

PluMA 2.0 Machine Learning Pipeline Refactor

Student: Alana Barned

Mentor: Dr. Trevor Cickovski, PhD

Instructor/Faculty: Dr. Masoud Sadjadi, PhD, Florida International University

PROBLEM

The software pipeline is a standard analysis tool in bioinformatics. "Pipeline" is a generic term for any sequential software control flow, where the output of one program feeds as input to the next. The inherently interdisciplinary bioinformatics landscape currently contains a great deal of pipelines, often implemented as single large scripts or functions. This takes away opportunities for bioinformaticians to reuse internal stages of these pipelines, often resulting in time reinventing the wheel that could be more optimally used to advance research.

CURRENT SYSTEM

PluMA attempts to fix this problem by providing an interface where bioinformaticians can prototype algorithms in one of many supported languages and wrap them as plugins, easily insertable and removable from other users' pipelines without knowledge of underlying implementation. Collaborators with the Bioinformatics Research Group (BioRG) have developed several machine learning pipelines that require software refactoring for compatibility with PluMA, implemented primarily with Python.

REQUIREMENTS

The goal of this project is to properly refactor these pipelines as a series of plugins using software design patterns, each of which will then be redistributed open-source and usable in other pipelines, enhancing the impact of the work of our collaborators and facilitating community-based development

SYSTEM DESIGN

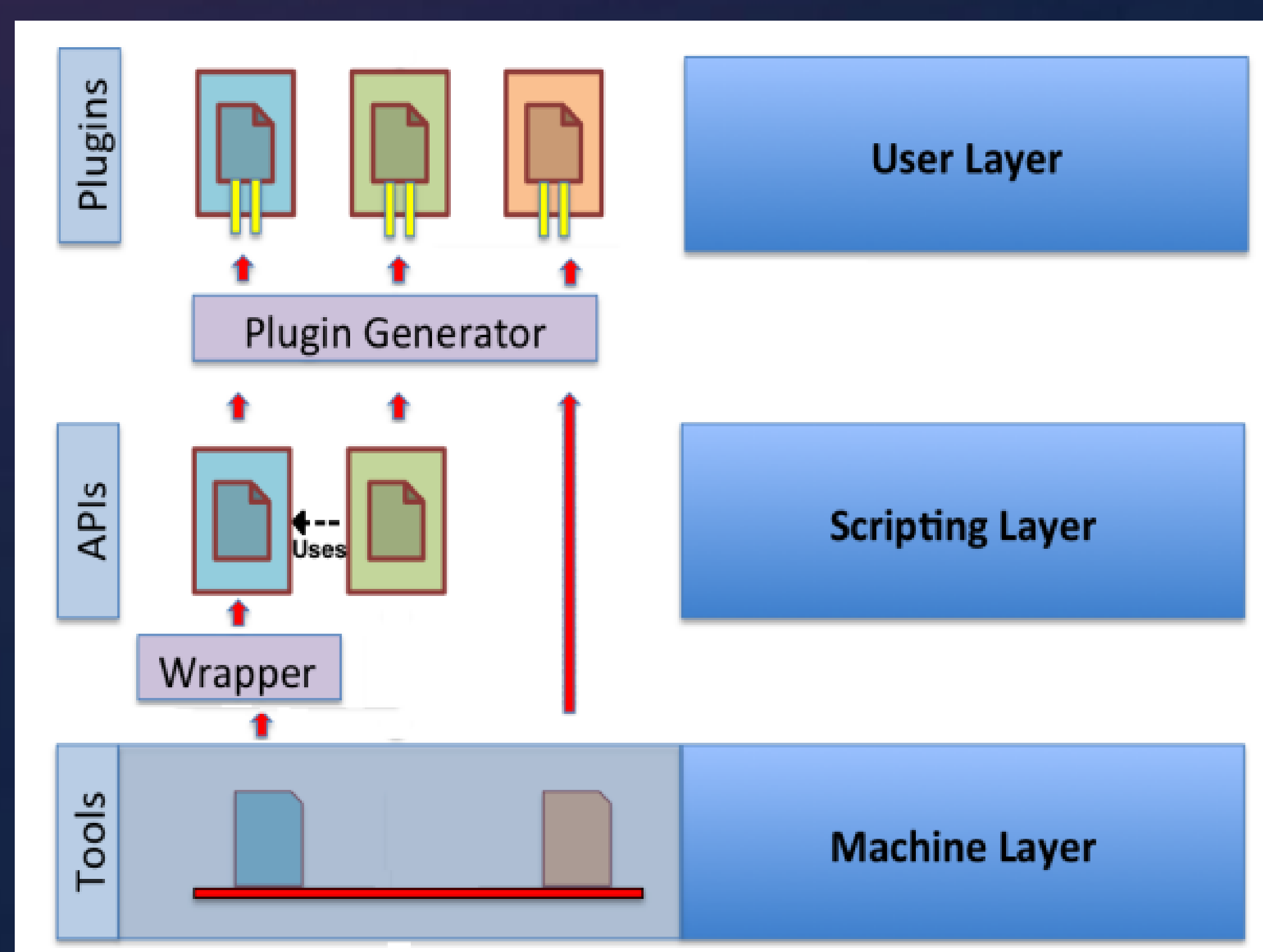


Figure 1: Conceptual design of PLUMA. Users interact with the software through the user layer, where they can assemble pipelines using plugin extensions in a variety of languages.

OBJECT DESIGN

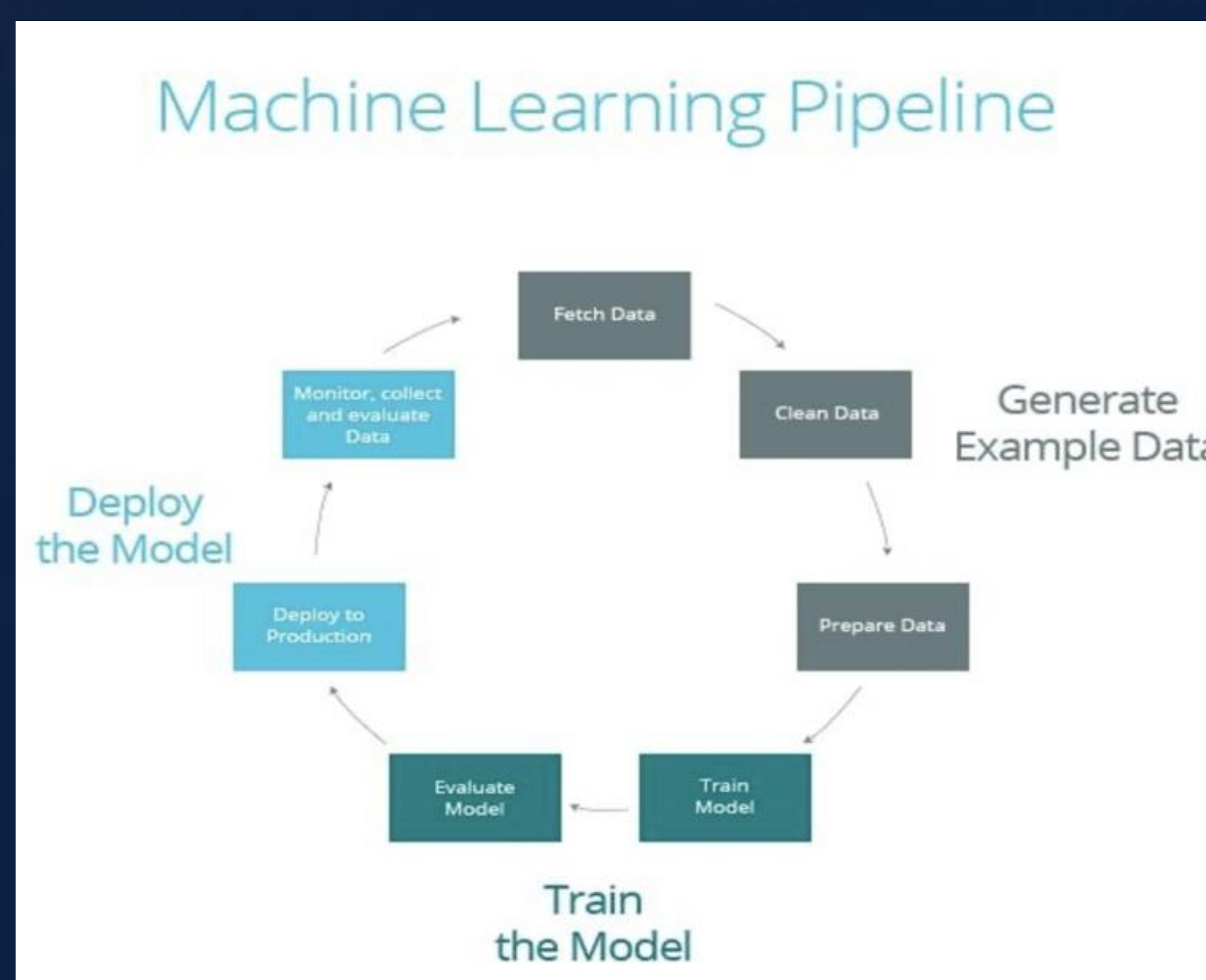


Figure 2: Steps involved in the machine learning pipeline.

IMPLEMENTATION



VERIFICATION

Files were tested in a Linux environment and with the utilization of several local development tools. All successfully running pipelines were tested and verified with appropriate data. The team collaborated on dependency updates, large-scale refactors, and git merges to ensure the project is properly documented and kept up-to-date.

SUMMARY

My role in this project was to document all dependencies and create outlines for all the files within the pipeline assigned to me - ml_flood. I was also required to convert all non-python files to python files and refactor and debug the code to ensure it ran on a Linux machine. All files would then be modularized to make it easier to break them down into several plugins. The aim was to have a fully functioning pipeline that was essentially built from several plugins which is the goal of PluMA.

REFERENCES

Resources were provided by Dr. Trevor Cickovski of the Bioinformatics Research Group at Florida International University.

- <https://biorg.cis.fiu.edu/pluma/>
- <https://github.com/movingpictures83/Flood/tree/main>

The material presented in this poster is based upon the work supported by the FIU Knight Foundation School of Computing and Information Science. We thank Dr. Trevor Cickovski for his assistance and mentorship that we received throughout the senior design project.