurations.
		Intermediate Outputs: ● **Model Weights**: PT, PTH, PB (PyTorch), HDF5 (TensorFlow/Keras/PyTorch). ● **Tokenizer States**: JSON files for BERT, GPT, and SentencePiece tokenizers. ● **Logs/Checkpoints**: TXT, CSV, TensorBoard logs, or MLflow files.

		Custom Datasets: ● Hugging Face datasets (Arrow, JSONL, Parquet) or TFRecord.
Deployment, inference pipelines	**Model Exchange**	**Model formats:** ● **ONNX**: Open Neural Network Exchange format for interoperability. ● **PT**, **PTH, PB**: PyTorch-specific weights. ● **HF:** stands for Hugging Face's Transformers format. It is a binary format that stores the model's parameters in a compressed format. ● **HDF5**: .h5 or .hdf5 file extension ● **SavedModel**: TensorFlow weights (SavedModel is stored in a directory saved_model.pb finary file for the model, saved_model.pbtxt a human readable version of the same file, and also we have two folders named *variables* and *assets*). ● **TFLite**: Optimized models for mobile (.tflite). ● **GGML (Glorot/Gated Gremlin MLmodel)**: Efficient, quantized model format for lightweight inference (e.g., for llama.cpp, gpt4all). Uses the **.bin** file extension. ● **GGUF (Glorot/Gated Gremlin Updatable Format):** updatable version of GGML that allows for fine-tuning or updating the model parameters.
		Tokenizer Files: ● JSON, SentencePiece models (.model + .vocab).
		Metadata and Deployment Configs: ● JSON, YAML for describing the model and tokenizer settings.
		Containerized models or Bundled Formats: ● **Docker images**: To encapsulate model-serving environments. ● **MLflow Model Bundles**: for deployment. ● Hugging Face Transformers repository format: Includes model (safetensors by default, model.bin), tokenizer (tokenizer.json), and configuration (config.json). ● BentoML format for LLM deployment.