

Disease Progression Modeling Workbench 360

Parthasarathy Suryanarayanan¹, Prithwish Chakraborty¹, Piyush Madan¹, Kibichii Bore², William Ogallo², Rachita Chandra¹, Mohamed Ghalwash¹, Italo Buleje¹, Sekou Remy², Sheryans Sethi¹, Shilpa Mahatma¹, Pablo Meyer¹, Jianying Hu¹
¹Center for Computational Health, IBM Research, NY, USA;
²IBM Research, Nairobi, Kenya

1 Background

Disease Progression Modeling (DPM)¹ aims to characterize the progression of a disease and its comorbidities over time using a wide range of analytics models including disease staging², patient trajectory analytics³, prediction⁴, and time-to-event estimations⁵ for key disease-related events. DPM has applications throughout the healthcare ecosystem, from providers (e.g., decision support for patient staging), to payers (e.g., care management), and pharmaceutical companies (e.g., clinical trial enrichment). But the complexity of building effective DPM models can be a roadblock for their rapid experimentation and adoption. Some of this is addressed by standardization of data model and tooling for data analysis and cohort selection⁶. However, there are still unmet needs to facilitate the development of advanced machine learning techniques such as deep learning with additional requirements such as experiment tracking and reproducibility⁷. Furthermore, to accelerate DPM research, a data scientist's available tools should include a framework for deploying models as cloud-ready microservices for rapid prototyping and dissemination⁸.

In this demonstration, we introduce Disease Progression Modeling Workbench 360 (DPM360) open source project (<https://ibm.github.io/DPM360/>). DPM360 is an easy-to-install system to help research and development of DPM models (Figure 1). It manages the entire modeling life cycle, from data analysis (e.g, cohort identification) to machine learning algorithm development and prototyping. DPM360 augments the advantages of data model standardization and tooling (OMOP-CDM, Athena, ATLAS) provided by the widely-adopted OHDSI⁹ initiative, with a powerful machine learning training framework, and a mechanism for rapid prototyping through automatic deployment of models as containerized services into a cloud environment. This enables a quicker and flexible implementation and validation of the models.

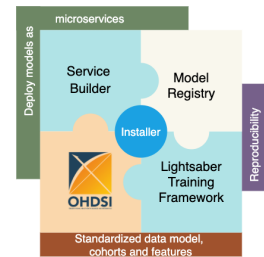


Figure 1: DPM360 Component View

2 Methods

The architecture, shown in Figure 2 has four main components.

(1) **Lightsaber**: an extensible training framework which provides blueprints for the development of disease progression models (DPM). It is designed ground up using state-of-the art open source tools^{10,11} to provide a simple modular and unified model training framework to support some of the common use cases for DPM. Lightsaber contains four key modules:

- *data ingestion* modules to support standardized methods of ingesting data
- *model trainers* to support standardized model training incorporating best practices
- *metrics* to support pre-built DPM problem specific model evaluation
- in-built model tracking and support for post-hoc model evaluation by integrating with a *Model Registry*.

Users can select specific modules and integrate them into their modeling workflow. Lightsaber also comes with a reusable library of state-of-the-art machine and deep learning algorithms for DPM (e.g. LSTM¹² for in-hospital mortality predictions).

Lightsaber integrates naturally with ATLAS using a client called Lightsaber Client for ATLAS (*LCA*), enabling automated extraction of features from the **OMOP CDM** model, thus complementing the ease and flexibility of defining standardized cohorts using ATLAS graphical user interface with the ability to quickly develop deep learning algorithms for DPM in Lightsaber using Python. *LCA* can be configured with the cohort details, covariate settings, model training settings for Lightsaber to extract the right set of features in formats currently supported in the OHDSI stack

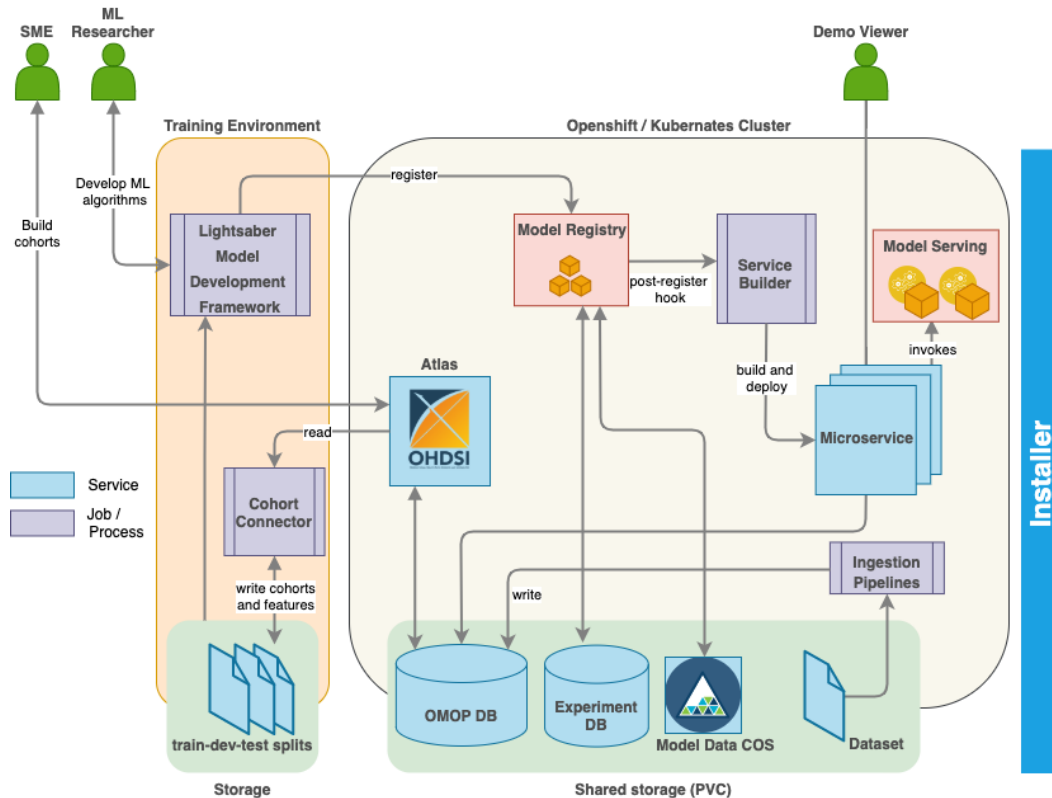


Figure 2: DPM360 Architecture

(see [FeatureExtraction](#) and [PatientLevelPrediction](#) R packages via the [Rpy2](#) interface). Additionally, the *LCA* uses custom queries and algorithms to extract and transform complex time series features into formats required for DPM in Lightsaber. For each feature extraction process, a YAML configuration file is automatically generated. This file specifies outcomes, covariate types, and file locations of the extracted feature files. Thus, Lightsaber allows a user to concentrate just on the logic of their model as it takes care of the rest.

(2) Tracking provenance of all aspects of model building is essential for trust and reproducibility - thus experiments ran using Lightsaber are automatically tracked in a **Model Registry** including model parameters, problem specific metrics, and model binaries allowing the identification of algorithms and parameters that result in the best model performance.

(3) The **Service Builder** component automatically converts registered models in *Model Registry* into microservices¹³, through the usage of hooks that listen for production ready models in the registry and thereafter start the model packaging execution pipeline. The pipeline includes extraction of model and its dependencies from the registry, containerization, and deployment in the target cluster ([Kubernetes](#) or [OpenShift](#)). Upon successful model deployment, a callback function updates model metadata in the registry with deployment status and model access endpoint. Using this endpoint, potential users (data scientist or product manager) can interact with the model, now deployed as a microservice, though a [Swagger](#) based interface.

(4) The **Installer** component installs the fully functional DPM360, including OHDSI tools, *Model Registry*, and *Service Builder* into a Kubernetes or OpenShift cluster using [Helm charts](#). Each of these components are run as services within the cluster. The implementation also uses [Persistent Volume Claims](#) to persist the data (e.g: model artifacts, ATLAS database files etc.).

3 Results

A detailed description of the example use cases, architecture and road map is available in our public GitHub repository at <https://ibm.github.io/DPM360/> which contains our initial implementation. To verify the integration of the overall system, we reproduced the results from clinical prediction benchmarks from MIMIC III¹⁴. Specifically, we defined and extracted cohorts related to in-hospital mortality in critical care settings from an OMOP CDM version of the MIMIC III dataset¹⁵. Using ATLAS, we defined our target cohort consisting of adult patients who have been hospitalized for the first time for at least two days, and who have at least one measurement recorded within the first 48 hours, and our outcome cohort of adult patients who died in hospital within 30 days of the first admission. We subsequently used the LCA to extract the features for the prediction task, and then trained a model using the Lightsaber framework. We obtained comparable metrics to the benchmark, thus proving the soundness of the overall system. Also, independently, the Lightsaber component has been successfully used for rapid experimentation and analysis of real-world data^{3,4}.

4 Conclusion

In the coming months, we plan to execute our published road-map for DPM360 and report on its effectiveness at improving collaborative research and rapid commercialization from several ongoing DPM collaborations.

Acknowledgements

We would like to thank Divya Pathak and Daby Sow for their guidance, and Sundar Saranathan for his contributions.

References

1. Wang X, Qi J, Yang Y, Yang P. A Survey of Disease Progression Modeling Techniques for Alzheimer’s Diseases. In: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN). vol. 1. IEEE; 2019. p. 1237–1242.
2. Sun Z, Ghosh S, Li Y, Cheng Y, Mohan A, Sampaio C, et al. A probabilistic disease progression modeling approach and its application to integrated Huntington’s disease observational data. *JAMIA open*. 2019;2(1):123–130.
3. Dey S, Bose A, Chakraborty P, Ghalwash M, Saenz AG, Ultro F, et al.. Impact of Clinical and Genomic Factors on SARS-CoV2 Disease Severity. Cold Spring Harbor Laboratory Press; 2021. Accepted at AMIA 2021 Annual Symposium. To appear, early access at <https://www.medrxiv.org/content/10.1101/2021.03.15.21253549v1>.
4. Chakraborty P, Codella J, Madan P, Li Y, Huang H, Park Y, et al.. Blending Knowledge in Deep Recurrent Networks for Adverse Event Prediction at Hospital Discharge ; 2021. Presented at AMIA 2021 Virtual Informatics Summit. To appear, early access at <https://arxiv.org/abs/2104.04377>.
5. Liu B, Li Y, Sun Z, Ghosh S, Ng K. Early prediction of diabetes complications from electronic health records: A multi-task survival analysis approach. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32; 2018. .
6. Hripcsak G, Ryan PB, Duke JD, Shah NH, Park RW, Huser V, et al. Characterizing treatment pathways at scale using the OHDSI network. *Proceedings of the National Academy of Sciences*. 2016;113(27):7329–7336.
7. McDermott M, Wang S, Marinsek N, Ranganath R, Ghassemi M, Foschini L. Reproducibility in machine learning for health. *arXiv preprint arXiv:190701463*. 2019.
8. Mäkinen S, Skogström H, Laaksonen E, Mikkonen T. Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? *arXiv preprint arXiv:210308942*. 2021.
9. Hripcsak G, Duke JD, Shah NH, Reich CG, Huser V, Schuemie MJ, et al. Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers. *Studies in health technology and informatics*. 2015;216:574.

10. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011;12:2825–2830.
11. Falcon W, et al. PyTorch Lightning. GitHub Note: <https://github.com/PyTorchLightning/pytorch-lightning>. 2019;3.
12. Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. *Neural computation*. 2000;12(10):2451–2471.
13. Dragoni N, Giallorenzo S, Lafuente AL, Mazzara M, Montesi F, Mustafin R, et al. Microservices: yesterday, today, and tomorrow. *Present and ulterior software engineering*. 2017:195–216.
14. Harutyunyan H, Khachatrian H, Kale DC, Ver Steeg G, Galstyan A. Multitask learning and benchmarking with clinical time series data. *Scientific Data*. 2019 Jun;6(1). Available from: <http://dx.doi.org/10.1038/s41597-019-0103-9>.
15. Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. *Scientific data*. 2016;3(1):1–9.