Model Deployment

Model Deployment

- Section Overview
    - General Theory and Concepts
    - Model Persistence Basics
    - Model Deployment as an API

Model Deployment

- General Theory and Concepts
    - When is a model ready for deployment?
    - How often to retrain your model?
    - When to revisit model algorithm choice and assumptions?

Model Deployment

- Model Persistence Basics
    - Quick walkthrough ML steps.
    - Review on saving and loading a fitted model.

Model Deployment

- Model Deployment as an API
  - Save a model as a serialized pickle file.
  - Create a .py script to read in JSON feature data and produce predictions.
  - Use Flask to accept features and return predictions as an API with POST and GET (using Postman).

Model Deployment

- What this section is not:
  - **Not** a comprehensive overview of APIs or web deployment.
  - If you have no experience with APIs or Web Development, this section helps you set up your ML work in a way to pass on to an experienced developer in APIs and Web plications.

Let’s get started!

Model Deployment General Concepts

Model Deployment

- Key Model Deployment Ideas
    - Choosing a Model
    - Purpose of Deployment
    - Performance Expectations
    - Retraining Intervals

Model Deployment

- Choosing a Model
  - Often you will explore multiple models and then compare performance metrics.
  - Consider tradeoffs between model interpretability and performance.
    - For example, are coefficients for features available?

Model Deployment

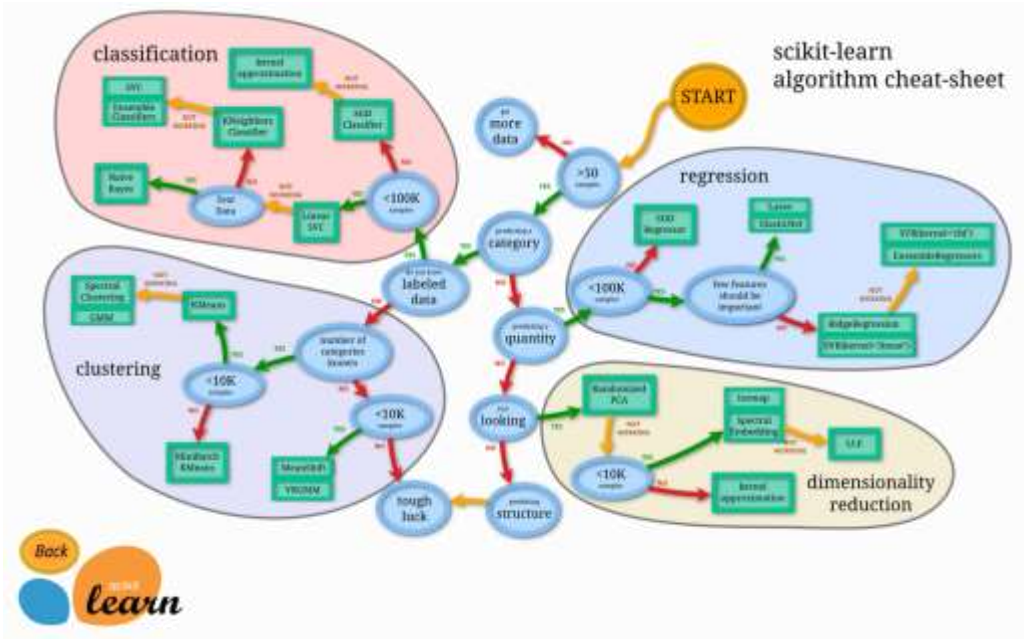
- Choosing a Model
  - Always test out multiple models and remember to perform cross-validation to fairly compare models and perform hyperparameter tuning.

Model Deployment

- Choosing a Model - Scikit-Learn Map
  - Check online for “Choosing the right estimator” for a guided map on which algorithms to first consider.
  - Remember you can always try multiple algorithms and simply compare performance metrics.

Model Deployment

- Choosing a Model - Scikit-Learn Map



PIERIAN DATA

Model Deployment

- Purpose of Deployment
  - Deployment considerations vary widely depending on the scale and usage of the model:
    - Small portfolio project?
    - Enterprise level deployment?

Model Deployment

- Purpose of Deployment
    - Small Portfolio Project
        - Consider writing a “blog” post instead of full deployment.
        - Set up simple API Flask-based website, possibly on a free tier service like Heroku.
            - Requires web dev skills.

Model Deployment

- Purpose of Deployment
  - Enterprise Level
    - Need to make considerations across multiple stakeholders.
    - Typically not in the realm of the data scientist who created the model.
    - Communicate with your team!

Model Deployment

- Performance Expectations
  - Make sure to set clear expectations on model performance based on cross validation (final hold-out set).
  - Do **not** set expectations based on the fully trained model, as it will not be representative of the true performance on unseen data.

Model Deployment

- Performance Expectations
    - Model Training Workflow:
        - Train|Test|Validation split.
        - Hyperparameter tuning.
        - Report results on final holdout set.
        - Retrain model on all data prior to deployment.

Model Deployment

- Retraining Intervals:
  - After deployment, how often should we retrain our model on new incoming data?
    - Answer: It completely depends on your situation!
  - Let’s consider some key factors.

Model Deployment

- Retraining Intervals Considerations:
  - Is performance still good?
    - More nuanced and harder to clarify once model is deployed, since in theory you wouldn't have the correct "labeled" data.
    - Depends on situation, use your best judgement here!

Model Deployment

- Retraining Intervals Considerations:
  - How often are you getting new data?
    - Data Considerations:
      - Size of new data.
      - New data is labeled.
      - Percentage of total data that was used for training.

Model Deployment

- Retraining Intervals Considerations:
  - How often are you getting new data?
    - For example, we have a total of 1GB of original training data.
    - But we receive 1GB of new labeled data a month!
    - After a month, we've only trained on 50% of available data.

Model Deployment

- Final Thoughts:
  - Model creation and deployment in an organization is almost never the purview of a single person.
  - Use your domain expertise and colleagues to figure out the best strategy.

Model Deployment

- Final Thoughts:
  - At the end of the day, there are no set 100% correct rules or answers.
  - We've seen how machine learning constantly evolves, use this degree of freedom to innovate!

Model Persistence

Model Deployment

- Let’s quickly review the “lifecycle” of creating, training, saving, and loading a machine learning model with Scikit-Learn.
- We’ll set up a saved model to be used in the next series of API lectures.

Model Deployment API

Part One: General Overview

PIERIAN DATA

Model Deployment

- The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.
- Two key communication protocols are GET and POST, which allow a client to obtain information and provide information.

Model Deployment

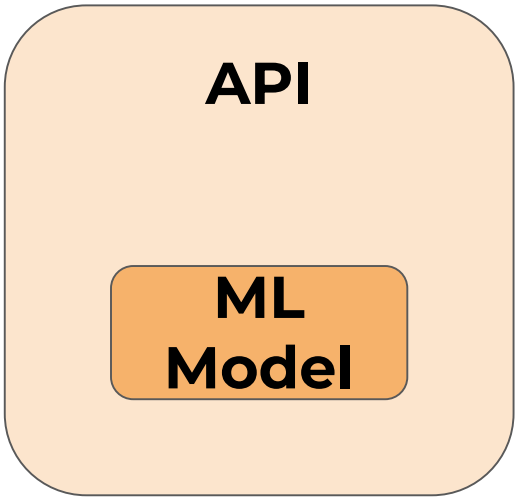
- API stands for Application Programming Interface.
- An API can serve as an interface for GET and POST requests.
- Our goal is to let our Scikit-Learn model be “served” as an API which can get and receive information.

Model Deployment

- General Overview

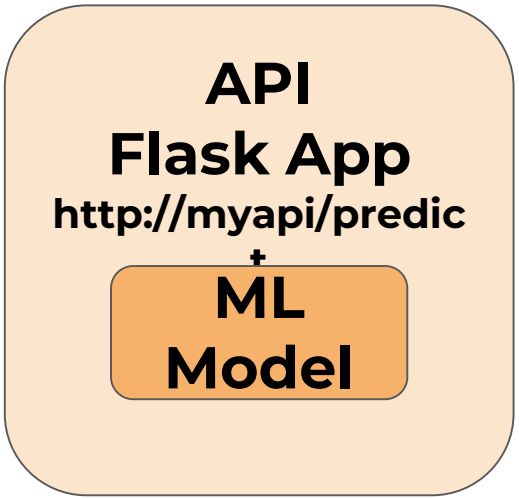
Model Deployment

- General Overview



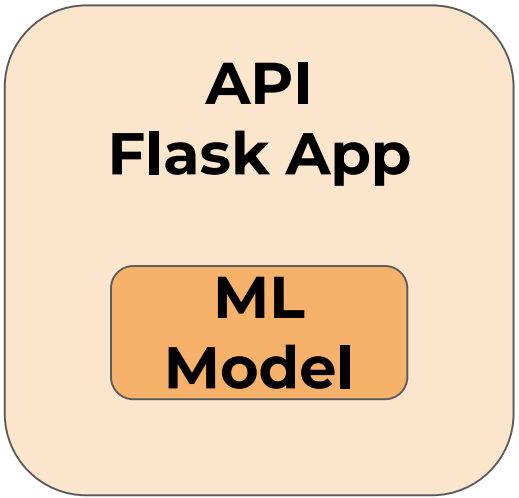
Model Deployment

- General Overview



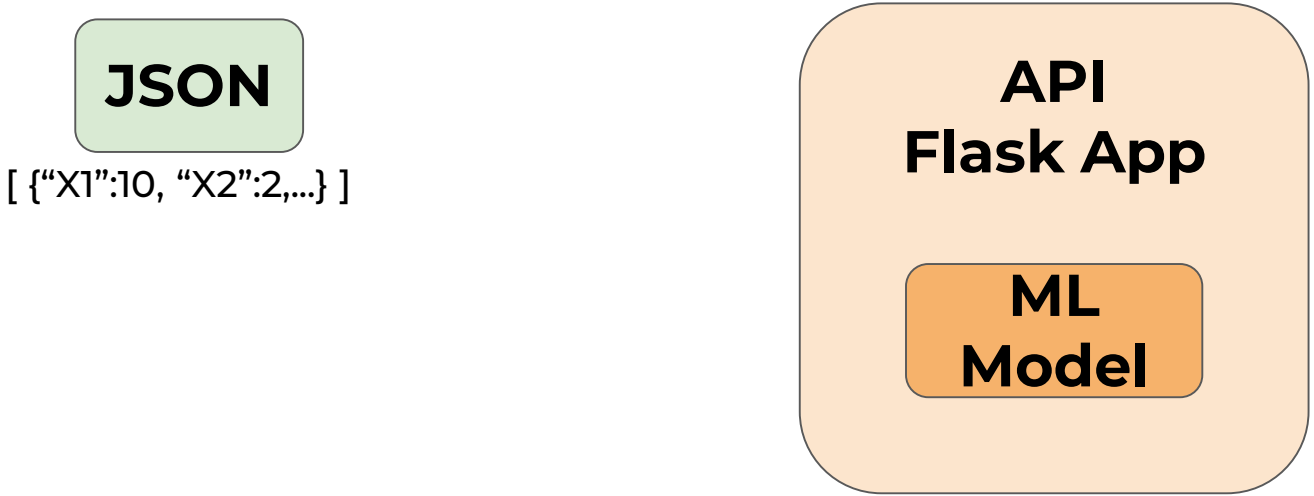
Model Deployment

- General Overview



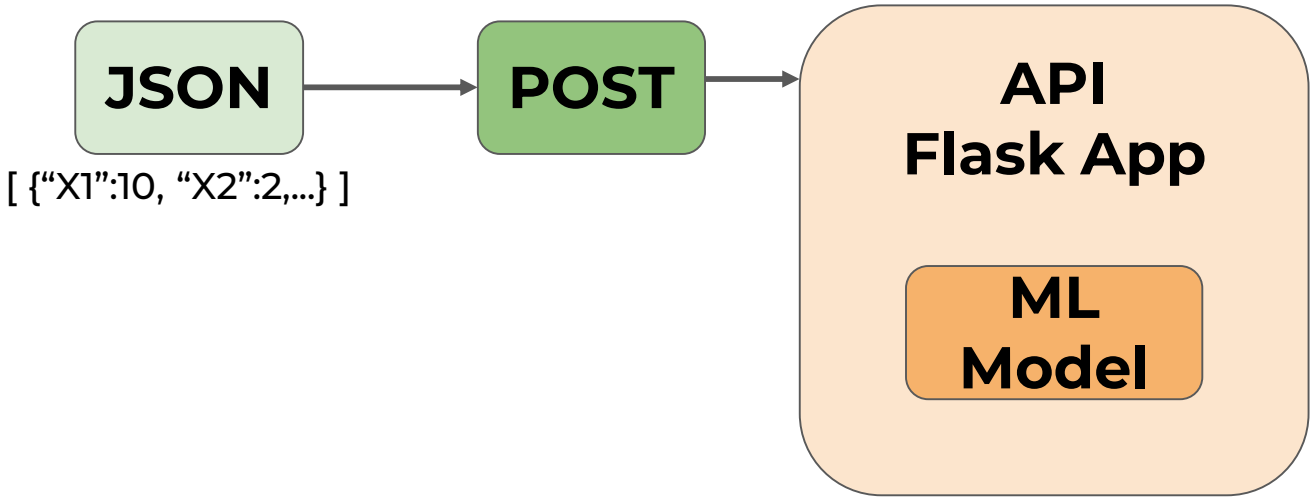
Model Deployment

- General Overview



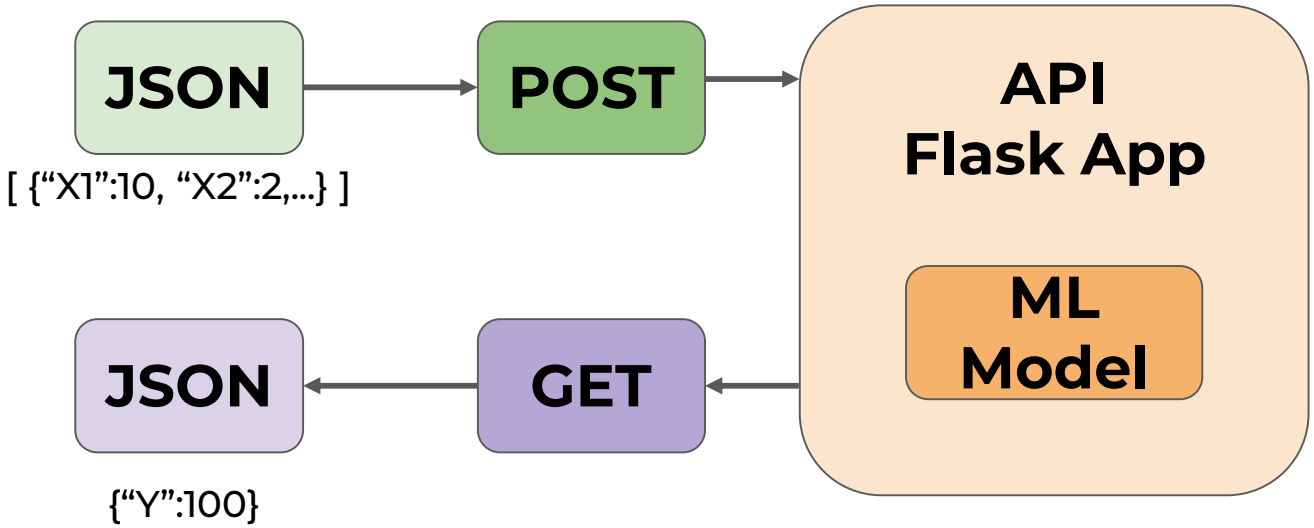
Model Deployment

- General Overview



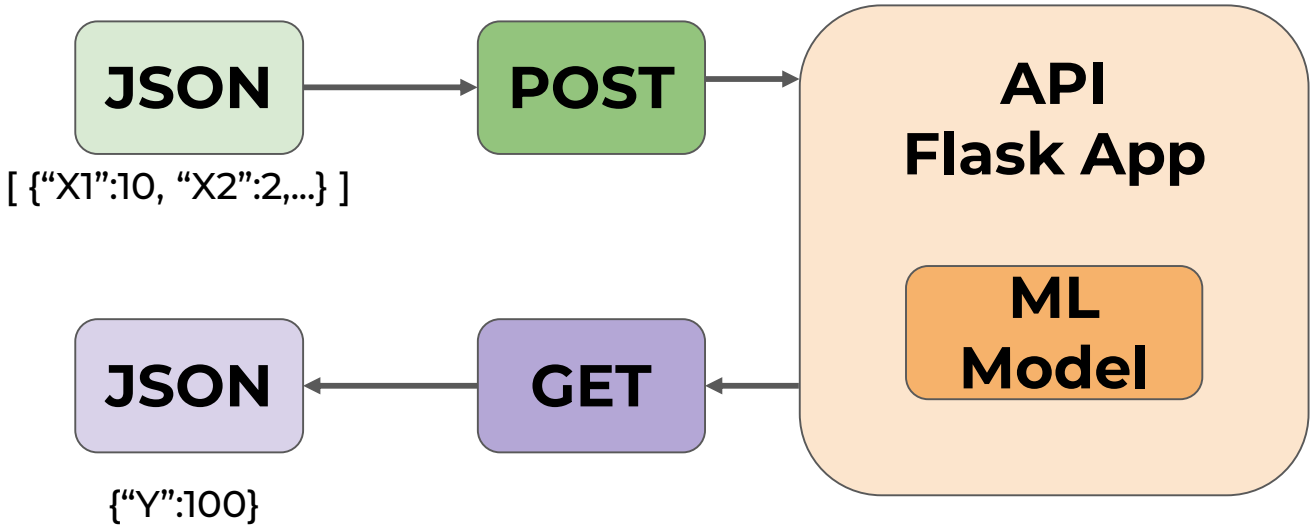
Model Deployment

- General Overview



Model Deployment

- General Overview

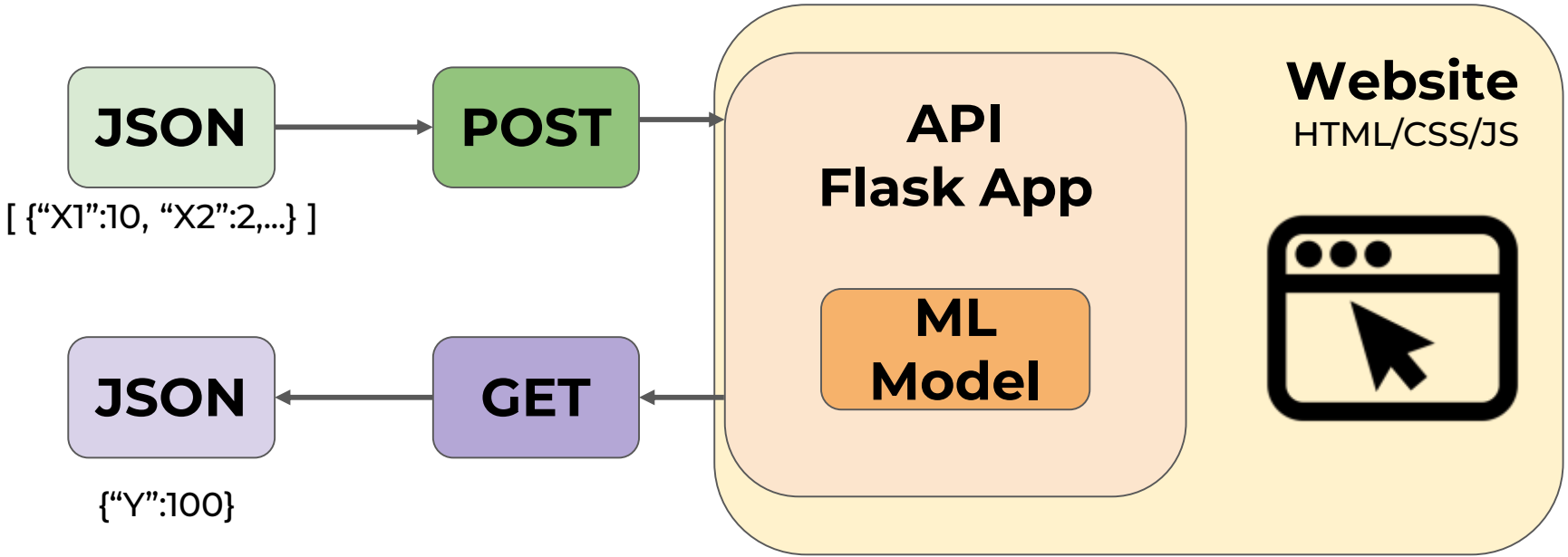


POSTMAN

PIERIAN DATA

Model Deployment

- General Overview



Model Deployment

- Model API Deployment Steps:
  - Install Flask
  - Create simple Flask App for API
  - Connect ML model to Flask API
  - Install Postman
  - Test API through Postman

Model Deployment

- ***Important Notes:***
  - We will only create the simple API interface and not a full website (full website would require HTML, CSS, and Web Development experience).
  - Take a look at a Django or Full-Stack Flask course for more information on creating web applications.

Model Deployment

- ***Important Notes:***
  - Due to the nature of a Flask API being served through HTTP, Jupyter can interfere with the application.
  - You must run the API code as a Python Script file (.py file).
  - Feel free to use **any** preferred editor for this.

Model Deployment

- Coming up next, let’s install Flask and create our Flask API Routing call in a .py python script for our Scikit-Learn model!

Model Deployment API

Part Two: Creating API Script

PIERIAN DATA

Model Deployment

- Important note!
  - We will need to install Flask library
    - **pip install Flask**
    - **conda install Flask**
  - Or install through Anaconda Navigator by searching for Flask library.

Model Deployment API

Part Three: Testing the API

PIERIAN DATA

Model Deployment

- Important note!
  - We will need to download and install Postman:
    - https://www.postman.com