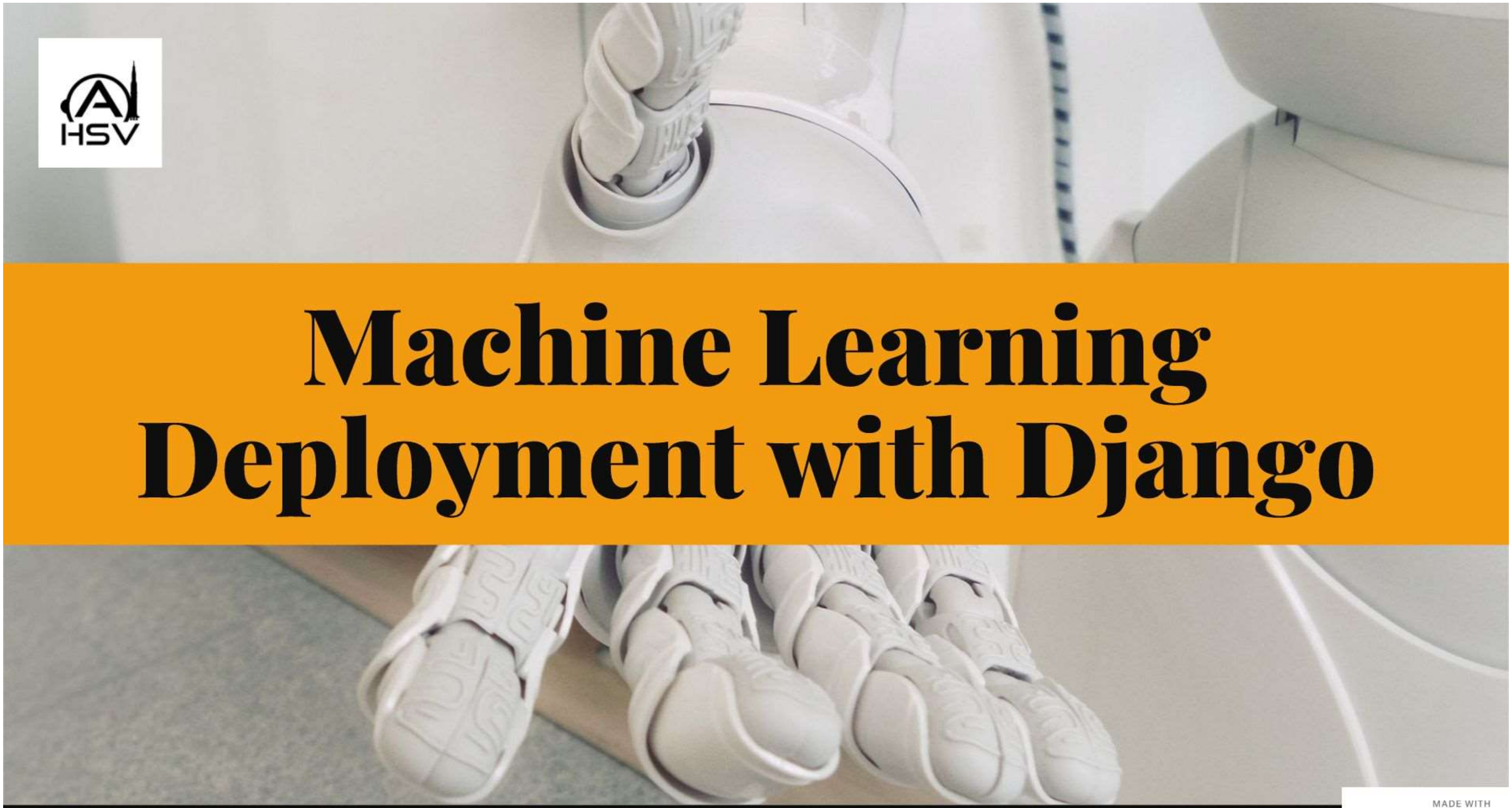
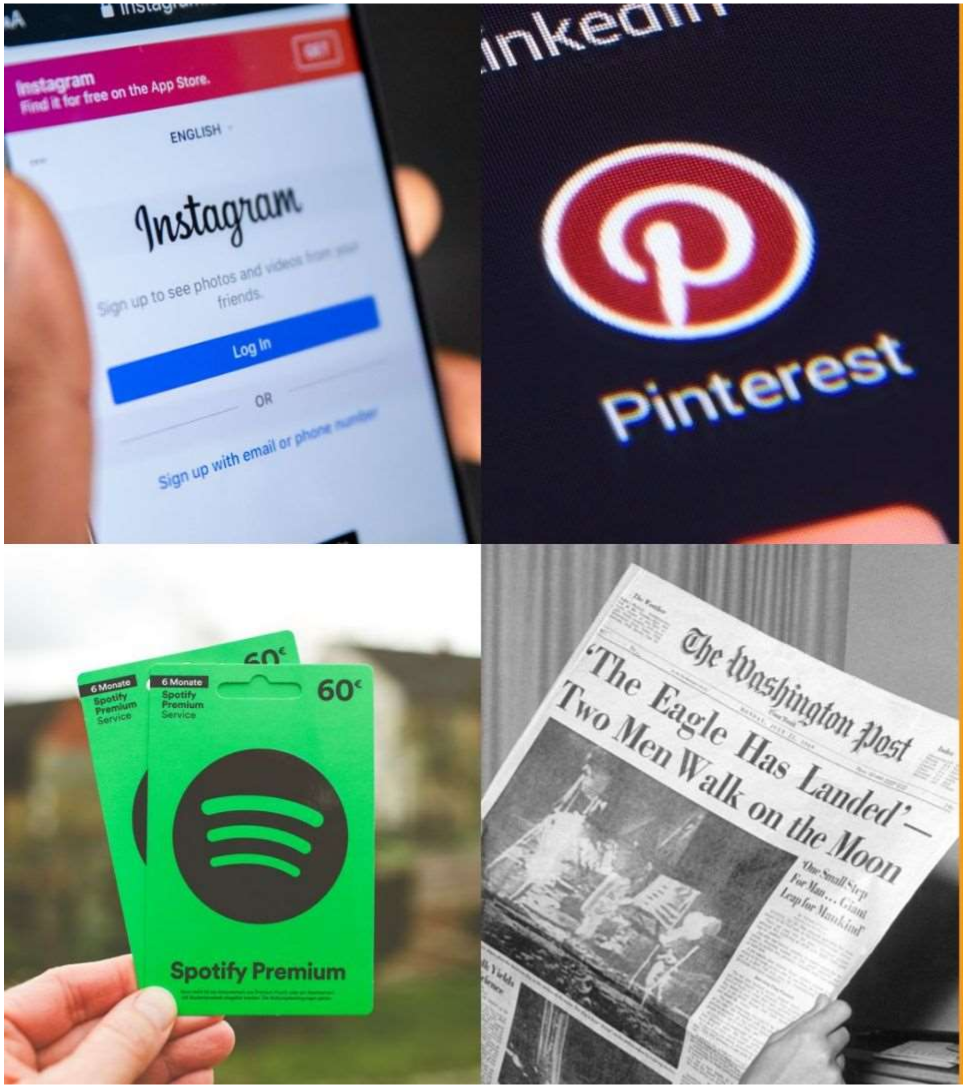


Machine Learning Deployment with Django

https://github.com/HSV-AI

By, Harsha Gandikota


Instagram
Find it for free on the App Store.
ENGLISH
Instagram
Sign up to see photos and videos from your friends.
Log In
OR
Sign up with email or phone number
Linkedin
Pinterest
6 Monate
Spotify Premium Service
60€
Spotify Premium
The Washington Post
'The Eagle Has Landed'—
Two Men Walk on the Moon
'One Small Step For Man... Giant Leap for Mankind'

What is Django ?

The web framework for perfectionists with deadlines.

An Open-Source python framework for web development.

Allows for rapid development of complex database-driven websites.

A high-level Web framework that takes care of much of the hassle of web development.

(No need to constantly reinvent the wheel)

https://djangostars.com/blog/10-popular-sites-made-on-django/

Main differences between Django and Flask

Why Django is more suited towards Enterprise

Django Provides Object-Relational Mapping

Allows for linking of database tables to actual Python classes

Django Provides a Model-View-Controller framework

Allows for a decoupled approach to development

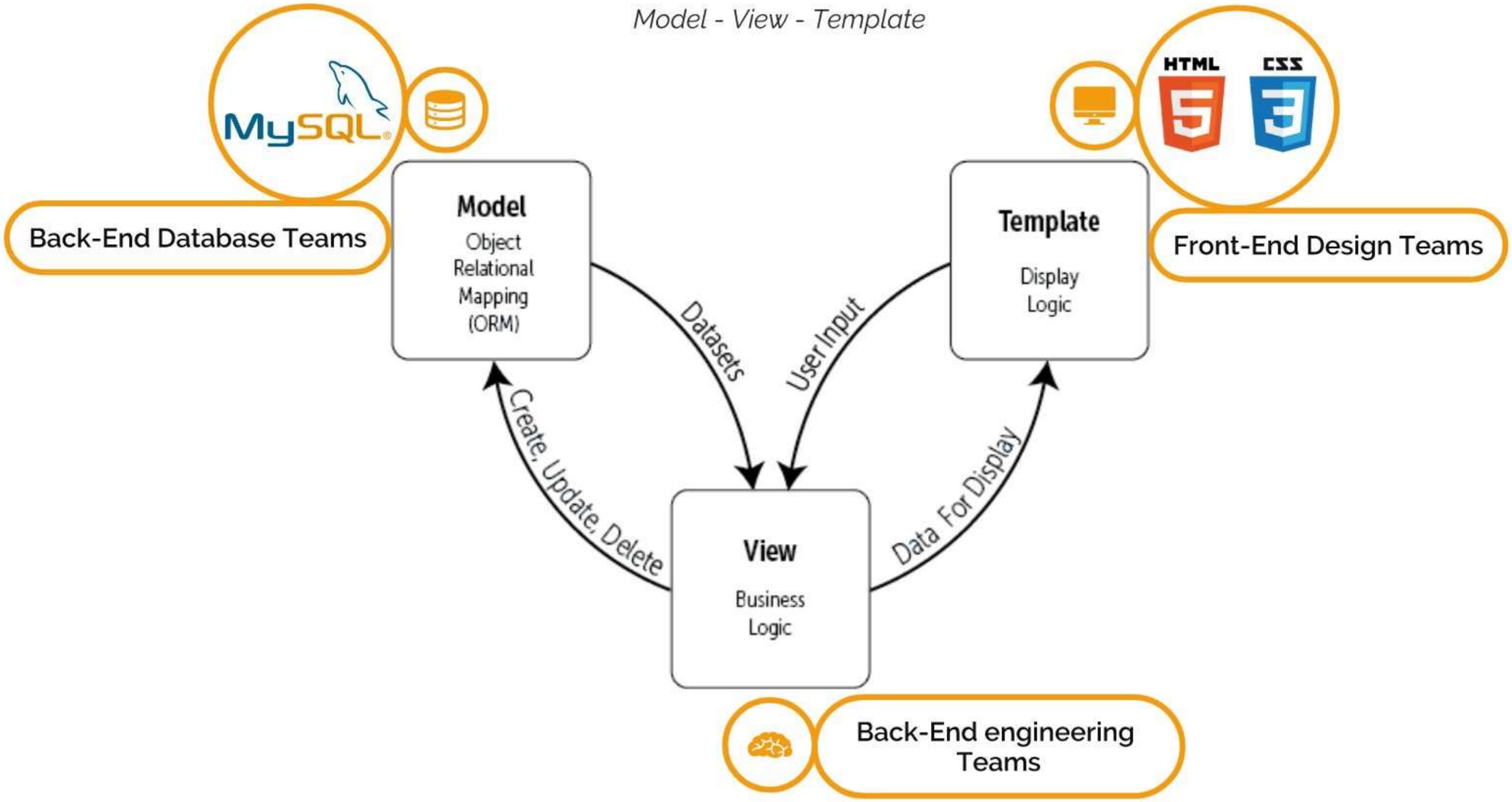
Easier to build complex Multi-Page websites with Django

Flask applications are mostly single page applications, not ideal for enterprise scale

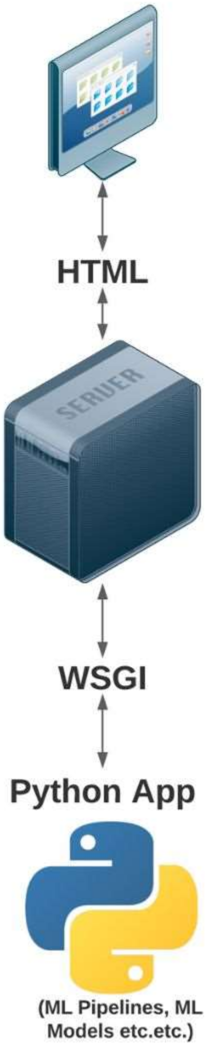
https://medium.com/@SteelKiwiDev/flask-vs-django-how-to-understand-whether-you-need-a-hammer-or-a-toolbox-39b8b3a2e4a5

Django Structure

Model - View - Template



https://djangobook.com/mdj2-django-structure/

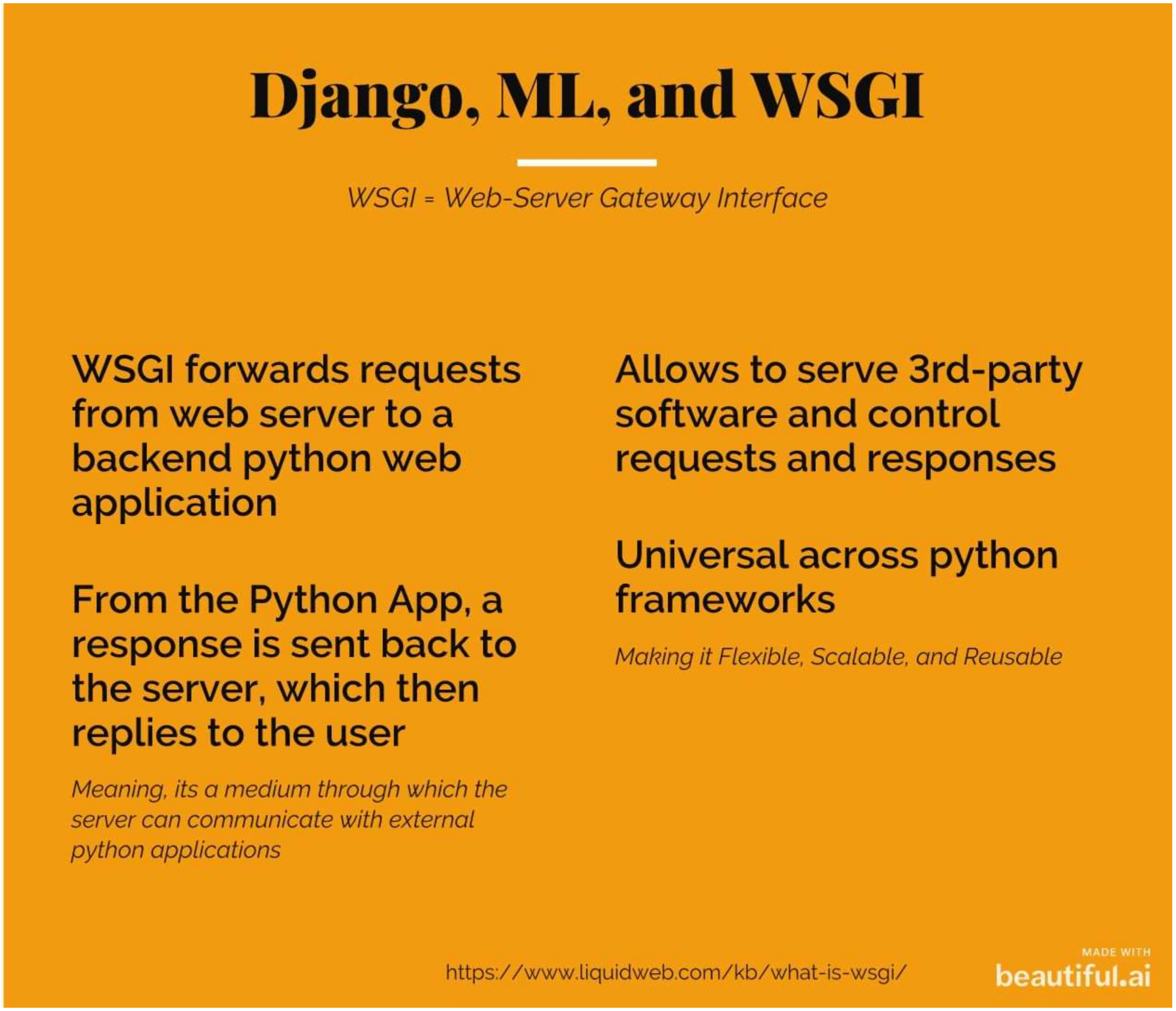


HTML

WSGI

Python App

(ML Pipelines, ML Models etc.etc.)



Django, ML, and WSGI

WSGI = Web-Server Gateway Interface

WSGI forwards requests from web server to a backend python web application

From the Python App, a response is sent back to the server, which then replies to the user

Meaning, its a medium through which the server can communicate with external python applications

Allows to serve 3rd-party software and control requests and responses

Universal across python frameworks

Making it Flexible, Scalable, and Reusable

https://www.liquidweb.com/kb/what-is-wsgi/

Django Rest Framework (DRF)

REST = Representational State Transfer

- DRF is an open source, well-matured Python and Django library intended to help APP developers build sophisticated web APIs.

  Features:

  > Serialization and DeSerialization

  > Authentication

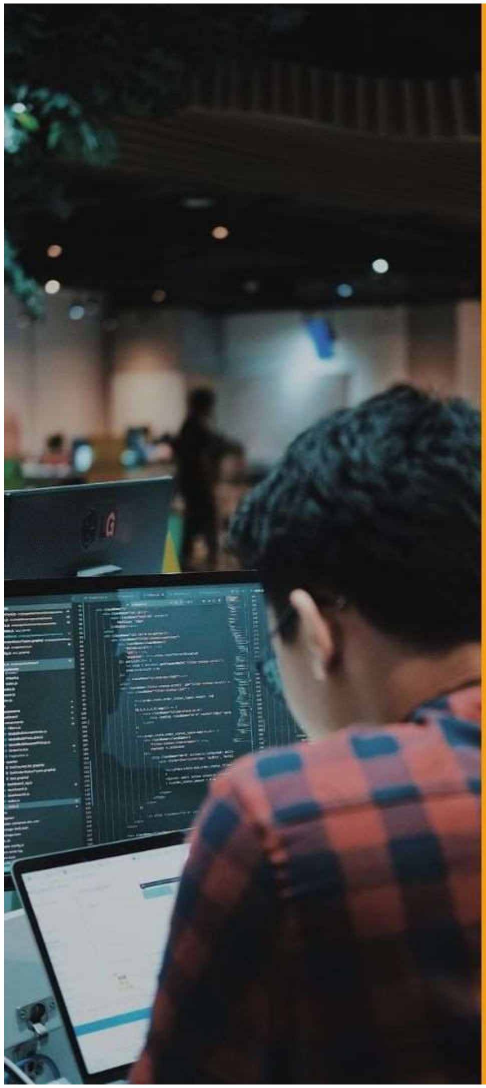
  > Generic In-built Views which prevents having to constantly reinvent the wheel

  > Automatic URL Routing

- DRF's modular, flexible, and customizable architecture makes the development of both simple, turnkey API endpoints and complicated REST constructs possible.

- In regards to ML, allows for an easier and affective way of achieving a **'Model As A Service'**(MaaS) paradigm, which makes Model **Integration** and **Deployment** quick and easy.

https://hub.packtpub.com/which-python-framework-is-best-for-building-restful-apis-django-or-flask/



ML Model Deployment

What to keep in mind...

- **Deployment: The process of taking a trained ML model and making its predictions available to users or other systems**

  Also to provide a **service** that meets the users **needs**. If the user's **needs** are **not** met, deployment is **incomplete**.

- **In order to decide how to deploy a model, you need to understand how end users should interact with the model's predictions.**

  > Will the users expect predictions to be calculated and delivered on a timely manner, or will they expect the predictions be available at any time?

  > How will predictions be returned to the user? Will the user want predictions to be returned and stored in a database, or will they want the results returned in the form of a graph?

https://mlinproduction.com/what-does-it-mean-to-deploy-a-machine-learning-model-deployment-series-01/

ML Model Deployment

Another thing to keep in mind: Creating a ***Proper*** *Interface*

- **Creating a proper interface to interact with our models is just as important(if not more) as the performance of the models themselves.**

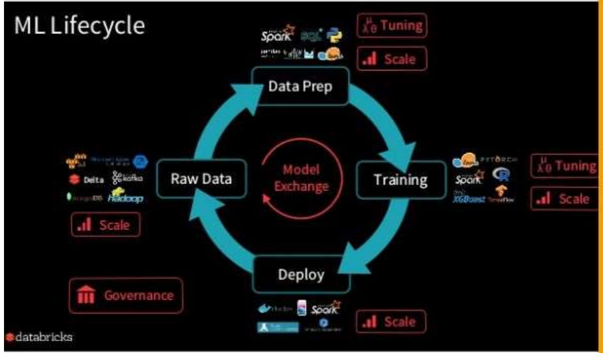
  i.e. Proper and Non-Ambiguous specification of inputs and outputs.

- **It's best if the interfacing is well-defined at the beginning itself.**

  Models can be optimized and improved as time goes on, but changing the interface will become more difficult ...

- **"Creating good interfaces up front will save your machine learning team A LOT of time as you take on additional projects by making deployments automated and repeatable."**

https://mlinproduction.com/software-interfaces-for-machine-learning-deployment-deployment-series-02/



Machine Learning Registry

Keeping track of ML models

- A **machine learning model registry** is a centralized tracking system that stores lineage, versioning, and related metadata for published machine learning models.

  > Registries provide a mechanism to store model metadata

  > Registries make it easier for data science teams to experiment, test and select desired models that can affectively generate predictions.

- **Model Registries also help provide governance to the Machine learning life cycle regardless whatever tools are being used.**

  MLFlow is an open source python model registry that helps with this.

https://mlinproduction.com/model-registries-for-ml-deployment-deployment-series-06/