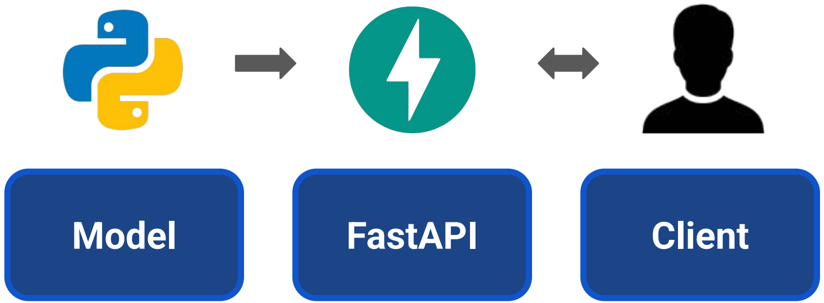
ML Meets the Web:

ML Model Deployment using FastAPI



Model

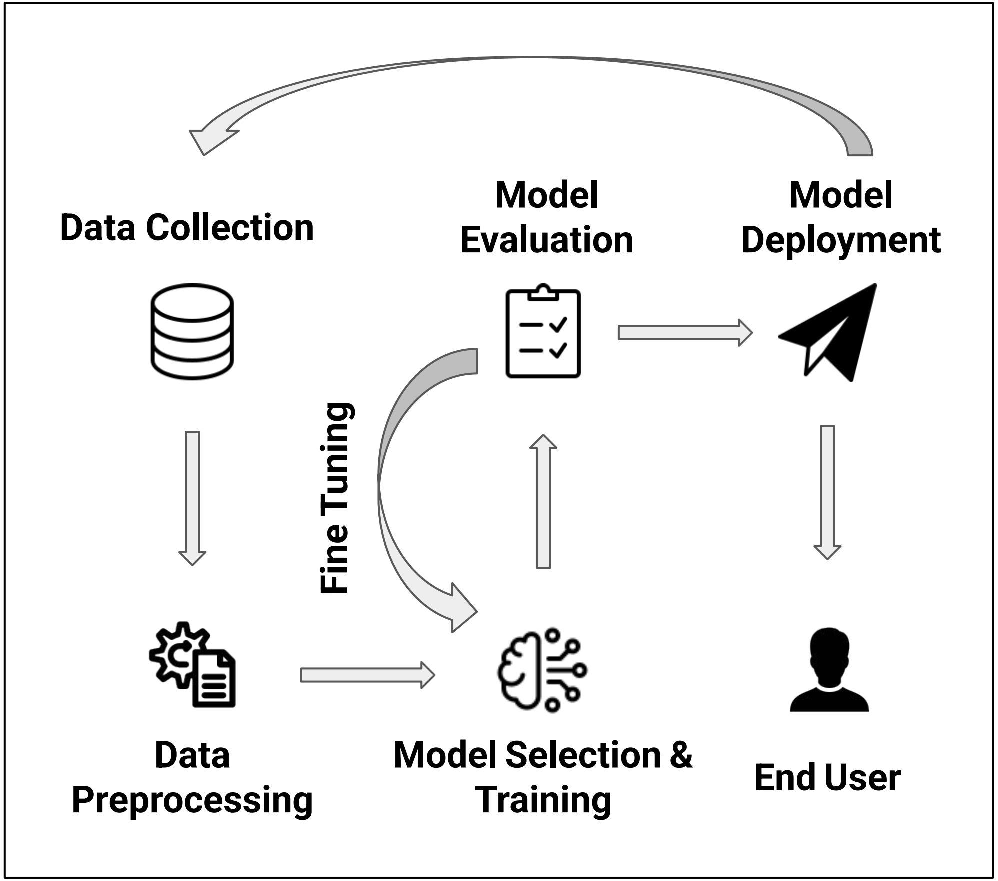
FastAPI

Client

Harshit Kumawat

What is Model Deployment?

- Model-to-production
- Real-world ML integration
- Easy access of model
- Predict with simple input
- Deployment Methods: API, cloud, docker, etc.



What is FastAPI & Why Use It?

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints.

Key Features:

- Automatic documentation
- Pydantic-based data validation
- Asynchronous support
- Fast to code
- Fewer bugs
- Easy, short, robust

Drawbacks:

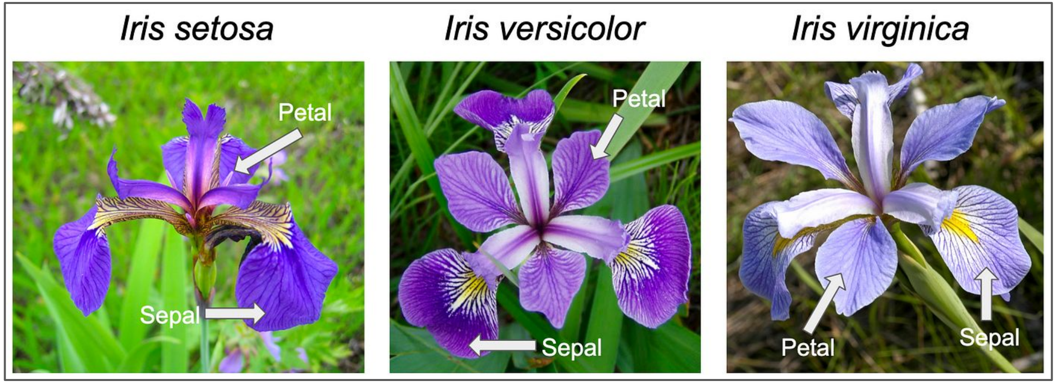
- Async complexity
- Fewer built-in tools
- Smaller ecosystem
- Debugging difficulty

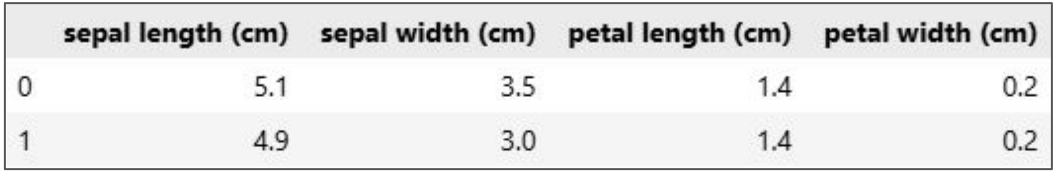
Installation & Key Concepts of FastAPI

1. **Installation:** `pip install fastapi, uvicorn, pydantic`
2. **Key Concepts:**
    a. **API Endpoints:** GET, POST, PUT, PATCH, DELETE
    b. **Response Structure:** JSON responses
    c. **Request Validation:** Done using Pydantic models
    d. **ASGI Server:** Run using Uvicorn

Iris Dataset

- Images of 3 iris species (classes)
  - 0: Setosa
  - 1: Versicolor
  - 2: Virginica
- 150 instances (50 per class)
- 4 features





	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2

Decision Tree

Gini Impurity (of i^th node):

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^{2}$$

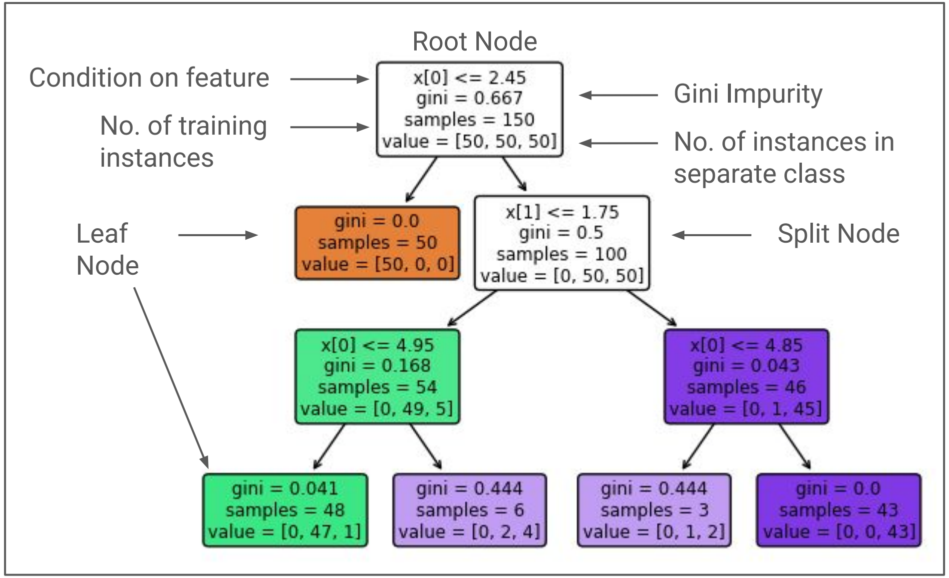
Where, $P_{i,\,k}$ is probability of k^th class in i^th node, n is no. of classes

For example,

$$G_i = 1 - \left[\left(\frac{0}{46}\right)^2 + \left(\frac{1}{46}\right)^2 + \left(\frac{45}{46}\right)^2\right]$$

$$G_i = 1 - [0 + 0.00472 + 0.95]$$

$$G_i \approx 0.043$$



Root Node

Condition on feature

No. of training instances

Gini Impurity

No. of instances in separate class

Leaf Node

Split Node

x[0] <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]

gini = 0.0
samples = 50
value = [50, 0, 0]

x[1] <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]

x[0] <= 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]

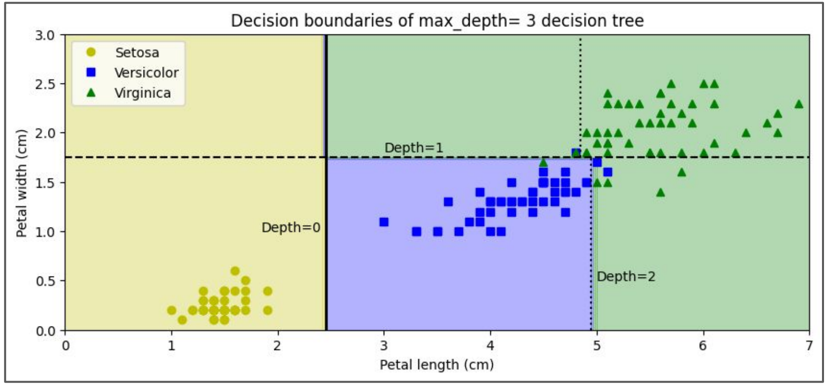
x[0] <= 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]

gini = 0.041
samples = 48
value = [0, 47, 1]

gini = 0.444
samples = 6
value = [0, 2, 4]

gini = 0.444
samples = 3
value = [0, 1, 2]

gini = 0.0
samples = 43
value = [0, 0, 43]

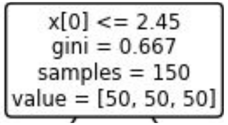


Decision boundaries of max_depth= 3 decision tree

CART Algorithm

- Classification And Regression Tree Algorithm
- Used to train decision tree
- Time Complexity:
  - Training: $O(n \times m\, log_2(m))$
  - Inference: $O(\, log_2(m))$
- Cost function (for classification)

$$J(k,\ t_k) = \frac{m_{left} G_{left}}{m} + \frac{m_{right} G_{right}}{m}$$

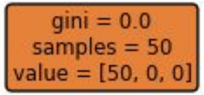


x[0] <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]

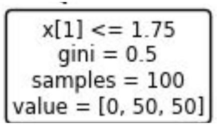
$k =$ sepal length (cm)

$t_k = 2.45$

$$G_0 = 1 - 3\left(\frac{50}{150}\right)^2 = 0.\overline{6} \approx 0.667$$



gini = 0.0
samples = 50
value = [50, 0, 0]



x[1] <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]

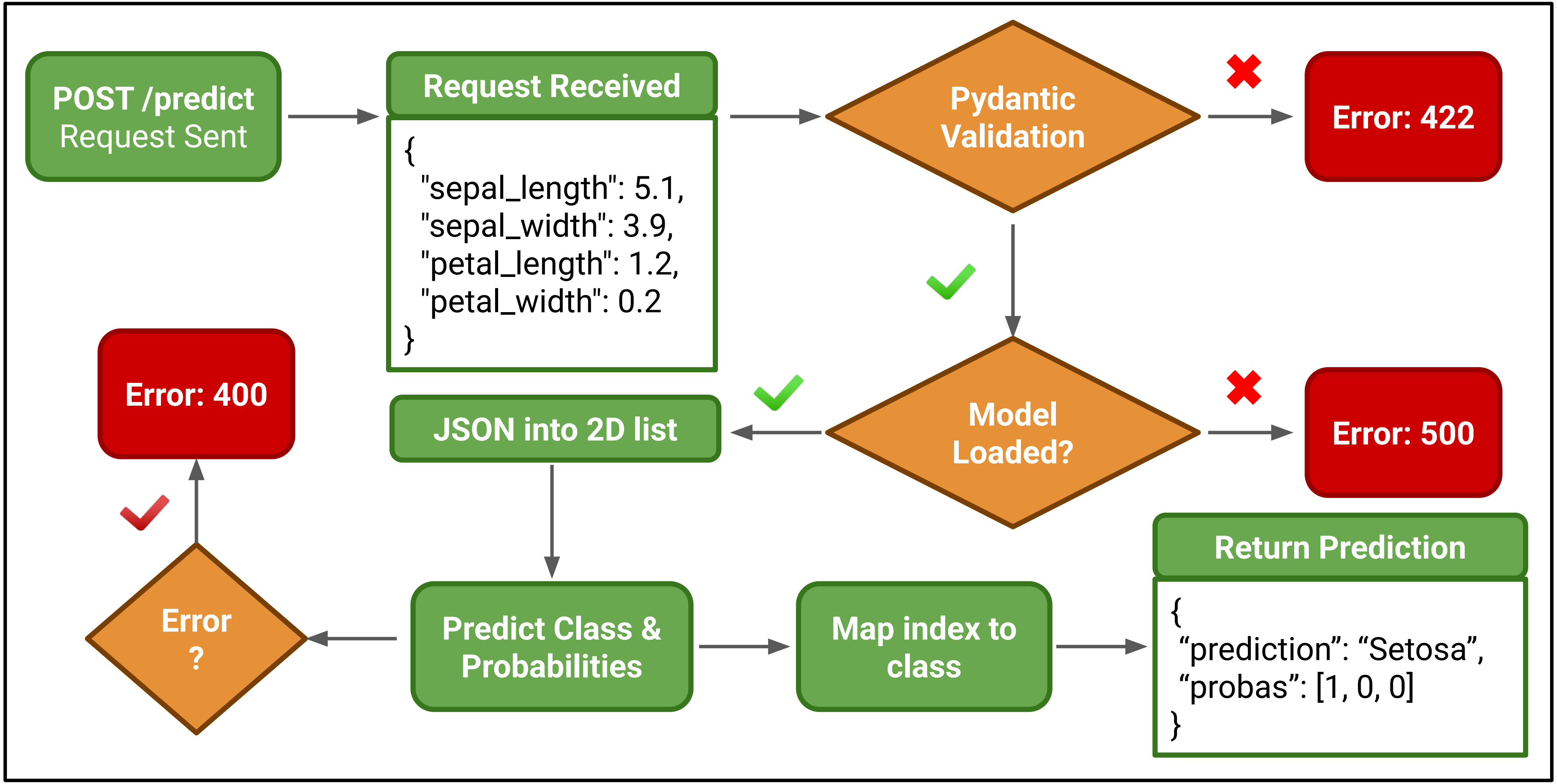
$$G_1 = 1 - \left(\frac{50}{50}\right)^2 = 0$$

$k =$ sepal width (cm)

$t_k = 1.75$

$$G_2 = 1 - 2\left(\frac{50}{100}\right)^2 = 0.5$$

$$J\left(k,\ t_k\right) = \frac{50\times0}{150} + \frac{100\times0.5}{150} = \frac{1}{3} = 0.\overline{3}$$



```
{
  "sepal_length": 5.1,
  "sepal_width": 3.9,
  "petal_length": 1.2,
  "petal_width": 0.2
}
```

```
{
 “prediction”: “Setosa”,
 “probas”: [1, 0, 0]
}
```

FastAPI-Based Model Prediction Flow

THANK YOU

Thank you for your time and consideration. I hope this walkthrough clearly explained the model deployment process using FastAPI.

Acknowledgements

Internship Assignment by:
TheProductWorks.in

Learning Resources:
FastAPI Documentation
GeeksForGeeks
FastAPI Deployment Tutorials Playlist - Krish Naik
Python FastAPI Tutorial: Build a REST API in 15 Minutes - pixegami

Attributions

Python, FastAPI and other icons for the diagrams are taken from Icons8.

Dataset: UCI Machine Learning Repository - Iris Dataset

Contact

Name: Harshit Kumawat
Email:harshitkumawat849@gmail.com
LinkedIn:https://www.linkedin.com/in/harshit-kumawat-8778ba259/

Project Repository