Overview on a human-centric interactive ML process for teaching ML in K-12

C. Gresse von Wangenheim

A.von Wangenheim

Working Paper	WP_GQS_01_2021_v10
Status	final
Publication	Public

Software Quality Group - GQS
National Institute for Research and Technology on Digital Convergence - INCOD
Department of Informatics and Statistics - INE
Federal University of Santa Catarina - UFSC
88049-200 Florianópolis - SC
Brazil

Teaching the development of adoptable and usable ML systems for human needs while keeping the user at the center of the entire development cycle is typically aimed at Human-centered Machine Learning (HCML) (Gillies et al. 2016)(Fails & Olsen, 2003). While on the other hand, the application of Interactive Machine Learning (IML) describes an interaction paradigm in which a humans are engaged in a tight interaction loop of iteratively modifying data and/or models, parameters in order to improve model performance (Amershi et al., 2014)(Ramos et al., 2020)(Dudley and Kristensson, 2018). Employing interactive machine learning has demonstrated several benefits as by training supervised learning algorithms using a high-level interface, users (even those without programming or ML expertise) can quickly build and refine ML systems. It also allows users to freely change the behaviour of the system in an exploratory manner. Building a ML application in a human-centric and interactive manner is an iterative process that requires students to execute a sequence of steps as for example presented in Table 1 (Amazon 2019; Amershi et al. 2019; Mathewson 2019; Watanabe et al. 2019, Fiebrink and Gillies, 2018).

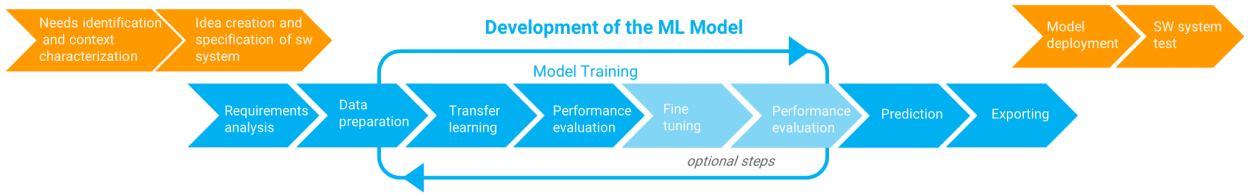

Figure 1. Overview on the human centric iterative ML process

In order to assure the development of useful intelligent solutions, this also incorporates elements of Design Thinking especially when teaching the development of ML models on the create stage as part of the use - modify-create cycle.

Table 1. Description of a human-centric ML development process

Phase	Steps	Description	Output(s)
Needs identification and characterization of the context	**Identification of needs**	Following the design thinking approach, the need for the software system is identified	Descrição da necessidade
	Characterization of the context	The context in which the software system will be applied is characterized in terms of target audience, device(s) and usage environment(s).	Caracterização do contexto (público alvo, ambiente de uso)
Idea creation and specification of the intelligent system	**Idea creation**	Based on this characterization, a solution is idealized.	Descrição do aplicativo proposto

	Specification of the intelligent system	The task(s) and requirements with respect to the software system are analyzed.	Requisitos do sistema inteligente
Requirements analysis of the ML model		During this stage, the main objective of the DL model and its target features are specified. This also includes the characterization of the inputs and expected outputs, specifying the problem.	Análise do modelo de *Deep Learning* Análise de implantação
Data preparation	**Data collection and cleaning**	During data collection, potentially available datasets are identified and/or data is collected. The data is cleaned, which involves removing inaccurate or noisy records from the dataset. The data is also standardized regarding file format and size (file format and size, p.ex. .jpeg and 224x224 pixel)	Dataset with cleaned and standardized images saved on HD or google drive.
	Data labeling	Data labeling involves the assignment of ground truth labels to each record for supervised learning.	Labeled dataset saved in separate directories (one for each class) on HD or google drive.
	Data pre-processing	In order to increase the dataset, typically augmentations are performed, such as crop, rotate etc. Batch size is defined. The data set is typically split into a training set to train the model, a validation set to select the best candidate from all models, and a test set to perform an unbiased performance evaluation of the chosen model on unseen data. Data quality verification is performed in order in order to verify if the objectives can be achieved with the given quality of the available data (incorrect or insufficient, unbalanced or biased data).	Data uploaded to the DL environment (TM or jupyter - dataloader)
Model training and evaluation	**Model learning: Transfer learning**	Then a model is built or more typically chosen from well-known models that have been proven effective in comparable problems or domains. Transfer learning is about leveraging feature representations from a pre-trained model, that are usually trained on massive datasets that are a standard benchmark (such as ImageNet) reusing the weights obtained in other tasks. Including the pre-trained models in a new model leads to lower training time and lower generalization error. It is particularly very useful when you have a small training dataset. In this case, you can, for example, use the weights from the pre-trained models to initialize the weights of the new model. Defining training routines involves setting the learning rate and number of epochs as well as performance metrics.	Selection of model architecture/pre-trained model Specification of training parameters Training execution
	Performance evaluation (TL)	The trained model is evaluated based on information on the loss/error rate during training, as well as specific performance metrics depending on the specific DL task to be achieved, such as accuracy, mean absolute error, etc.	Analysis of loss/error rate Analysis of task-specific performance metrics
	Model learning: Fine-tuning	Typically models are trained in two steps: first by transfer learning and then fine-tuning optimizing hyperparameters to improve performance. Fine-tuning is an optional step in transfer learning that will usually improve the performance of the model. Fine-tuning is done by unfreezing the base model or part of it and training the entire model again on the whole dataset at a very low learning rate. The low learning rate will increase the performance of the model on the new dataset while preventing overfitting.	Identification of appropriate learning rate Unfreeze of weights Train
	Performance evaluation (FT)	The trained model is evaluated based on information on the loss/error rate during training, as well as specific performance metrics depending on the specific DL task to be achieved, such as accuracy, mean absolute error, etc.	Analysis of loss/error rate Analysis of task-specific performance metrics
Prediction		The model is tested with new data in order to obtain an approximation of how the model will perform in the real world.	Testing with new data

Model export	The model is exported in order to enable its integration into a software system. Model export can be either in the mode specific format or by using open formats such as ONNX (onnx.ai) built to represent diverse types of machine learning models.	DL model exported (tensorflow or specific model or using ONNX)
Model deployment	During the production/deployment phase, the model is deployed into a software system to create a usable system and apply it to new incoming events in real-time.	Implantação do modelo de ML em app SW system with integrated DL model (App Inventor app .aia)
SW system test	Usability test/ talvez performance test?	Avaliação de usabilidade

ML models are typically developed on specific development platforms such as Google Teachable Machine or Jupyter Notebooks and once trained with acceptable performance are exported. Once exported they can be deployed as part of software systems or mobile applications using the specific deployment platform, as, for example, in K-12 block-based programming environments, including Scratch or App Inventor (Marques et al., 2020).

References

Amershi, S. et al. (2019). Software Engineering for Machine Learning: A Case Study. In *Proc. of the 41st International Conference on Software Engineering: Software Engineering in Practice,* IEEE Press, 291–300.

Amazon. (2019). *Amazon Machine Learning*, AWS Documentation. https://docs.aws.amazon.com/machine-learning/latest/dg/building-machine-learning.html

Dudley, J. J., Kristensson, P. O. (2018). A Review of User Interface Design for Interactive Machine Learning.ACM Trans. Interact. Intelligent Systems ,8(8).

Gillies, M. et al. (2016) Human-Centered Machine Learning. Proc. of the *Conference on Human Factors in Computing Systems*, ACM, San Jose, CA, USA, 3558-3565.

Mathewson, K. W. (2019). A Human-Centered Approach to Interactive Machine Learning. arXiv:1905.06289v1 [cs.HC].

Ramos, G., Meek, C., Simard, P., Suh, J., Ghorashi, S. (2020) Interactive machine teaching: a human-centered approach to building machine-learned models, *Human–Computer Interaction*, 35(5-6).

Watanabe, Y. et al. (2019). *Preliminary Systematic Literature Review of Machine Learning System Development Process*. arXiv:1910.05528 [cs.LG].