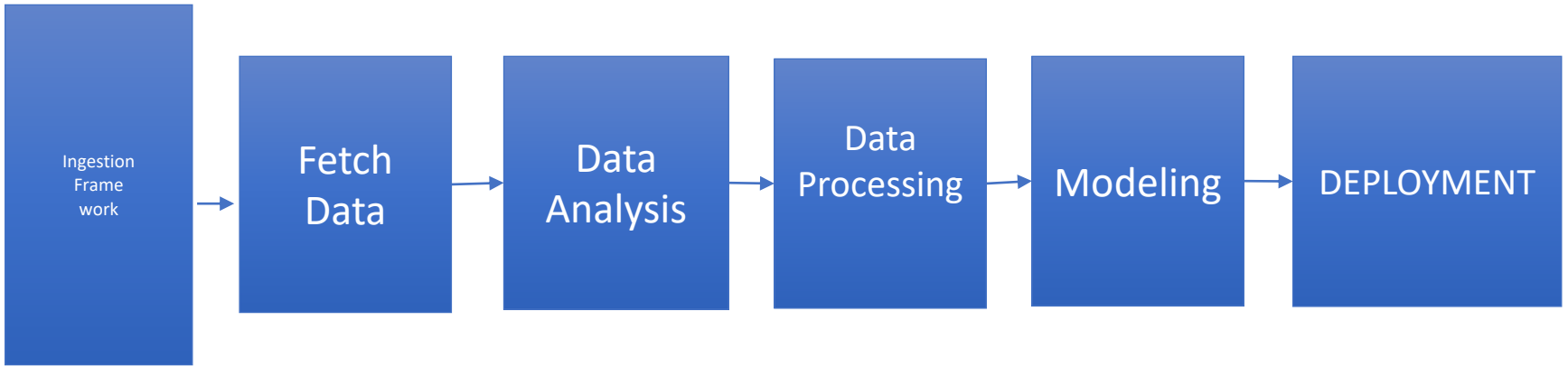
Machine Learning- 1 (Supervised Regression) - Model Deployment using Flask

Phases of ML/DS Project



- Excel
- csv

- Type anomalies
- Missing , Null
- Correlation
- Skewness
- EDA (plots)

- MVT
- Transformation
- Feature Engineering

- Model Selection
- Training
- Hyperparameter tuning
- Ensemble
- Cross validation

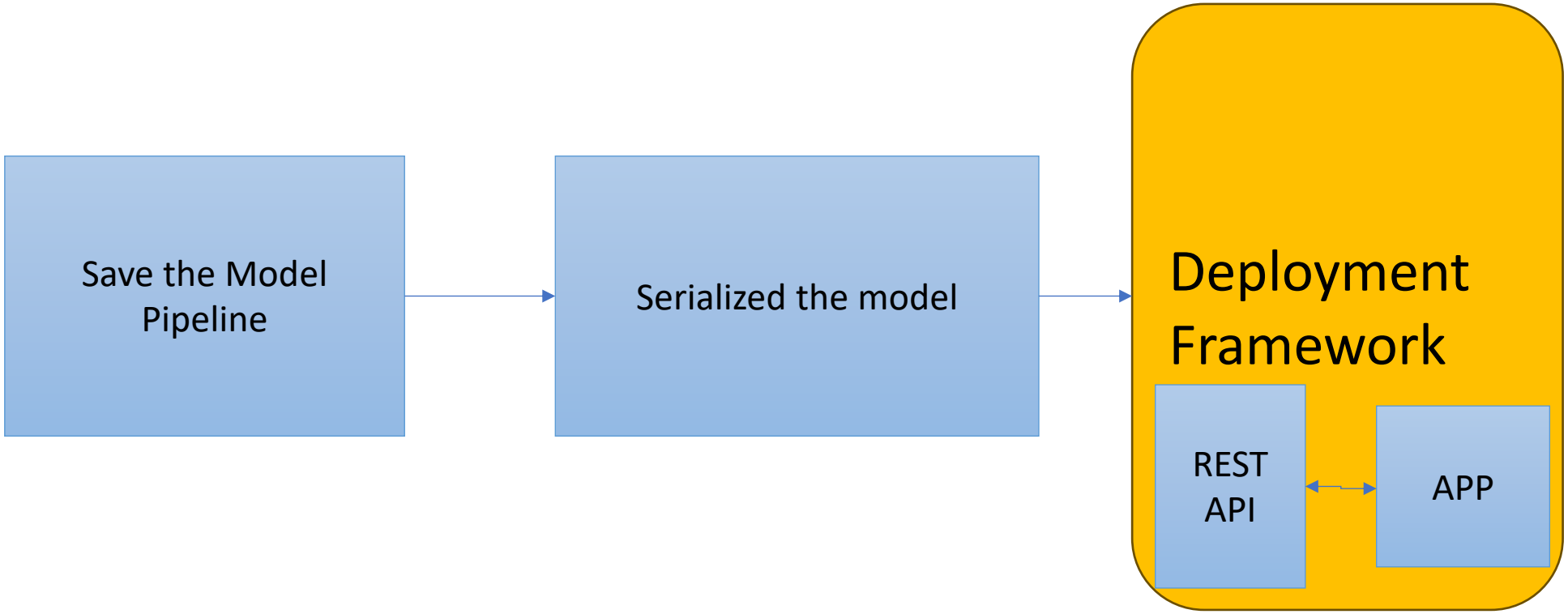
What is Deployment?

- Trained ML model available to others.
- Through web API/REST
- Takes input and returns output

REST

- **Representational state transfer** (**REST**) is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web.
- REST application talks with world wide web

Typical Deployment Approach



Flask as a Python Deployment Framework

Python (micro*)web framework used for building web applications.

It is classified as a microframework because it does not require particular tools or libraries

Key features of Flask

Minimalistic:

Routing

Templating

HTTP Request/Response Handling

Lightweight and Scalable

Python Integration

Pre-requisite for Windows

Install Jupyter notebook (anaconda)

Install Python IDE/code-editor: (Preferred Spyder as it is default with anaconda, PyCharm)

Python version = 3.10

Execution in 3 Steps

Step 1 : Execute '**Train_Deploy**' in Jupyter notebook

Step 2 : Execute '**Test_ Deploy**' in Jupyter notebook

Step 3 : Open project 'ml1_deployment_flask' in (Spyder) IDE and Run file '**app.py**'. Our Model gets hosted at ***http://127.0.0.1:5000/*** ; paste this URL on web browser and enter input values and press 'submit' to get loan_status prediction

Step1: Train_Deploy

Fetches loan_status dataset and build a full pipeline model on it

Final model is saved as pickle file i.e. a serialized binary file

Step2:
Test_Deploy

Generates a sample input feature set (single record)

Fetches the previously saved full pipeline model i.e. pickle file

Predicts target feature using fetched model

Step 3: ml1_deployment_flask : app.py

Uses flask for model deployment

Templates folder stores .html files

In app.py, main function invokes index.html on localhost (default port 5000)

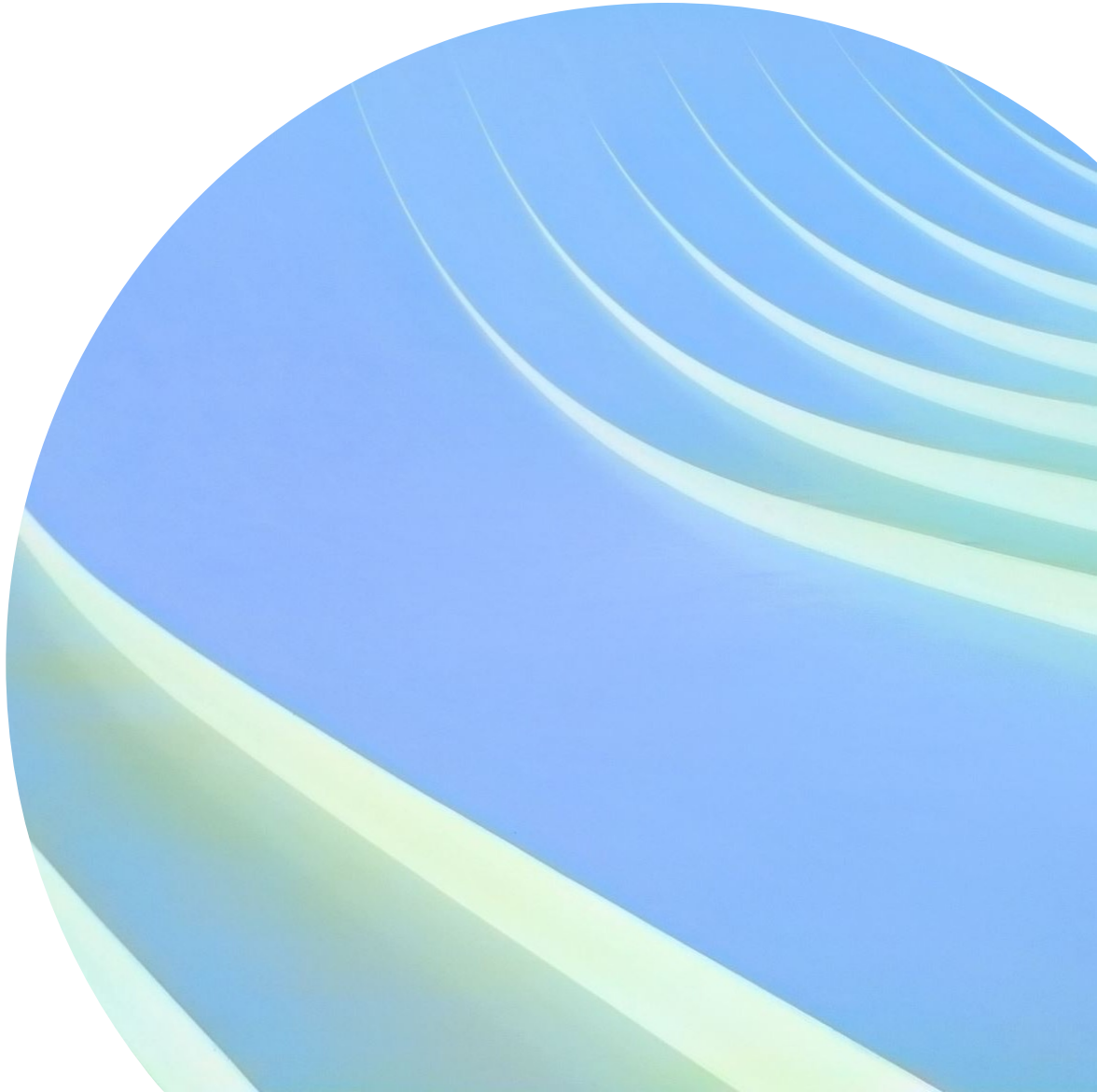
Once form (index.html) is submitted, data is POSTed into fetched model as a feature record and target is predicted (similar to Test_Deploy)

Deployment vs Hosting

Deployment	Hosting
Model accessible via server	Model accessible via web host(domain name) (Internet)
Local/cloud server, localhost	Webserver
Flask, Django, etc.	Web hosting services provider, PaaS
IP address, port	Domain name, Web space, URL
Before Hosting	After Deployment

Hosting Services : PaaS Providers

- PythonAnywhere
- Heroku (Github)
- Streamlit/ FastAPI (no .html requirement)
- Cloud
  - AWS Elastic Beanstalk:
  - Google App Engine



Hosting on **www.pythonanywhere.com**

1. Signup & login : to **www.pythonanywhere.com**
2. Setup virtual environment for the project & install required libraries
3. Create Default web app :
   - (Select ->Manual configuration, Python 3.10)
4. Create Project folder & Upload project files
5. Configure the configuration file & project path
6. Refresh Web app

Hosting Steps on **www.pythonanywhere.com**

A. Create your account & login

B. Create a default web app (on python 3.10)

C. Setup/create virtual environment for the project

1. Console > Bash
2. *mkvirtualenv --python=/usr/bin/python3.10 my-virtualenv*

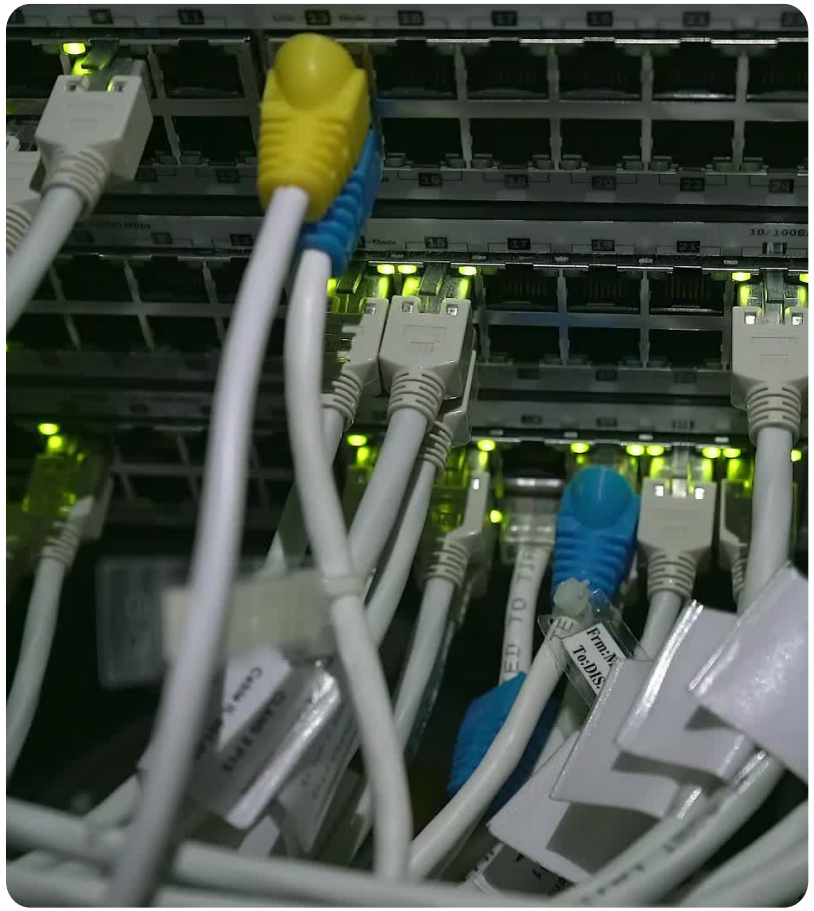
D. 1.Configure the configuration file WSGI file

1. Set up source folder
2. Upload the project files (app.py, full_pipeline.pkl, index.html)
3. Setup for paths for : virtual environment, source directory, working directory

(WEB →

1. Install required libraries on virtual environment console (pip install scikit-learn==1.3.0 Flask==2.3.2 pandas==2.0.3 joblib==1.3.1)

E. Reload web app



Debug Guide

Rebuild local & project with same libraries/packages version (requirements.txt)

Future Study

CI/CD – updating model

ML Ops

Cloud/kuberneties/ docker deployment

Stremlit/

Thank You

For any query mail @ kaustuv.kunal@greatlearning.in