Building and Deploying Reproducible Machine Learning Pipelines

By Maryam Najafian

Reproducibility of modeling is a problem that exists for any machine learning practitioner. The consequences of an irreproducible model can include significant financial costs, lost time, and even loss of personal reputation (if results prove unable to be replicated). This page is developed to share what I found necessary to create reproducible codes that enable smooth collaborations.

- Bad process: Problem statement > code solution
- Good process: Problem statement > mental model > code solution

1. Configuration files in python

Here are some approaches to create a configuration file for your project:

- https://www.hackerearth.com/practice/notes/samarthbhargav/a-design-pattern-for-configuration-management-in-python/
- https://hackersandslackers.com/simplify-your-python-projects-configuration/
- https://martin-thoma.com/configuration-files-in-python/

2. Introduction to creating a reproducible ML/AI pipeline

The following articles provide an overview of the different stages involved in creating a reproducible ML/AI pipeline.

- Building and Deploying a Reproducible Machine Learning Pipeline - article
- Building a Reproducible Machine Learning Pipeline - long article
- Reproducible Machine Learning - presentation, Kaggle
- The Machine Learning Reproducibility Crisis - article, by Google developer
- A systems perspective to reproducibility in Production Machine Learning
- Hidden technical debt in machine learning systems - Google

2.1 Scikit-Learn and Sklearn Pipeline (Recommended)

Scikit-Learn object consists of Transformer, Predictor, and Pipeline:

Transformers - class that have fit and transform method, it transforms data:

- Scalers
- Feature selectors
- One hot encoders

Predictor - class that has fit and predict methods, it fits and predicts:

- Any ML algorithm like SVM, Random Forest, NNs

Pipeline - class that allows you to list and run transformations and predictors in sequence:

- All steps should be transformers except the last one
- Last step should be a predictor

Advantages

- Can be tested, versioned, tracked and controlled
- Can build future models on top
- Good software developer practice
- Leverages the power of acknowledged Sklearn API
- Data scientists are already familiar with the pipeline, and leads to reduced over-head
- Engineering steps can be packaged and re-used in future ML models

Disadvantages:

- Requires team of software developers to build it
- Requires team of software developers to maintain it

The following articles provide step-by-step guidance on creating a reproducible pipeline based on the widely popular Scikit-Learn and Sklearn Pipeline.

- Introduction to Scikit-Learn
- Six reasons why I recommend Scikit-Learn
- Why you should learn Scikit-Learn
- If You've Never Used Sklearn's Pipeline Constructor...You're Doing It Wrong
- Deep dive into SKlearn pipelines
- SKlearn pipeline tutorial
- SKlearn pipeline example
- Managing Machine Learning workflows with Sklearn pipelines
- A simple example of pipeline in Machine Learning using SKlearn

2.2 AutoML, Facebook, and Uber Pipelines (FYI)

The following articles provides other architectures used in alternative organizations (I have not used them personally):

- Introducing FBLearner Flow: Facebook's AI backbone
- Scaling Machine Learning as a service: Uber's pipeline

3. Reproducibility in Setting the Seed for Keras

To ensure you have reproducible results after using Keras for training stage, please follow the following guidelines.

- Keras documentation for producing reproducible ML system
- How to get reproducible results in keras, StackOverflow
- Any way to note down or control the random seeds? in git Keras issues
- mnist_cnn.py does not give reproducible results, in git Keras issues

4. Enhancing Code Readability with PEP8 Standard

It is not easy to deal with code that comprises non-standard coding styles and conventions. A great way to achieve some standardization and uniformity is if people use linters. Nobody wants to receive negative comments about their code or send negative comments about other people's code. The best way to avoid it is to use linters before-hands to ensure you have a readable code, with standard format (e.g. PEP 8). In some IDEs like PYCHARM, ATOM you can use *PEP8 linters* and formatters that give you real time feedback and provide your with auto-correction for formatting as you code.

- Why we need PEP8
- How to follow PEP8 standard with examples

5. Unit Testing & Test-Driven Development

Unit Testing is the first level of software testing where the smallest testable parts of a software are tested. This is used to validate that each unit of the software performs as designed.

- How to unittest ML code
- Python testing taxonomy

OOP concepts supported by unittest framework:

- **Test fixture:**
  A test fixture is used as a baseline for running tests to ensure that there is a fixed environment in which tests are run so that results are repeatable.
- **Test case:**
  A test case is a set of conditions which is used to determine whether a system under test works correctly.
- **Test suite:**
  Test suite is a collection of test cases that are used to test a software program to show that it has some specified set of behaviours by executing the aggregated tests together.

Test runner:

A test runner is a component which set up the execution of tests and provides the outcome to the user.

6. Prerequisite before reading this article

Here are the links to refresh your memory on Object Oriented Programming (OOP) in Python before you start your ML and Data Science project:

https://www.youtube.com/playlist?list=PLwv4n_SJu9-P38DrseSKjBJZcwDM8Uucg

https://github.com/ashwin-pajankar/Python-OOP

https://www.youtube.com/playlist?list=PLiHa1s-EL3vjIVdLySA3aBr8bMlW7r_4f

https://www.youtube.com/playlist?list=PLN0iJDXT7K2vB3EGwKpDV-VIylhs3dEV8

Methods for OOP:

https://www.pythonlikeyoumeanit.com/Module4_OOP/Special_Methods.html

Other useful links for BI and data science practitioners:

https://drivendata.github.io/cookiecutter-data-science/
https://www.elastic.co/training/kibana-data-analyst