# Module 4: Build MLApps using New Age Tool

## Demo 6 – Documentation: FastAPI Application for Iris Species Prediction

**FastAPI Application for Iris Species Prediction**

**Solution:**

This FastAPI application serves a machine learning model trained on the Iris dataset. The model predicts the species of an Iris flower based on four input features: sepal length, sepal width, petal length, and petal width. The application defines an endpoint that receives these features via a POST request and returns the predicted species.

**Requirements**

1. Python 3.6+
2. fastapi
3. uvicorn
4. scikit-learn
5. numpy
6. pickle

To install the necessary libraries, use the following command:

```
bash
Copy code
pip install fastapi uvicorn scikit-learn numpy pickle5
```

**File Structure**

- **main.py**: Contains the API and model handling code.
- **iris_model.pkl:** The pre-trained logistic regression model on the Iris dataset.

**Code Breakdown**

main.py

```
python
Copy code
from fastapi import FastAPI
from pydantic import BaseModel
import numpy as np
import pickle

app = FastAPI()

# Data model to validate incoming requests
class IrisFeatures(BaseModel):
    sepal_length: float
    sepal_width: float
    petal_length: float
    petal_width: float

@app.post("/predict")
```

```
def predict_iris(iris: IrisFeatures):
    # Load the trained model from a file
    with open("iris_model.pkl", "rb") as f:
        model = pickle.load(f)

    # Convert the incoming JSON data to a NumPy array suitable for the
model
    test_array = np.array([[iris.sepal_length, iris.sepal_width,
iris.petal_length, iris.petal_width]])

    # Predict using the model
    prediction = model.predict(test_array)

    # Mapping the numeric prediction to a species name
    species = {0: "Iris-setosa", 1: "Iris-versicolor", 2: "Iris-
virginica"}
    predicted_species = species[int(prediction[0])]

    # Return the prediction result
    return {"prediction": predicted_species}
```

## Usage

Starting the Server:

Run the server using uvicorn with the following command:

```
bash
Copy code
uvicorn main:app –reload
```

This command starts the FastAPI application with live reloading enabled.

## Making Predictions:

To make predictions, send a POST request to http://localhost:8000/predict with a JSON payload containing the Iris features. Use a tool like curl, Postman, or any HTTP client in a programming language.

**Example curl command:**

```
bash
Copy code
curl -X 'POST' \
  'http://localhost:8000/predict' \
  -H 'Content-Type: application/json' \
  -d '{
      "sepal_length": 5.1,
      "sepal_width": 3.5,
      "petal_length": 1.4,
      "petal_width": 0.2
    }'
```

## Error Handling

The application assumes the model file **iris_model.pkl** is correctly placed and accessible. Error handling for missing or corrupted model files, or invalid input data, should be implemented for production use.

## Conclusion

This FastAPI application provides a simple and effective way to serve a machine learning model for predicting Iris species. It demonstrates the use of FastAPI for machine learning applications, including how to handle input data, perform predictions, and respond with meaningful information.